

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

DIGITÁLNÍ HODINY ŘÍZENÉ PROTOKOLEM NTP

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

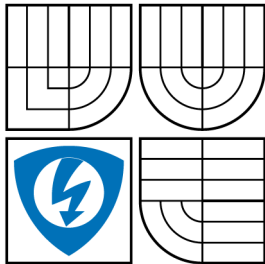
AUTOR PRÁCE
AUTHOR

LUKÁŠ VYKYDAL

Brno 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

DIGITÁLNÍ HODINY ŘÍZENÉ PROTOKOLEM NTP DIGITAL CLOCK SYNCHRONIZED BY NTP PROTOCOL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LUKÁŠ VYKYDAL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JIŘÍ SEKORA,

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Bakalářská práce

bakalářský studijní obor
Elektronika a sdělovací technika

Student: Lukáš Vykydal

ID: 154908

Ročník: 3

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Digitální hodiny řízené protokolem NTP

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s principem synchronizace času pomocí NTP (Network Time Protocol). Navrhněte modul samostatného hardwarového klienta, který bude připojen do sítě Ethernet a bude disponovat zobrazovací jednotkou, na které bude po synchronizaci zobrazován čas. Sestavte příslušný program.

Realizujte modul klienta s rozhraním Ethernet, který po spuštění synchronizuje čas pomocí NTP. Čas bude zobrazován na display klienta a bude aktualizován ve zvoleném intervalu.

DOPORUČENÁ LITERATURA:

[1] KABELOVÁ, A., DOSTÁLEK, L. Velký průvodce protokoly TCP/IP a systémem DNS. 5., aktualiz. vyd. Brno: Computer Press, 2008, 488 s. ISBN 978-80-251-2236-5.

[2] MILLS, D. et al.: RFC 5905 - Network Time Protocol Version 4: Protocol and Algorithms Speci

Termín zadání: 9.2.2015

Termín odevzdání: 28.5.2015

Vedoucí práce: Ing. Jiří Sekora

Konzultanti bakalářské práce:

doc. Ing. Tomáš Kratochvíl, Ph.D.

Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Práce popisuje princip synchronizace pomocí protokolu NTP a potřebnou úroveň síťové komunikace. Následně se práce zabývá návrhem digitálních hodin, jejichž čas bude synchronizován s časovým normálem přes síť Internet. Tím bude dosaženo automatické nastavení času a jeho udržování s přijatelnou odchylkou.

KLÍČOVÁ SLOVA

digitální hodiny, NTP, síťová komunikace, UDP, ARM

ABSTRACT

The goal of this work is the principle of NTP synchronization and required level of network communication. In the second part the work describes design of digital clock which is synchronized using this technique. That way we achieve automatic time adjustment and keeping the time within reasonable error.

KEYWORDS

digital clock, NTP, network communication, UDP, ARM

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Digitální hodiny řízené protokolem NTP“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu semestrální práce panu Ing. Jiřímu Sekorovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)

OBSAH

Úvod	10
1 Měření času	11
1.1 Časové systémy	11
1.2 Synchronizace času	11
1.2.1 Synchronizace vysíláním časových značek	11
1.2.2 Synchronizace dotázáním na čas	12
2 Protokol NTP	13
2.1 Formát času	13
2.2 Protokol NTP	13
2.3 Kiss-o'-Death paket	16
2.4 Dotaz na server	17
2.5 Synchronizační algoritmus	18
2.5.1 Zpracování odpovědi serveru	18
2.5.2 Filtrační algoritmus	19
2.5.3 Selekční algoritmus	19
2.5.4 Skupinový algoritmus	20
2.5.5 Kombinační algoritmus	20
2.6 Porovnání NTP/SNTP	23
2.7 Návrh pooling intervalu	23
2.8 NTP servery	24
3 Realizace hardwaru	25
3.1 MPU - ARM	25
3.1.1 SPI	25
3.1.2 Programování	28
3.1.3 Napájení mikrokontroléru	28
3.2 WizNet W5100	28
3.3 Napájení	29
3.3.1 Popis PoE	29
3.3.2 Implementace PoE	30
3.4 Zobrazení	31
4 Realizace software	32
4.1 Periferie mikrokontroléru	32
4.1.1 PMC	32
4.1.2 PIO	32

4.1.3	Sběrnice SPI	33
4.1.4	Rozhraní UART	33
4.1.5	Časovač/čítač	33
4.2	WizNet	33
4.3	DHCP klient	34
4.4	SNTP klient	34
4.4.1	Zpracování odpovědi	35
4.4.2	Převedení časového razítka	36
5	Výsledky	39
5.1	Test časové stability	39
5.2	Problém první synchronizace	39
	Závěr	41
	Literatura	42
	Seznam zkratk	44
	Seznam příloh	45
A	Schéma zapojení	46
B	Obsah příloženého CD	55

SEZNAM OBRÁZKŮ

2.1	Synchronizace NTP	18
2.2	Vývojový diagram Marzullova algoritmu	21
2.3	Vývojový diagram skupinového algoritmu	22
3.1	Blokové schéma navrženého zařízení	26
3.2	Přenos 1 bajtu na SPI sběrnici - Mód 0	27
3.3	Další módy SPI sběrnice	27
4.1	Stavový automat DHCP klienta	35
4.2	Stavový automat SNTP klienta	36
4.3	Zpracování odpovědi od NTP serveru	37
4.4	Vývojový diagram převodu časového razítka na čas	38
A.1	Rozdělení schématu na bloky	46
A.2	Schéma multiplexu pro digitronový displej	47
A.3	Schéma síťového rozhraní s čipem WizNet W5100	48
A.4	Schéma PoE interface	49
A.5	Schéma zapojení ARM procesoru	50
A.6	Schéma 3V3 stabilizátoru včetně výpočtů hodnot	51
A.7	Schéma měniče pro napájení digitronů včetně výpočtů hodnot	52
A.8	Deska pločného spoje, vrchní strana	53
A.9	Deska plošného spoje, spodní strana	54

SEZNAM TABULEK

2.1	Formát NTP paketu	14
2.2	První bajt paketu	14
2.3	Módy protokolu	15
2.4	Význam Leap Indicator bitů	15
2.5	Seznam identifikátorů časových normálů	16
2.6	Seznam Kiss-o'-Death kódů	17
2.7	Přesnost hodin v závislosti na pooling intervalu.	24
2.8	NTP servery	24
3.1	Základní porovnání ATSAM3X8E a ATSAM3S2C	25
3.2	SPI zprávy pro WizNet W5100	28
3.3	Třídy PD dle IEEE 802.3af	30
3.4	Úbytky napětí pro různé výkony na napětí pro CAT5e kabel.	30
5.1	Test synchronizace pro $\delta = 79$ s.	40

ÚVOD

Čas je jednou ze základních fyzikálních jednotek. Čas lze chápat jako polohu na časové ose, která je jednou ze čtyř os časoprostoru.

Jednotkou času je sekunda [s]. Dříve byla vztažena k astronomickému roku, ale nyní je definována pomocí oscilací atomů cesia [2]. Toho se využívá v časovém normálu nazvaném atomové hodiny. Je tím dosahována frekvenční přesnost až kolem $2 \cdot 10^{-16}$ [3].

Přesnost času zobrazovaného hodinami je přímo závislá na přesnosti a stabilitě oscilátoru hodin. I krystalový oscilátor s tolerancí ± 20 ppm může vést k odchylce až 1 min za 34 dní, což může být v řadě použití již nepřijatelná chyba, proto se provádí synchronizace hodin k některému z časových normalů. Pro synchronizaci lze využít vysílané časové značky (např. DCF77, WWVB, GPS, Galileo), nebo také počítačové sítě s využitím protokolu NTP.

Cílem této práce je vysvětlit synchronizaci hodin pomocí sítě Internet s využitím služby NTP. S využitím těchto poznatků bude možné zkonstruovat digitální hodiny, jež budou synchronizovány právě pomocí sítě Internet a protokolu NTP. Tím bude zajištěna dlouhodobá přesnost zobrazovaného času na jednotky sekund.

1 MĚŘENÍ ČASU

1.1 Časové systémy

Existují dva základní časové systémy: TAI a UT.

TAI (mezinárodní atomový čas) je odvozen od atomových hodin, kde je využito oscilací atomu cesia. Jedná se o zatím nejpřesnější způsob měření času. Naměřený čas na jednotlivých atomových hodinách se koriguje a průměruje. Výsledkem je právě čas TAI.

UT (univerzální čas) je čas, který je odvozen od rotace Země. Existují tři verze univerzálního času. UT0 (dříve známý jako GMT) je libovolný čas přepočítaný na nultý poledník Země. UT1 vychází z UT0, ale navíc provádí korekci změny polohy pólů a je proto přesnější. Existuje ještě čas UT2, který vznikne korekcí UT1 o sezónní variace v rychlosti otáčení Země.

Výhody obou předchozích systémů kombinuje UTC (univerzální koordinovaný čas). Vzniká z času TAI jednosekundovými opravami tak, aby rozdíl mezi UT1 a UTC byl maximálně $\pm 0,9$ s. Tyto opravy se nazývají přestupné sekundy. Přestupná sekunda se vkládá do 30. června nebo 31. prosince.

Aktuální rozdíl je $t_{TAI} - t_{UT1} = 35$ s [11].

1.2 Synchronizace času

Cílem synchronizace času je upravit aktuální hodnotu času a udržet správnou frekvenci hodinového oscilátoru. Synchronizace času by tedy měla probíhat současně dvěma různými způsoby, a to skokovou změnou stavu hodin a úpravou frekvence hodinového oscilátoru. Výsledkem této kombinace bude velmi přesné udržování času i s krátkými výpadky synchronizačního zdroje.

Problémem synchronizace je také stav, kdy se lokální hodiny výrazně předbíhají oproti synchronizačnímu zdroji. Otázkou na konkrétní implementaci zůstává, jestli se mají hodiny skokově změnit „zpět“, což by mohlo v některých situacích způsobit problémy, nebo se má jen snížit frekvence hodin a čekat, dokud se rozdíl nevyrovná.

1.2.1 Synchronizace vysíláním časových značek

Pro velmi jednoduchou synchronizaci času byly po světě vybudovány rádiové stanice, které pravidelně vysílají přesné časové informace. V Evropě se jedná o systém DCF77, jehož vysílač se nachází v Německém Mainflingen [4]. Konkrétně DCF77 vysílá na frekvenci 77,5 kHz sekundové časové značky, na které lze zavěsit lokální oscilátor hodin.

Další možnost synchronizace pomocí časových značek představují navigační systémy (např. GPS). Tyto systémy samy o sobě vyžadují pro správnou funkci přesný čas, který vysílají společně s navigačním signálem.

Výhodou vysílání časových značek je fakt, že klient dostává informaci trvale a jen s velmi malým zpožděním (řádově desítky mikrosekund) a velikost tohoto zpoždění je přibližně známá - je daná vzdáleností přijímače od vysílače.

1.2.2 Synchronizace dotázáním na čas

Tento způsob je uplatňován v počítačových sítích. Na rozdíl od předchozího způsobu se zde klient aktivně ptá na časové údaje časových služeb. Problémem v tomto případě je zpoždění sítí a samotná doba reakce vzdáleného systému.

Nejjednodušší variantou je protokol TIME definovaný v RFC868 [5]. Jedná se o službu, která na jakýkoli příchozí datagram na portu 37 odpoví počtem sekund od 0:00:00 1. 1. 1900. Tento protokol nijak nezohledňuje zpoždění počítačové sítě, tudíž výsledná chyba může být v nejhorším případě rovna odezvě sítě. V běžné situaci se může jednat i o sekundy až desítky sekund.

Tento problém řeší protokol NTP (a jeho zjednodušená varianta SNTP), který bude popsán v následující kapitole.

2 PROTOKOL NTP

Network Time Protocol (dále jen NTP) slouží k přesné synchronizaci času prostřednictvím sítí s nedeterministickou dobou odezvy. Využívá k tomu detekci doby odezvy a postupnou minimalizaci odchylky časů serverů a klienta. Dle RFC5905 lze s aktuální verzí dosáhnout přesnosti synchronizace až na desítky mikrosekund [7].

Následující sekce se bude zabývat módem klient/server. Ostatní módy protokolu NTP jsou nad rámec rozsahu této práce.

2.1 Formát času

Protokol NTP reprezentuje čas jako časové razítko (timestamp). Toto časové razítko je ve formátu čísla s pevnou desetinnou čárkou dělicí počet bitů rovnoměrně mezi celou a desetinnou část. NTP definuje tři verze časového razítka:

- NTP Short, délka 32 bit, odpovídá času 2^{16} s = 18 hodin, 12 min, 16 s
- NTP Timestamp, délka 64 bit, odpovídá času 2^{32} s = ±136 let, 36 dní, 6 h, 28 min, 16 s
- NTP Long, délka 128 bit, odpovídá času 2^{64} s

NTP Timestamp odpovídá času v rámci jedné NTP éry. Éra číslo 0 počíná o půlnoci 1. 1. 1900 UTC času a končí 7. 2. 2036 6:28:16 UTC.

2.2 Protokol NTP

Protokol NTP je postaven nad UDP¹. Pro NTP je dle IANA vyhrazen port 123 [10]. Protokol je bezstavový a využívá datagramy popsané v tabulce 2.1.

První bajt paketu má logický význam tří polí. Jejich rozdělení je vidět v tabulce 2.1. Bity 0 a 1 udávají informaci o přechodné sekundě dle tabulky 2.4. Bity 2 až 4 udávají verzi protokolu. Aktuální verze má číslo 4. Bity 5 až 7 udávají typ zprávy dle tabulky 2.3.

Stratum udává úroveň serveru v synchronizační struktuře:

- 0 — neplatná odpověď nebo Kiss-o'-Death paket
- 1 — primární server (připojený k časovému standardu)
- 2–15 — servery synchronizované pomocí NTP; číslo značí úroveň v synchronizační struktuře
- 16 — nesynchronizovaný server

¹angl. User Datagram Protocol

Tab. 2.1: Formát NTP paketu

LI/VN/Mode	Stratum	Poll	Precision
Root Delay (32 bit)			
Root Dispersion (32 bit)			
Reference Identifier (32 bit)			
Reference Timestamp (64 bit)			
Originate Timestamp (64 bit)			
Receive Timestamp (64 bit)			
Transmit Timestamp (64 bit)			
Extension Field 1 (proměnný)			
Extension Field 2 (proměnný)			
Key Identifier (32 bit)			
Message Digest (128 bit)			

Tab. 2.2: První bajt paketu

Bity	0	1	2	3	4	5	6	7
Význam	Přestupná sekunda dle 2.4		Verze			Mód dle 2.3		

Poll udává maximální čas mezi dvěma úspěšnými požadavky pro udržení synchronizace. Při přenosu representován jako dvojkový logaritmus skutečné hodnoty.

Precision udává přesnost hodinového zdroje. Při přenosu je representován jako

Tab. 2.3: Módy protokolu

Číslo	Mód
0	Rezervovaný
1	Symetrický aktivní
2	Symetrický pasivní
3	Požadavek klienta
4	Odpověď serveru
5	Broadcast
6	NTP kontrolní zpráva
7	Vyhrazeno pro lokální použití

Tab. 2.4: Význam Leap Indicator bitů

Číslo	Význam
0	Žádné upozornění
1	Poslední minuta dne bude mít 61 sekund
2	Poslední minuta dne bude mít 59 sekund
3	Neznámo - nesynchronizováno

dvojkový logaritmus skutečné hodnoty.

Root Delay je doba odezvy k referenčnímu zdroji hodin. Tato hodnota je uložena ve formátu NTP Short timestampu.

Root Dispersion je nejistotou synchronizace k referenčnímu zdroji hodnot. Hodnota je ve formátu NTP Short timestampu.

Reference Identifier udává zdroj synchronizace pro daný synchronizační server. V případě primárních serverů jde o typ časového normálu (dle tabulky 2.5), u nižších úrovní synchronizace je zde uložena uložena adresa synchronizačního serveru (IPv4), nebo čtyři první bajty MD5 hashe IPv6 adresy.

Extension Field (1,2) umožňuje pomocí NTP předávat další kontrolní zprávy. Jejich seznam je definovaný v RFC5906 [8]. Primárně se jedná o zprávy pro vyjednání vazby².

Key Identifier a Message Digest formují volitelnou část pro ověření pravosti NTP datagramu.

²angl. association - myšleno jako vazba mezi serverem a klientem.

Tab. 2.5: Seznam identifikátorů časových normálů

Identifikátor	Časový normál
GOES	Geostacionární satelit
GPS	System GPS
GAL	System Galileo
PPS	Libovolný zdroj 1 puls za skundu
IRIG	Inter-Range Instrumentation Group (jde o formát dat)
WWVB	DV stanice WWVB, Fort Collins Colorado 60 kHz
DCF	DV stanice DFC, Mainflingen Německo 77,5 kHz
HBG	DV stanice HBG, Prangins 75 kHz
MSF	DV stanice MSF, Anthorn Velká Británie 60 kHz
JJY	DV stanice JJY, Fukušima Japonsko 40 kHz, Saga Japonsko 60 kHz
LORC	SV stanice Loran C 100 kHz
TDF	SV stanice Allouis, Francie 162 kHz
CHU	KV stanice CHU, Ottawa Ontario
WWV	KV stanice WWV Fort Collins Colorado
WWVH	KV stanice WWVH Kauai Havaj
NIST	NIST telefonní modem
ACTS	NIST telefonní modem
USNO	USNO telefonní modem
PTB	Evropský telefonní modem

2.3 Kiss-o'-Death paket

Přijatá zpráva s hodnotou Stratum = 0 může obsahovat upozornění klienta o problémech při synchronizaci. Význam paketu je definovaný obsahem Reference Identifier pole dle tabulky 2.6. Aktuální seznam Kiss-o'-Death kódů spravuje IANA. V těchto paketech není význam časových polí validní.

Nejvýznamnější jsou kódy „DENY“, „RSTR“, „RATE“. Tyto je třeba implementovat i v SNTP klientu. Kódy začínající X jsou určeny pro testování a je vhodné je ignorovat.

³paket určený pro konkrétní cíl

⁴paket určený pro celou síť ve které se síťové zařízení nachází

Tab. 2.6: Seznam Kiss-o'-Death kódů

Reference Identifier	Význam
ACST	Vazba (klient/server) patří <i>unicast</i> ³ serveru
AUTH	Ověření klienta selhalo
AUTO	Autokey ověření selhalo
BCST	Vazba (klient/server) patří <i>broadcast</i> ⁴ serveru
CRYP	Kryptografické ověření (nebo identifikace) selhalo
DENY	Nepovolený přístup k serveru
DROP	Ztracené spojení v symetrickém módu
INIT	Vazba (klient/server) nebyla ještě synchronizována
MCST	Vazba (klient/server) patří k dynamicky vyhledaným serverům
NKEY	Klíč není nainstalován nebo není důvěryhodný
RATE	Nepovolený přístup z důvodu příliš častých požadavků
RSTR	Nepovolené připojení k serveru
RMOT	Předání vazby od jiného serveru s ntpdc
STEP	Došlo ke skokové změně času a vazba se ještě znovu nesynchronizovala
X***	Pro testovací účely - musí být zahozen klientem

2.4 Dotaz na server

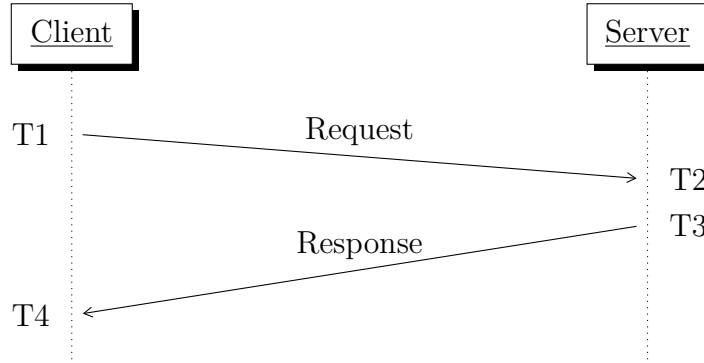
Při dotazu na službu NTP klient vytvoří dle struktury 2.1 paket, který následovně naplní:

- Vyplní mód požadavku na Client a doplní verzi protokolu (aktuálně v. 4)
 - Transmit Timestamp (T1) nastaví na hodnotu aktuálního času
- a odešle jej na vybraný NTP server. Průběh požadavku je vidět na obrázku 2.1

Po přijetí má klient k dispozici následující časové údaje:

- T1 — Originate Timestamp - čas odeslání požadavku od klienta
- T2 — Receive Timestamp - čas přijetí požadavku serverem
- T3 — Transmit Timestamp - čas odeslání odpovědi ze serveru
- T4 — Destination Timestamp - čas přijetí odpovědi klientem

Časy T1 a T4 leží na klientově časové ose a T2 a T3 leží na časové ose serveru. Pro další výpočty se předpokládá, že hodnota času je rozdílná, ale frekvence hodin jsou si blízké.



Obr. 2.1: Synchronizace NTP

2.5 Synchronizační algoritmus

Následující sekce se zabývá popisem algoritmu popsaného v normě RFC5905 [7], kapitoly 8 až 13 a v knize *Computer network time synchronization: the Network Time Protocol*. [9]

Celý proces synchronizace lze rozdělit do pěti kroků. Jedná se o filtraci a statistické vyhodnocení odchylek časů proti jednotlivým NTP serverům.

2.5.1 Zpracování odpovědi serveru

Pro jednotlivé odpovědi jsou vypočítané hodnoty časové odchylky θ , doby odezvy δ a maximální chyby času ϵ dle:

$$\theta = \frac{T4 + T1}{2} - \frac{T3 - T2}{2} \quad [s;s,s,s,s] \quad (2.1)$$

$$\delta = (T4 - T1) - (T3 - T2) \quad [s;s,s,s,s] \quad (2.2)$$

$$\epsilon(t_0) = r \cdot \rho + s \cdot \rho + \Phi \cdot (T4 - T1) \quad [s;s,s,-,s,s] \quad (2.3)$$

$$\epsilon(t) = \epsilon(t_0) + \Phi \cdot (t - t_0) \quad [s;s,-,s,s] \quad (2.4)$$

kde $T1 - T4$ jsou jednotlivé časové značky z odpovědi, $r \cdot \rho$ je přesnost vzdáleného serveru (proměnná Precision z odpovědi serveru), $s \cdot \rho$ je přesnost lokálního zdroje hodin včetně přesnosti čtení hodin a Φ je přesnost lokálního zdroje hodin.

Hodnoty $[\delta, \theta, \epsilon, t]$ jsou stěžejní pro další výpočty. Proměnná t je zde rovna času přijetí odpovědi v časové ose klienta (hodnota $T4$).

Ve výrazech 2.3 a 2.4 je vhodné povšimnout si, že zohledňuje i maximální chybu hodin během čekání na odpověď serveru a zpracování požadavku.

2.5.2 Filtrační algoritmus

Cílem filtračního algoritmu je vybrat takové odpovědi serveru, u kterých je největší pravděpodobnost, že obsahují přesné časové údaje.

Filtrační algoritmus vždy zpracovává posledních osm odpovědí daného serveru. Vstupní data každého požadavku tvoří hodnoty $[\theta, \delta, \epsilon, t]$ z předchozího kroku. Dokud je požadavků méně než osm, doplní se kombinací $[0, \text{MAXDISP}, \text{MAXDISP}, 0]$, kde MAXDISP je maximální přípustný rozptyl hodinového signálu.

Zpracování začíná seřazením odpovědí dle doby odezvy δ . Poté se vybere odpověď s nejmenší hodnotou δ a porovná se čas přijetí odpovědi s poslední odpovědí, která již prošla filtrem. Pokud je novější, algoritmus pokračuje. Dále tedy pokračuje vždy odpověď s nejkratší dobou odezvy δ taková, která ještě nebyla následujícími kroky zpracována.

Rozptyl odpovědí serveru se vypočítá jako exponenciální klouzavý průměr.

$$\epsilon = \sum_{i=0}^{n-1} \frac{\epsilon_i}{2^{i+1}} \quad (2.5)$$

Dále se spočítá *jitter*⁵ daného serveru. Využije se k tomu výraz 2.6. Dle výrazu si všimneme, že jde o efektivní hodnotu rozdílů časových odchylek proti časové odchylce „nejrychlejší“ odpovědi. Tato hodnota nám říká, jak jsou časové údaje od daného serveru stabilní.

$$\psi = \frac{1}{n-1} \sqrt{\sum_{j=1}^{n-1} (\theta_0 - \theta_j)^2} \quad (2.6)$$

Výstupem tohoto bloku jsou hodnoty $[\delta, \theta, \epsilon, \psi, t]$, kde ϵ, ψ jsou nově spočítané hodnoty a δ, θ, t jsou hodnoty z odpovědi s nejkratší dobou odezvy.

Dále lze vyjádřit synchronizační vzdálenost:

$$\lambda(t) = \frac{\delta}{2} + \epsilon(t) \quad (2.7)$$

2.5.3 Selekční algoritmus

Cílem selekčního algoritmu je najít takovou množinu odpovědí NTP serverů, které leží v jediném celistvém a co možná nejdelším intervalu.

Jako selekční algoritmus se využívá Marzullův algoritmus [7] [1].

Předchozí dvě části synchronizačního algoritmu se prováděly zvláště pro každý server. Selekční algoritmus již odebírá výstupy filtračního algoritmu pro jednotlivé servery a z nich vybírá vhodné kandidáty pro synchronizaci hodin. Do algoritmu

⁵nestabilita zdroje frekvence; v tomto případě odvozená od stability časového rozdílu v odpovědích

vstupují jednotlivé vzorky jako intervaly $\langle \theta - \lambda, \theta + \lambda \rangle$ a algoritmus hledá nejdelší interval, který je průnikem vstupních vzorků. Hodnota λ je určena dle vztahu 2.7. Vývojový diagram k algoritmu je na obrázku 2.2.

V prvním kroku algoritmus rozdělí interval na tři body - počátek, střed a konec. Poté algoritmus postupně hledá interval, na kterém se shodne co nejvíce serverů. Přitom postupně připouští existenci „lhářů“⁶. Ty jsou postupně algoritmem vyřazeny a zůstává pouze co největší počet serverů, které se shodnou na jednom časovém intervalu (při zohlednění jejich chyb).

2.5.4 Skupinový algoritmus

Cílem skupinového algoritmu⁷ je vyřadit takové servery, které mají *jitter* od ostatních serverů větší než nejlepší klient od svého serveru. Vyřazením takového serveru dojde ke snížení celkového *jitteru* a tím ke zpřesnění synchronizace času.

Algoritmus setřídí odpovědi dle výsledku výrazu 2.8, kde *MAXDIST* je maximální přípustný rozptyl a *p.stratum*, λ jsou hodnoty daného klienta.

$$\lambda_p = p.stratum \cdot MAXDIST + \lambda \quad (2.8)$$

Poté pro každého klienta spočítá směrodatnou odchylku *jitteru* a provede vyřazení serverů, které celkový *jitter* systému zvyšují. Toto provádí, dokud mu zbyde alespoň *DMIN* serverů (ve výchozí implementaci tři), případně dokud by již dalším odstraněným serverem nedošlo k zlepšení výsledných hodnot.

2.5.5 Kombinační algoritmus

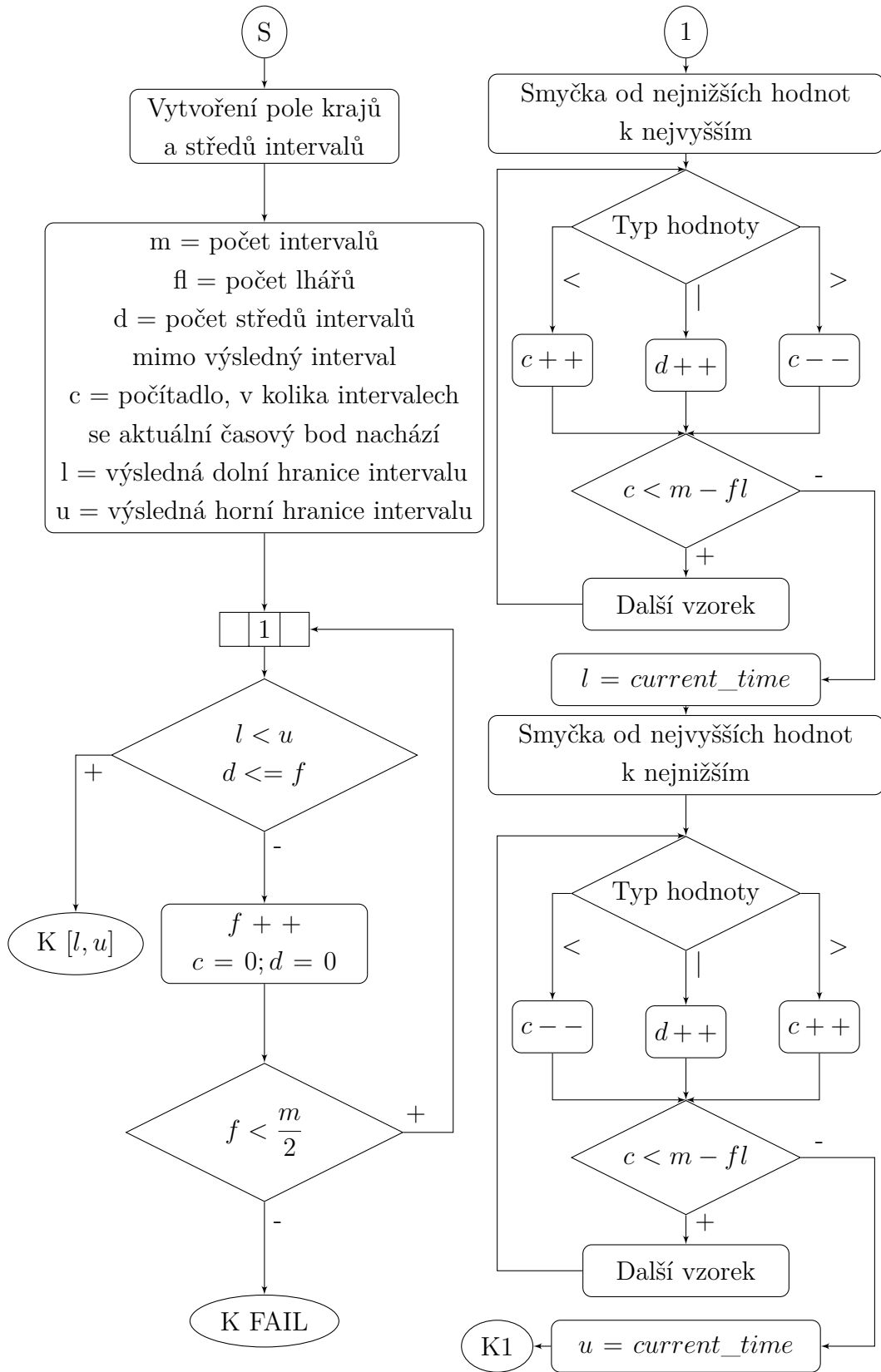
Cílem kombinačního algoritmu je vypočítat poslední a finální hodnoty na jejichž základu se provede synchronizace hodin.

Vzorky, které přežily průchod předchozími částmi algoritmu, se podílí na opravě lokálních parametrů hodin. Pro výsledný offset se použije hodnota θ dle vzorce 2.9. Jedná se o vážený průměr přeživších vzorků, vážený dle $\frac{1}{\lambda}$, kde λ má význam synchronizační vzdálenosti dle vzorce 2.7. Tato hodnota se použije pro úpravu systémových hodin (hodnotovou, případně frekvenční). Také se vypočítá nová hodnota *jitter* dle vzorce 2.10.

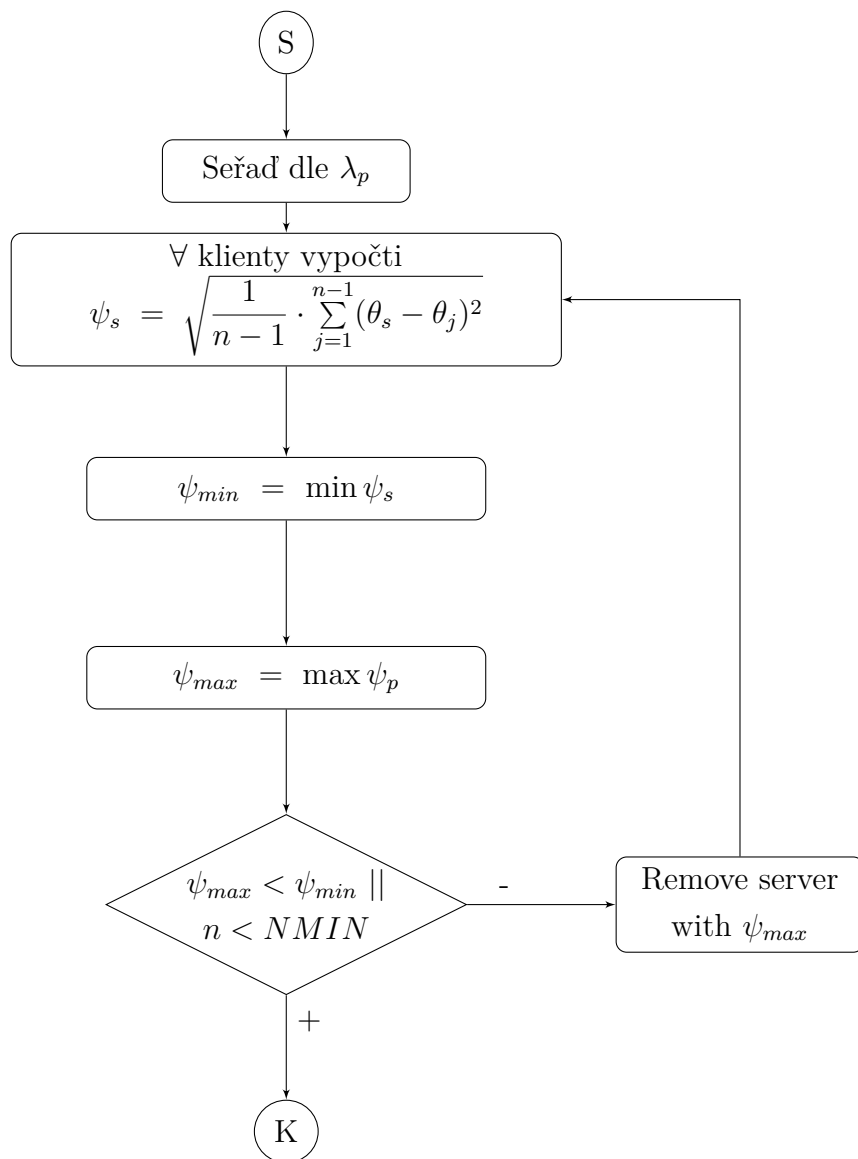
$$\theta = \frac{\sum_{n=0}^{N-1} \frac{\theta_n}{\lambda_n}}{\sum_{n=0}^{N-1} \frac{1}{\lambda_n}} \quad (2.9)$$

⁶angl. falsetickers - označuje počet odpovědí, které mají špatně určenou chybu serveru nebo jsou výrazně odlišné od zbytku odpovědí NTP serverů

⁷angl. Cluster Algorithm



Obr. 2.2: Vývojový diagram Marzullova algoritmu



Obr. 2.3: Vývojový diagram skupinového algoritmu

$$\psi = \sqrt{\frac{\sum_{n=0}^{N-1} \frac{\sqrt{\theta_n - \theta_0}}{\lambda_n}}{\sum_{n=0}^{N-1} \frac{1}{\lambda_n}}} \quad (2.10)$$

Dále můžeme vypočítat celkový rozptyl pomocí vzorce 2.11. Celkovou chybu synchronizace lze určit dle vzorce 2.12. Proměnná λ je celková doba odezvy k primárnímu serveru.

$$\epsilon = p.\epsilon_r + p.\epsilon + p.\psi + \Phi \cdot (t - p.t) + |\theta| \quad (2.11)$$

$$\lambda = \epsilon + \frac{\delta}{2} \quad (2.12)$$

2.6 Porovnání NTP/SNTP

Zjednodušení NTP protokolu - protokol SNTP - vychází ze stejného základu, neklade ovšem tak velké požadavky na přesnost synchronizace. SNTP server pracuje pouze s jedním zdrojem hodnot. SNTP klient komunikuje pouze s jedním vybraným NTP serverem. Toto výrazně snižuje náročnost implementace, jelikož většina z předchozích algoritmů nemusí být implementována.

Nevýhodou tohoto zjednodušení je, že výsledná synchronizace není odolná vůči časové chybě jednoho serveru. Oproti tomu plná NTP implementace vysoce nepřesné odpovědi zahodí nejpozději v selekčním algoritmu.

Přesnost tohoto protokolu navíc je omezena symetričností odezev mezi serverem a klientem.

2.7 Návrh pooling intervalu

Pro udržení navržené přesnosti u SNTP musí klient pravidelně kontrolovat nastavený čas proti vybranému NTP serveru. Tento časový údaj vyplývá z přesnosti lokálního oscilátoru klienta. Typicky se může jednat o krystal se stabilitou $\delta_f = \pm 20 \cdot 10^{-6}$. Stanovená přesnost v rámci této práce je $\Delta = 1$ s.

$$T_{MAX} = \frac{\Delta}{\left| \frac{1}{1 + \delta_f} - 1 \right|} \quad [s;s;-] \quad (2.13)$$

Pomocí výrazu 2.13 určíme, že je potřeba pooling interval alespoň 13,8 hod či kratší. Vliv pooling intervalu pro požadovanou přesnost je vidět na tabulce 2.7.

Tab. 2.7: Přesnost hodin v závislosti na pooling intervalu.

T_{MAX}	$T_{MAX}[s]$	$\Delta[ms](\delta_f = \pm 20 \cdot 10^{-6})$	$\Delta[ms](\delta_f = \pm 30 \cdot 10^{-6})$
10s	10	0,200	0,300
1m	60	1,200	1,800
1h	3600	71,999	107,997
2h	7200	143,997	215,994
12h	43200	863,983	1295,961
24h	86400	1727,965	2591,922

2.8 NTP servery

Jako synchronizační zdroj si lze zvolit buď konkrétní server, nebo nechat přiřazení IP adresy na *NTP poolu*⁸. NTP pool je k dispozici na `pool.net.org`. Překlad této adresy závisí na umístění tazatele a u nás vrací adresy na servery synchronizované proti serverům (`tik.cesnet.cz` a `tak.cesnet.cz`).

Tab. 2.8: NTP servery

Adresa	Startum	Reference Identifier
<code>ntp.cesnet.cz</code> (alias pro <code>tak.cesnet.cz</code>)	1	GPS
<code>time.ufr.cz</code>	1	atomové hodiny
<code>0.nettime.pool.org</code>	3	46.243.50.14
<code>time.windows.com</code>	2	216.229.0.179 (<code>nist1-lnk.binary.net</code>)
<code>nist1-lnk.binary.net</code>	1	NIST Telephone modem
<code>0.pool.ntp.org</code>	2	195.113.144.238 (<code>tak.cesnet.cz</code>)

U serveru `time.ufr.cz` jde o atomový normál ČMI [12].

K dispozici jsou také NTP servrové aplikace. Pro operační systém Windows jde o aplikaci NetTime⁹, případně lze využít službu Windows Time (součást operačního systému, NTP server nutno povolit v registrech). V rámci operačního systému Linux dk se využívá oficiální implementace NTP.

⁸skupina NTP serverů, mezi kterými probíhá synchronizace

⁹dostupná z <http://www.timesynctool.com/>

3 REALIZACE HARDWARU

Pro hardwarové řešení byly vybrány ARM mikrokontroléry od firmy Atmel. Byly zvoleny z důvodu dostatečného výpočetního výkonu i pro složitější problémy. Pro síťovou komunikaci byl vybrán síťový radič WizNet W5100. Tento radič v sobě integruje vše potřebné pro síťovou komunikaci na úrovni protokolu TCP a UDP. Jedná se o spolehlivé a jednoduché řešení.

Blokové schéma je vidět na obrázku 3.1. Jsou zde rozkresleny i čistě softwarové bloky zařízení.

Celkový návrh byl realizován na oboustranné DPS o rozměrech 130x55 mm.

3.1 MPU - ARM

Pro prvotní vývoj byla použita deska Arduino Due s mikrokontrolérem ATSAM3X8E. Jedná se o velmi dobře vybavený mikrokontrolér s jádrem Cortex-M3. Pro řešení zadané úlohy je až přespříliš vybavený a proto byl pro konečné řešení vybrán jednodušší mikrokontrolér ATSAM3S2C. Jejich srovnání je vidět v tabulce 3.1. Pro použití v tomto projektu jsou oba mikrokontroléry stejně vhodné a proto byla hlavním kritériem výběru cena.

Tab. 3.1: Základní porovnání ATSAM3X8E a ATSAM3S2C

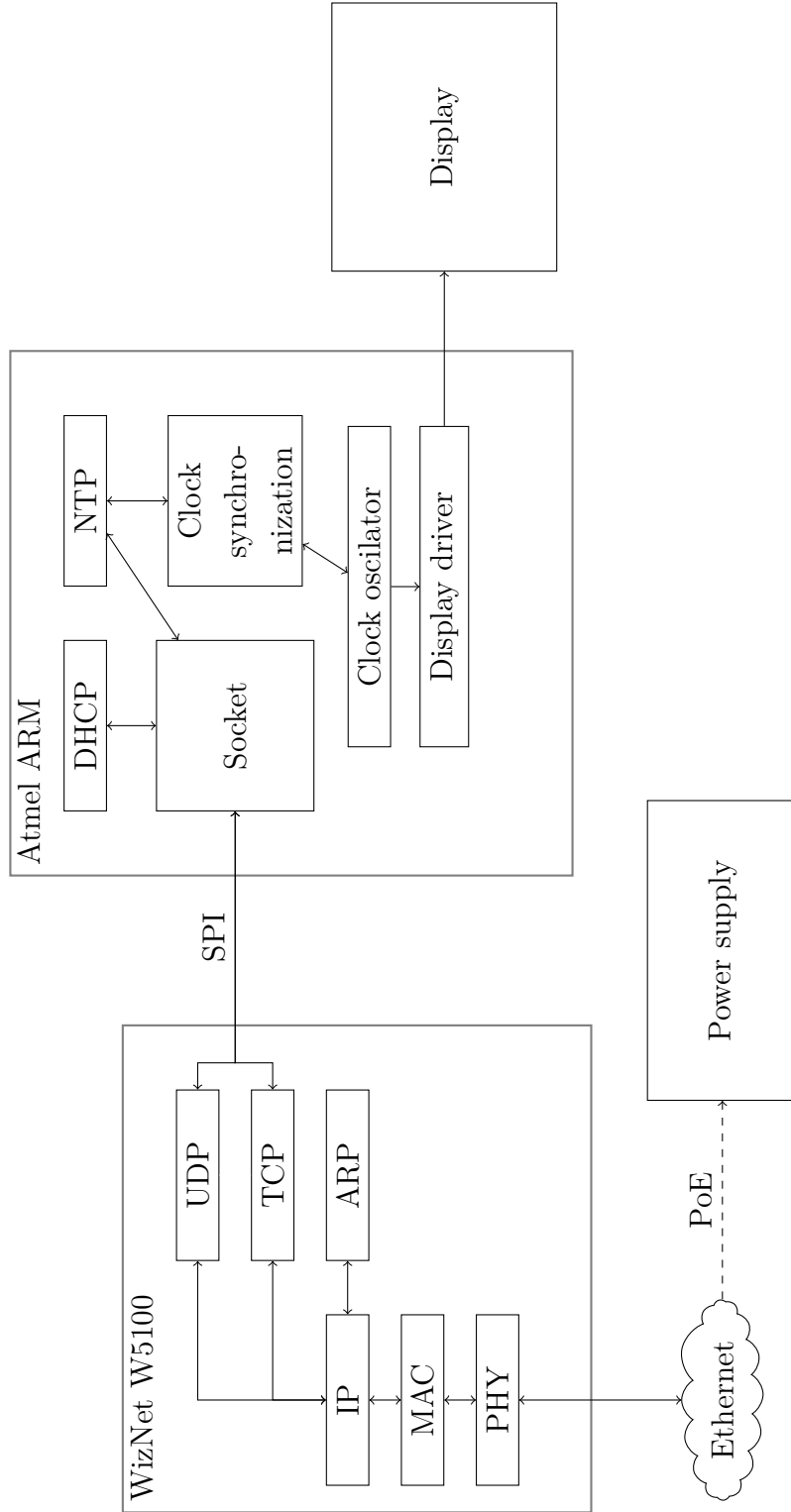
	ATSAM3X8E	ATSAM3S2C
I/O pinů	103	79
Porty	A, B, C, D	A, B, C
Flash paměť	512 kB	128 kB
SRAM paměť	96 kB	32 kB
Maximální frekvence	84MHz	64MHz
Pouzdro	LQFP-144	LQFP-100
SPI radič	1 + 3USART	1 + 2USART

Oba mikrokontroléry mají pouzdro LQFP s roztečí 0,5 mm.

3.1.1 SPI

SPI¹ je čtyřvodičová sériová sběrnice, která se často používá pro komunikaci mezi mikrokontrolérem a podpůrnými obvody. Komunikující zařízení se označují jako

¹angl. Serial Peripheral Interface - Sériová sběrnice pro připojení periférií

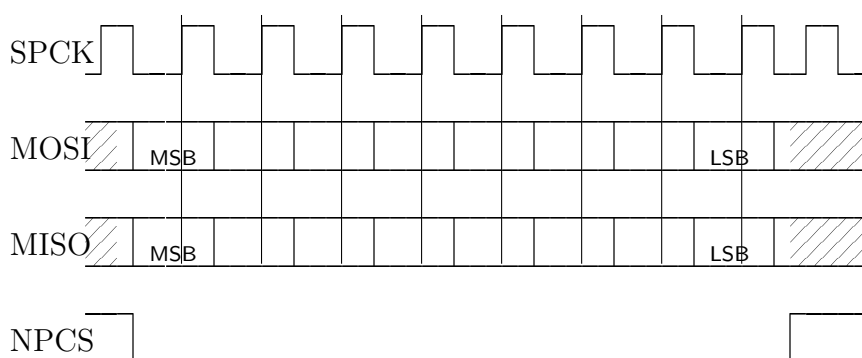


Obr. 3.1: Blokové schéma navrženého zařízení

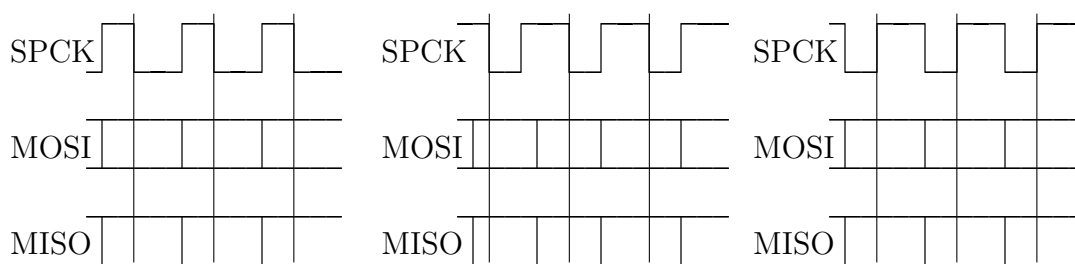
*master*² a *slave*³. Vodiče jsou označovány jako:

- MOSI, Master Out Slave In, data vyslaná *master* zařízením
- MISO, Master In Slave Out, data vyslaná *slave* zařízením
- SPCK, Serial Peripheral Clock, hodinový signál pro sběrnici
- NPCS(X), Peripheral Chip Select, výběr *slave* zařízení

U SPI sběrnice rozlišujeme čtyři módy lišící se polaritou hodinového signálu a jeho fází. Časování signálu pro celý bajt v módu 0 lze vidět na 3.2. Rozdíl mezi dalšími módy je vyznačen na 3.3. Svislé čáry udávají okamžik čtení hodnoty na datových pinech.



Obr. 3.2: Přenos 1 bajtu na SPI sběrnici - Múd 0



(a) SPI Múd 1

(b) SPI Múd 2

(c) SPI Múd 3

Obr. 3.3: Další módy SPI sběrnice

²obvod, který řídí komunikaci a generuje hodinový signál

³vybraný obvod, který na komunikaci odpovídá

3.1.2 Programování

Pro naprogramování Atmel ARM mikrokontrolérů není potřeba žádný specializovaný programátor. Tyto mikrokontroléry obsahují *bootloader*⁴ SAM-BA umístěný v ROM paměti, který umožňuje programování paměti programu pomocí UART nebo USB sběrnice. Komunikace po UART probíhá s parametry 115200Bd 8N1.

Pro programování lze využít utilitu BOSSA⁵.

3.1.3 Napájení mikrokontroléru

Mikrokontrolér využívá napětí 3,3 V a 1,8 V. Napětí 1,8 V je generováno vestavěným lineárním regulátorem, který je dostupný přímo v mikrokontroléru. To zjednodušuje návrh napájecích obvodů.

Pro napájecí napětí 3,3 V je na desce osazen snižující DC/DC měnič s obvodem MC34063.

3.2 WizNet W5100

Obvod WizNet představuje integrované řešení síťového řadiče od PHY úrovně po vrstvu protokolu TCP a UDP. Dle OSI modelu jde o fyzickou až transportní vrstvu. Pro potřeby aplikace se stačí seznámit s využitím UDP komunikace.

Komunikace s čipem W5100 je řešena pomocí sběrnice SPI s maximální hodinovou frekvencí 14 MHz. Sběrnice SPI se využívá v módu 0 nebo 3. Komunikace probíhá pomocí zpráv dlouhých čtyři bajty, pomocí nichž mikrokontrolér provádí zápis nebo čtení z paměti obvodu W5100. Přesný formát zpráv je vidět v tabulce 3.2.

Tab. 3.2: SPI zprávy pro WizNet W5100

<i>Byte</i>		1	2	3	4
Zápis	MOSI	0xF0	Vyšší bajt adresy	Nižší bajt adresy	Data
	MISO	0x01	0x02	0x03	0x04
Čtení	MOSI	0x0F	Vyšší bajt adresy	Nižší bajt adresy	–
	MISO	0x01	0x02	0x03	Data

⁴program umožňující přeprogramování mikrokontroléru, na kterém běží

⁵dostupná na <http://sourceforge.net/projects/b-o-s-s-a/>

Wiznet W5100 podporuje až 4 *sokety*⁶ zároveň. Pro využití v tomto projektu to není problém. W5100 je k dispozici v pouzdře LQFP-80, které má rozteč 0,4 mm.

3.3 Napájení

3.3.1 Popis PoE

Pro napájení zařízení bylo využito napájení pomocí ethernetového kabelu, který je současně využit pro přenos dat. V kabelu využitém pro 100BASE-T⁷ jsou dva volné páry, proto se někdy využívají pro napájení koncových zařízení. U 1000BASE-T⁸ lze využít pouze středy vinutí izolačních transformátorů pro ethernet. Obě rozhraní využívají kabelů CAT5 (1000BASE-T alespoň CAT5e), které mají měděné žíly o průměru 0,51 mm (AWG-24). Tyto žíly mají měrný odpor $0,084 \Omega m^{-1}$.

Specifikace napájení zařízení pomocí ethernetu je definována v IEEE 802.3af [13] (Power over Ethernet - zkráceně PoE). Tato norma také definuje rozdělení na napáječe (Power Sourcing Equipment - PSE) a napájené zařízení (Powered Device - PD). Tato implementace je poměrně rozsáhlá a z toho důvodu ji někteří výrobci nevyužívají a použijí pouze volné páry.

Technologie PoE využívá napájecí napětí 48 V (o maximálním dodaném výkonu 15 W) a proces detekce ke zjištění kompatibility koncového zařízení. Zjištění kompatibility sestává ze tří fází:

- Ve fázi detekce se kontroluje napětím menším než 5 V přítomnost impedance o parametrech ($R = 24,9 \Omega (\pm 1 \%)$, $C \leq 10 \text{ nF}$). Po zjištění těchto parametrů může proces připojení pokračovat.
- Ve fázi klasifikace může zařízení vznést požadavek na potřebný výkon. Při napětí 14,5 V až 20,0 V zařízení začne odebírat proud, který zařadí zařízení do jedné ze tří výkonových kategorií dle tabulky 3.3.
- Po úspěšné detekci a klasifikaci dojde k připojení napájení 48 V_{nom} na port PSE.

Tímto postupem je zajištěno, že nemůže dojít k poškození zařízení, které není připraveno na připojení k PSE.

Pro vývoj byl použit PoE injektor TP-Link TL-POE150S, který je osazen obvodem LTC4263.

⁶angl. socket, jde o koncový bod komunikace v počítačových sítích

⁷Fast Ethernet

⁸Gigabit Ethernet

Tab. 3.3: Třídy PD dle IEEE 802.3af

Třída zařízení	Maximální požadovaný výkon [W]	$I_{min}[mA]$	$I_{max}[mA]$
0	15,4	0	5
1	4,0	8	13
2	7,0	16	21
3	15,4	21	31
4	Rezervováno	35	45

Tab. 3.4: Úbytky napětí pro různé výkony na napětí pro CAT5e kabel. (délka $l = 15$ m, $\rho = 0,084 \Omega m^{-1}$)

		$P[W]$				
		1,000	2,000	5,000	10,000	15,000
$U_{out}[V]$	3,3	0,382	0,764	1,909	3,818	5,727
	5	0,252	0,504	1,260	2,520	3,780
	12	0,105	0,210	0,525	1,050	1,575
	24	0,052	0,105	0,262	0,525	0,788
	48	0,026	0,052	0,131	0,262	0,394

3.3.2 Implementace PoE

Implementaci PoE interface lze vidět na schématu A.4.

Pro fázi detekce je využit odpor R501, který je trvale připojen mezi napájení.

Pro fázi klasifikace je zapojen proudový zdroj (tranzistor Q501, napěťová reference D512), který při napětí vyšším než cca 12,3 V odebírá 17 mA, což je nastaveno pomocí odporů R505, R509, R510. Toto by při plném napětí 48 V znamenalo zbytečnou ztrátu 0,8 W, proto je obvod doplněn tranzistorem Q503, Q504 pro odpojení tohoto proudového zdroje při napětí vyšším než 20 V.

Zařízení je připojeno k PoE při zvýšení napětí nad 20 V pomocí tranzistoru Q502. Pro snížení napětí na vhodnějších 5 V je využit DC/DC měnič od firmy Mean Well SCW05C-05, ve schématu U501. Tímto měničem je také zajištěno galvanické oddělení od PoE injektoru.

Pro napájení mikrokontroléru je ještě zařazen spínaný stabilizátor na 3,3 V. Napájení jádra mikrokontroléru a obvodu WizNet (1,8 V) je řešeno jejich vlastními vestavěnými lineárními regulátory.

3.4 Zobrazení

Pro zobrazení byl zvolen šestimístný displej složený z digitronů Z570M. Tyto digitrony vyžadují napájecí napětí alespoň 170 V, kterého je dosaženo zvyšujícím DC/DC měničem dle schématu A.7.

Step-up měnič zvyšuje napětí z 5 V na cca 80 V a následně je zapojen násobič dvěma. Tím je dosaženo požadované výstupní napětí. Měnič je osazen obvodem MC34063 využitým pouze jako kontrolér spínaného zdroje. Výkonový tranzistor je MOS-FET typu N umožňující řízení velmi malým napětím (logic-level mosfet), konkrétně jde o typ IRL540NS.

Samotné řízení digitronů je provedeno multiplexem s využitím tranzistorů SMBTA42 (NPN $U_{CE0} = U_{CB0} = 300$ V) a SMBTA92 (PNP, stejné parametry). Obvodové zapojení je vidět na schématu A.2 Frekvence multiplexu byla zvolena na 760 Hz. Při přechodu mezi jednotlivými digitrony je přidána pauza $30 \mu\text{s}$ ($10 \mu\text{s}$ po vypnutí spodního tranzistoru + $10 \mu\text{s}$ mezi přepnutím horních tranzistorů + $10 \mu\text{s}$ před sepnutím spodního tranzistoru), aby bylo omezeno parazitické svícení číslic v sousedních digitronech.

4 REALIZACE SOFTWARE

Program pro mikrokontrolér je psaný v jazyce C. Pro překlad je využit překladač GCC dodávaný s programem Atmel Studio 6.1. K vývoji nebyly použity Arduino knihovny ani knihovny ASF¹, ale byly využity hlavičkové soubory pro daný mikrokontrolér.

4.1 Periferie mikrokontroléru

Pro přístup k konfiguračním registrům mikrokontroléru byly využity hlavičkové soubory dodávané firmou Atmel. Tyto hlavičkové soubory definují struktury pro přístup k registrům pro konkrétní periferii. Tento přístup značně zpřehledňuje výsledný kód programu.

Přístup ke konfiguračním slovům je rozdělen na tři registry - Enable(Set) Register, Disable(Clear) Register a Status Register. První dva jsou pouze pro zápis a nastaví danou hodnotu na 1 (Enable/Set) nebo 0 (Disable/Clear). Status Register slouží k přečtení aktuální hodnoty.

4.1.1 PMC

PMC² je blok mikrokontroléru, který řídí povolení systémového taktu jednotlivých periférií. Pro základní použití jsou nejdůležitější registry (PCER, PCDR, PCSR), pomocí kterých můžeme povolit systémový takt jednotlivým perifériím.

4.1.2 PIO

PIO³ jsou bloky, které řídí výstupní paralelní porty. Pro každý pin lze nastavit, jestli na něm bude povolena periferie, nebo se použije jako obyčejný I/O pin. Nastavení se provede pomocí registrů (PER, PDR, PSR). Lze také ovládat interní pull-up rezistor (PUER, PUDR, PUSR). Pro zápis na port se použijí registry (SODR, CODR, ODSR), číst data na portu lze pomocí PDSR.

Každý pin také může mít povoleno přerušení při změně hodnoty (IER, IDR, ISR). Po změně hodnoty se vyvolá přerušení pro port daného pinu a následně je třeba zkontrolovat, z jakého pinu dané přerušení přišlo.

¹angl. Atmel Software Framework - knihovna pro „usnadnění“ vývoje firmwaru a portace na jiné mikrokontrolry z rodiny Atmel ARM

²angl. Power Management Controller

³angl. Parallel Input/Output

4.1.3 Sběrnice SPI

Mikrokontrolér má k dispozici jeden řadič SPI sběrnice, na které umožňuje komunikovat až s patnácti zařízeními. SPI řadič lze také využívat v roli *slave*.

Pro nastavení SPI rozhraní jsou určeny registry (CR, MR, CSR). Pro samotnou komunikaci jsou určeny registry (RDR, TDR), stav SPI rozhraní je k dispozici v registru SR.

Řadič podporuje až 16bitové přenosové slovo (transfer) a umožňuje nastavovat pauzy v přenosu. Maximální pauza mezi dvěma přenosy, ve které ještě nedojde ke zrušení výběru čipu (přechod CS do úrovně H), je označena jako DLYBCT. Pauza mezi vybráním čipu a začátkem hodinového signálu SPCK je DLYBS. Existuje taktéž pauza při změně vybrané *slave* periferie označovaná jako DLYBCS.

4.1.4 Rozhraní UART

Pro usnadnění vývoje je u zařízení využito sériové rozhraní UART, na kterém probíhají ladící výpisy. Jako velmi užitečné se ukázalo odchyťávat Hard Fault⁴ přerušení a na základě obsahu registrů a zásobníku dohledat přesné místo v kódu, kde k chybě došlo.

Pro nastavení rozhraní jsou k dispozici registry (MR, CR, BRGR). Pro komunikaci využijeme (THR, RHR) a taktéž SR.

4.1.5 Časovač/čítač

Mikrokontrolér má k dispozici šest kanálů 16bitového časovače, které jsou nezávisle konfigurovatelné. Pro použití v hodinách jsou použité časovače provozování v módu WAVE.

Pro udržení časové frekvence je využit 16bitový časovač. Jeho doba přetečení je přibližně nastavena na 10 μ s (není potřeba nastavovat přesně, protože je tato hodnota kompenzována dodatečně při synchronizaci). Při každém přetečení se přičte odpovídající časový údaj do počítadla času. Tento přírůstek se během synchronizace koriguje, až je dosažena správná hodinová frekvence pro 1 s.

4.2 WizNet

Pro usnadnění práce byl napsán driver pro práci s obvodem W5100. Tento driver se skládá ze dvou částí.

⁴přerušení při kritické chybě programu

- Nízkoúrovňová implementace, která řeší přístup k SPI sběrnici a zápis hodnot do paměťového prostoru obvodu W5100. Komunikace po SPI sběrnici probíhá po 16bitových slovech. To zajišťuje menší pauzy mezi jednotlivými přenesenými bajty.
- Vysokoúrovňová implementace, která řeší přístup k obvodu W5100 na úrovni soketu, bez nutnosti řešit rozdělení adresního prostoru obvodu W5100.

Toto rozdělení bylo velmi výhodné pro implementaci dalších protokolů.

4.3 DHCP klient

DHCP slouží k samonastavení IP adresy zařízení. Protokol DHCP staví nad starším protokolem BOOTP, jehož rozšířením protokol DHCP je. Komunikace probíhá pomocí UDP protokolu na portu 67 a 68. Většinu času se komunikuje pomocí *broadcast* paketů.

DHCP rozlišuje pět základních typů paketů:

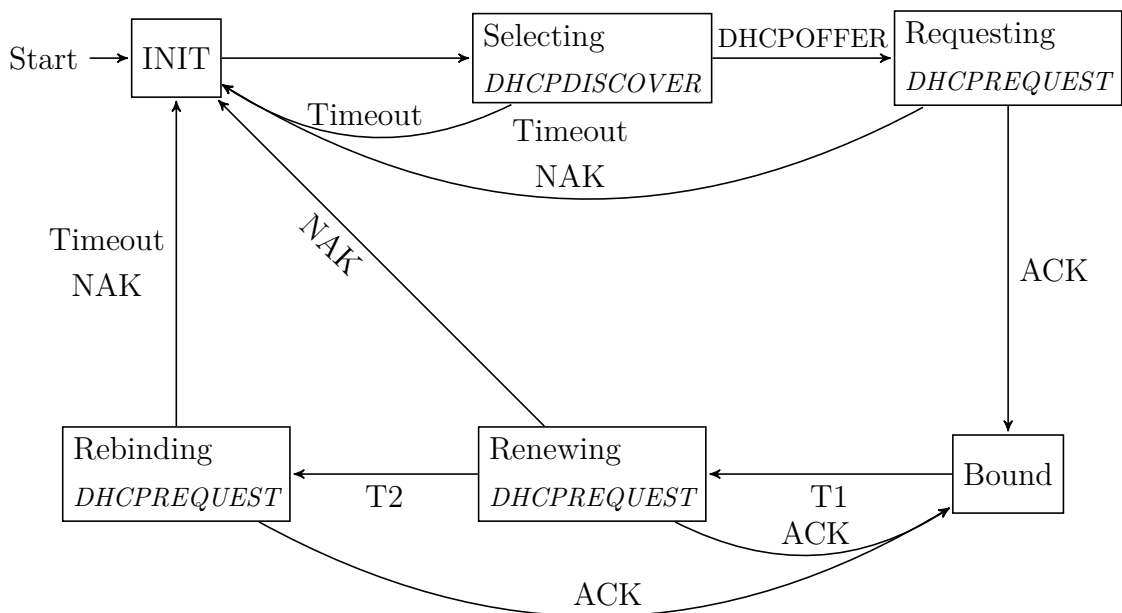
- DHCPDISCOVER, klient zjišťuje existenci DHCP serverů a žádá je o nabídku síťové adresy
- DHCPOFFER, DHCP server klientu nabízí síťovou adresu
- DHCPREQUEST, klient si žádá o přidělení nabízené adresy
- DHCPACK, DHCP server potvrzuje přidělení adresy klientovi
- DHCPNAK, DHCP server ruší (odmítá) přidělení adresy klientovi

S přidělenou adresou DHCP server taktéž určí, na jak dlouho je adresa přidělena a klient si po uplynutí této doby (ve stavovém diagramu označena jako T1) požádá o adresu znovu. Pokud mu jeho DHCP server neodpoví, tak se po uplynutí doby T2 ptá všech DHCP serverů v síti (pakety posílá jako *broadcast*).

Pro implementaci je využit stavový automat dle obrázku 4.1. Při implementaci je třeba také myslet na stav, kdy klient dostane více paketů DHCPOFFER nebo DHCPACK od různých DHCP serverů. Standard v tomto případě nepředepisuje žádný výběr adres. V navržené implementaci se vždy využije 1. příchozí paket a další se ignorují.

4.4 SNTP klient

Pro hodiny byl napsán zjednodušený klient, jelikož jeho dosahovaná přesnost postačuje k běžnému použití. Klient pracuje proti jedinému serveru, jehož adresa je trvale uložena v paměti programu. Po inicializaci funguje dle stavového automatu na obrázku 4.2 (stavový automat je popsán v angličtině z důvodu korespondence s popisem v kódu implementace). Takto navržený klient dokáže vyřešit základní



Obr. 4.1: Stavový automat DHCP klienta

chybové stavy a v případě závažné chyby, ze které se nedokáže zotavit, neposílá zbytečně další požadavky na server.

Časový interval pro synchronizaci byl zvolen na 2 hod. To znamená maximální chybu 143 ms.

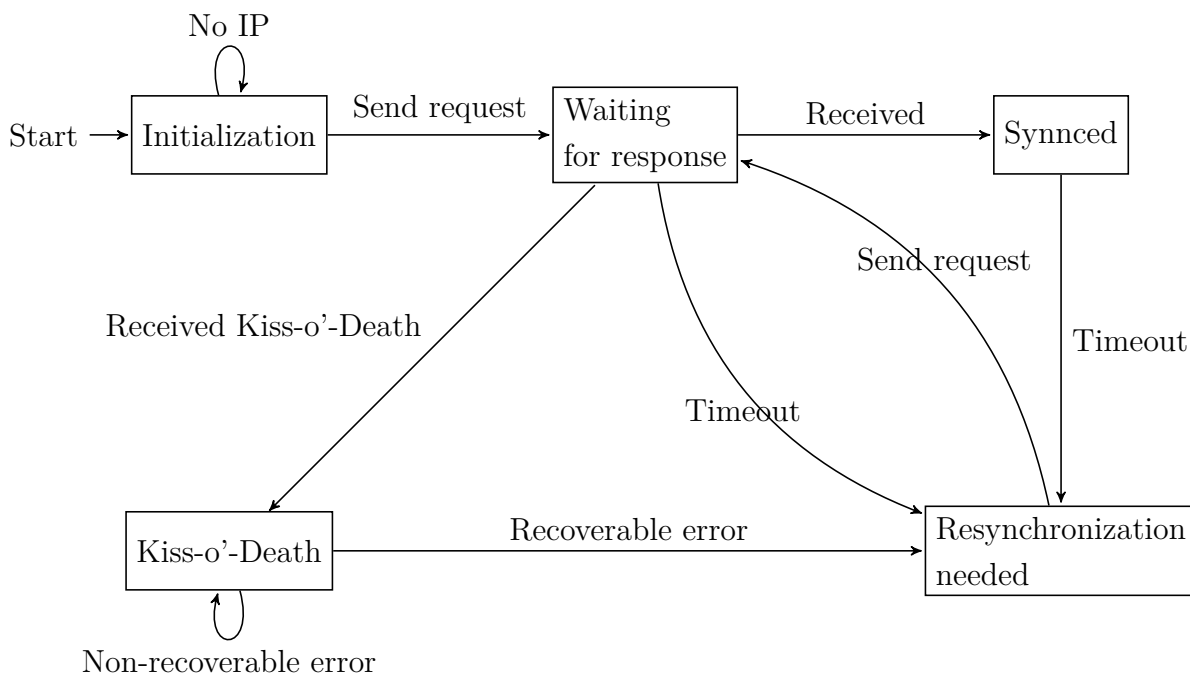
Při začátku synchronizace je vhodné provést dva požadavky rychleji po sobě. To umožní rychlejší korekci frekvence hodin.

4.4.1 Zpracování odpovědi

Dle přijaté odpovědi se provádí dvojí korekce hodin. Jako první se provede korekce časového údaje o rozdíl θ zjištěný dle vzorce 2.1. V případě malé chyby (menší než 1 min) se provede i korekce časového oscilátoru upravením hodnoty přírůstku pro časovač/čítač. Tyto hodnoty jsou průměrovány s exponenciálním zapomínáním ($k = 1/2$). Přesný způsob zpracování je vidět na vývojovém diagramu 4.3. Zpracování Kiss-o'-Death odpovědi není detailně rozkresleno.

V případě, že jde o Kiss-o'-Death paket, algoritmus určí, o jakou chybu jde. Na nejkritičtější chyby klient reaguje ukončením synchronizace a setrváním ve stavu Kiss-o'-Death. Jde to následující chyby:

- AUTH
- AUTO
- CRYP
- DENY



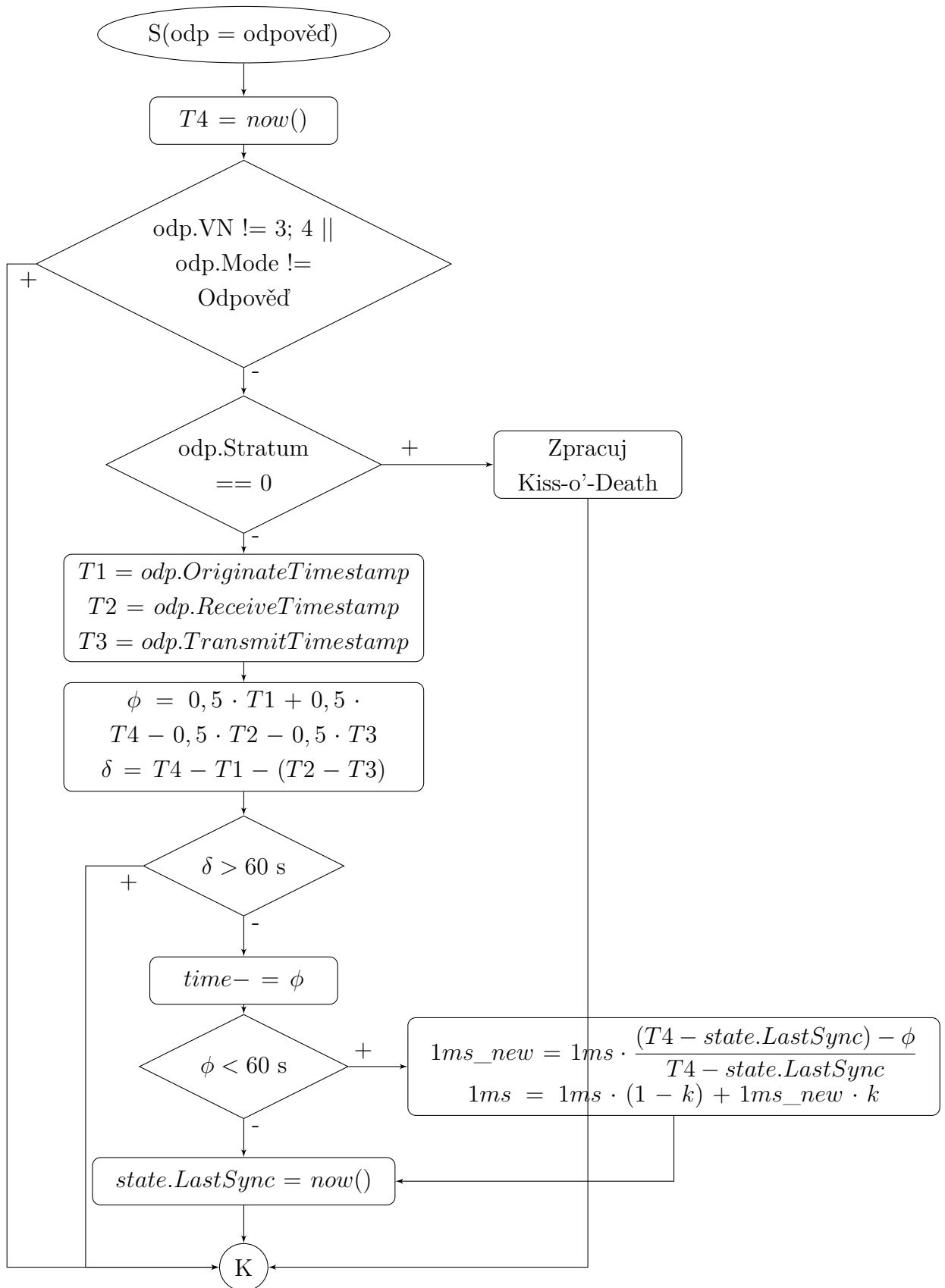
Obr. 4.2: Stavový automat SNTP klienta

- RSTR

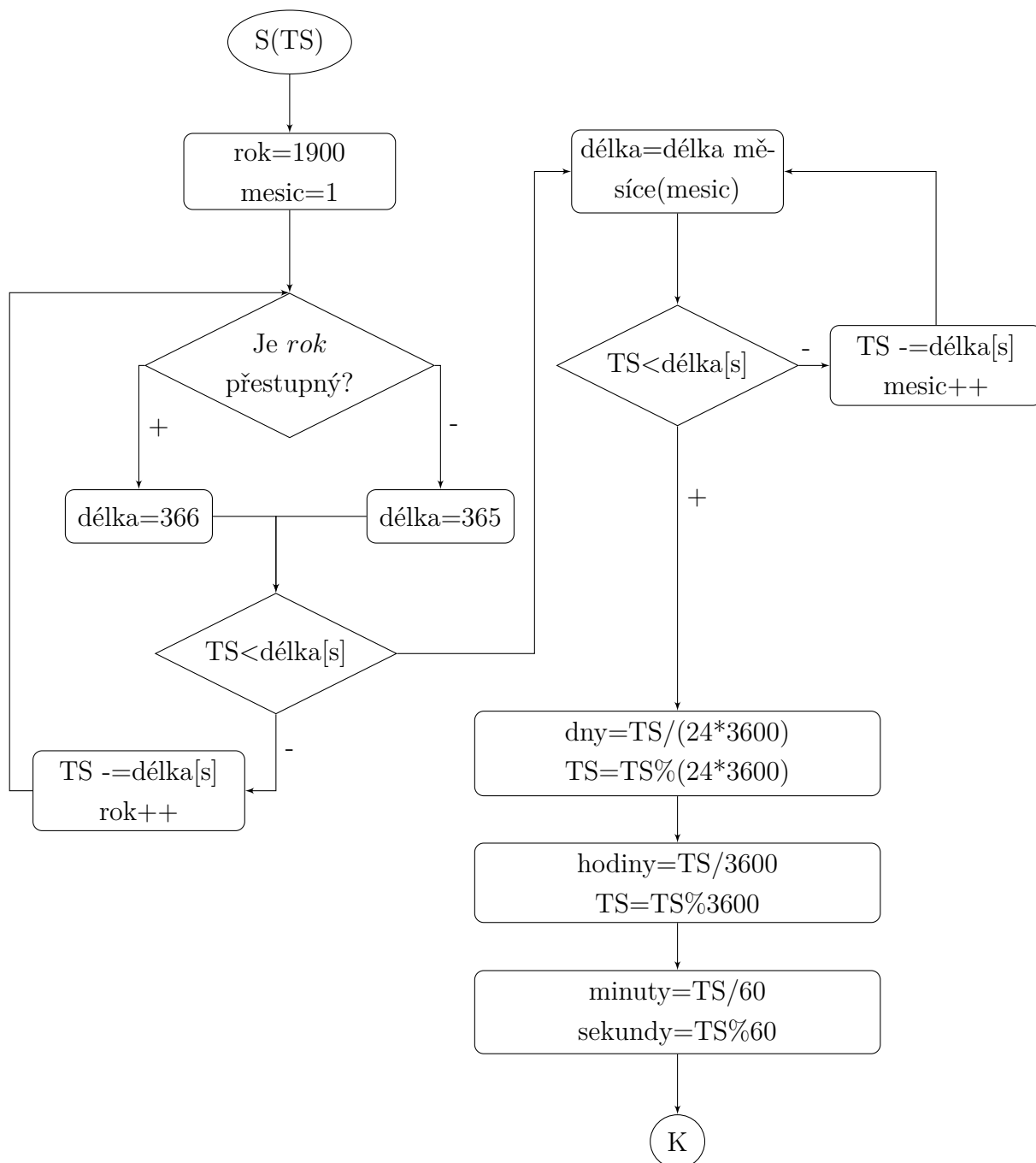
4.4.2 Převedení časového razítka

Převedení času z časového razítka (timestampu) provádím pomocí postupného odčítání času. Přesný algoritmus je vidět na vývojovém diagramu 4.4. Po tomto převodu se provede převod do lokálního času a zohlednění letního času.

Pro zobrazení na digitronovém displeji není třeba provádět převodění data a je využito jednodušší funkce.



Obr. 4.3: Zpracování odpovědi od NTP serveru



Obr. 4.4: Vývojový diagram převodu časového razítka na čas

5 VÝSLEDKY

Všechny testy probíhaly v lokální síti proti NetTime NTP serveru¹ pod operačním systémem Windows 8.1. Tento software běžel na notebooku Lenovo ThinkPad X220t. Počítačová síť byla realizována pomocí rozhraní 100BASE-T ethernet. Připojení bylo realizováno přes jeden síťový přepínač ZYXEL ES-108a - v2.

5.1 Test časové stability

Test probíhal se synchronizačním intervalem přibližně 79 s. Tento časový údaj nebyl ovšem přesně odměřován². Filtrace nastavení hodnoty přírůstku časového registru nebyla použita. Taktéž nebyla využita podmínka pro korekci hodinového přírůstku pouze při chybě menší než 1 min.

Dle rovnice 2.13 je pro synchronizační interval $T_{MAX} = 79$ s a stabilitu krystalu $\delta_f = \pm 20 \cdot 10^{-6}$ maximální teoretická chyba $\Delta = 1,6$ ms. Synchronizační algoritmus se pod tuto chybu dostal po páté synchronizaci a držel se pod ní.

Shodnost časových údajů T2 a T3 je dána použitým serverem (NetTime). Jelikož jde o OpenSource projekt, můžeme vidět v jeho zdrojovém kódu `ntptime.pas` - funkce `TNTPServerThread.Dorequest`³, že v odpovědi nastavuje `Receive Timestamp` a `Transmit Timestamp` na stejnou hodnotu.

5.2 Problém první synchronizace

Při první synchronizaci čeká NTP klienta problém synchronizovat velký rozdíl času. Pokud i v tomto případě dojde ke korekci frekvence hodinového oscilátoru, dojde ke korekci špatným směrem. To má za následek zrychlení hodinového oscilátoru a následnou velkou a zbytečnou chybu při následujících synchronizacích. Je to velice dobře vidět v tabulce 5.1 pro $N = 2 - 4$. Z tohoto důvodu je vhodné provést korekci frekvence hodin až při chybách menších než 1 min.

Tento problém také ovlivní hodnotu odezvy δ pro synchronizace $N = 2 - 3$. Zde je dobře vidět, že při synchronizaci $N = 1$ došlo k přílišnému zvýšení hodinové frekvence a při synchronizaci $N = 2$ poté došlo k opačné korekci. Tyto problémy by pomohla odstranit filtrace korekčních hodnot pro hodinový oscilátor.

¹k dispozici na <http://www.timesynctool.com/>

²bylo využito druhého časovače, který nebyl korigován dle synchronizace

³k dispozici na serveru SourceForge - <http://nettime.cvs.sourceforge.net/viewvc/nettime/NetTime/ntptime.pas?revision=1.1.1.1&view=markup>

Tab. 5.1: Test synchronizace pro $\delta = 79$ s. Nebyla použita korekce hodinového kmitočtu ani zamezení změnám kmitočtu při velkých časových rozdílech.

N	T1	T4	$ABS(\phi)[s]$
	T2	T3	$\delta[ms]$
1	1.1.1900 0:0:9.932732	1.1.1900 0:0:9.952580	–
	13.5.2015 20:31:43.669999	13.5.2015 20:31:43.669999	19,848
2	13.5.2015 21:08:34.929810	13.5.2015 21:08:35.445516	35 min 32.693663 s
	13.5.2015 20:33:02.493999	13.5.2015 20:33:02.493999	515,705
3	13.5.2015 20:33:27.986171	13.5.2015 20:33:27.991944	53,328942
	13.5.2015 20:34:21.317999	13.5.2015 20:34:21.317999	5,773
4	13.5.2015 20:35:39.398197	13.5.2015 20:35:39.416405	0,733698
	13.5.2015 20:35:40.140999	13.5.2015 20:35:40.140999	18,207
5	13.5.2015 20:36:57.986171	13.5.2015 20:36:57.991944	0,0166702
	13.5.2015 20:36:58.017999	13.5.2015 20:36:58.017999	18,379
6	13.5.2015 20:38:17.780050	13.5.2015 20:38:17.798426	0,001238
	13.5.2015 20:38:17.787999	13.5.2015 20:38:17.787999	18,375
7	13.5.2015 20:39:36.604426	13.5.2015 20:39:36.622801	0,001386
	13.5.2015 20:39:36.614999	13.5.2015 20:39:36.614999	18,374
8	13.5.2015 20:40:55.430184	13.5.2015 20:40:55.448559	0,000372
	13.5.2015 20:40:55.438999	13.5.2015 20:40:55.438999	18,375
9	13.5.2015 20:42:14.253807	13.5.2015 20:42:14.272182	0,000995
	13.5.2015 20:42:14.261999	13.5.2015 20:42:14.261999	18,375
10	13.5.2015 20:43:33.075801	13.5.2015 20:43:33.094176	0,001011
	13.5.2015 20:43:33.085999	13.5.2015 20:43:33.085999	18,374
11	13.5.2015 20:44:51.900807	13.5.2015 20:44:51.919182	0,000995
	13.5.2015 20:44:51.908999	13.5.2015 20:44:51.908999	18,375
12	13.5.2015 20:46:10.722801	13.5.2015 20:46:10.741176	0,001011
	13.5.2015 20:46:10.732999	13.5.2015 20:46:10.732999	18,374

ZÁVĚR

Cílem bakalářské práce bylo využít poznatky o synchronizaci času pomocí sítě Internet a protokolu NTP získané během práce na seminárním projektu k realizaci samostatného modulu klienta.

V rámci řešení práce bylo provedeno seznámení se s protokolem NTP, s protokolem DHCP, s principem synchronizace času a s technologií PoE (dle normy IEEE 802.3af) a následně bylo navrženo a realizováno požadované zařízení. K realizaci PoE interface byly využity diskrétní součástky a modul izolovaného snižujícího DC/DC měniče. Pro realizaci řídicí části byl využit mikrokontrolér. Při konečném hardwarovém návrhu byl namísto původního mikrokontroléru ATSAM3X8E (který byl zvolen v semestrálním projektu) použit levnější, ale svými parametry ekvivalentní, mikrokontrolér ATSAM3S2C. V závěru byla navržena deska plošných spojů, která byla zhotovena, osazena a oživena.

V programu pro mikrokontrolér byla dokončena implementace klienta SNTP protokolu, jehož přesnost je více než dostatečná k praktickému použití v digitálních hodinách. Zadání práce sice nestanovilo požadovanou přesnost, ta však byla po konzultaci s vedoucí pracem stanovena na $\Delta < 1$ s po dobu 10 hodin.

Součástí řešení byla literární rešerše, kde byly s ohledem na charakter zadání práce využity coby zdroje především standardy a doporučení (dokumenty RFC). Téma je natolik specifické, že není možné přímo vycházet z literatury, která by explicitně řešila protokol NTP. Mimo standardů a doporučení je jediným nalezeným relevantním zdrojem [9].

Během řešení realizační části práce se potvrdilo, že ne všechny implementace NTP serveru pracují přesně dle očekávání a čas přijetí požadavku a čas odeslání odpovědi považují za stejný. Implementaci SNTP klienta to sice nevadí, avšak tato situace zkresluje údaj o době odezvy sítě. V rámci kapitoly Výsledky byly popsány dva zásadní problémy, které vyvstaly při řešení zadání, a sice problém časové stability a problém první synchronizace. Tyto problémy byly zcela vyřešeny.

Domnívám se, že zadané cíle práce byly dosaženy a sestavené zařízení tak nabízí plně použitelné samostatné digitální hodiny řízené ze sítě Internet.

LITERATURA

- [1] Marzullo K.; Owicki S. *Maintaining the Time in a Distributed System*[online]. Srpen 1983. Dostupné z URL: <<http://infolab.stanford.edu/pub/cstr/reports/csl/tr/83/247/CSL-TR-83-247.pdf>>
- [2] Organisation Intergouvernementale de la Convention du Mètre *The International System of Units (SI)*[online]. 2006. Dostupné z URL: <http://www.bipm.org/utils/common/pdf/si_brochure_8.pdf>
- [3] Phys.org *Accuracy of the NPL caesium fountain clock further improved*[online]. Únor 2014. Dostupné z URL: <<http://phys.org/news/2014-02-accuracy-npl-caesium-fountain-clock.html>>
- [4] Poupa M. *Vše o času*[online]. 1997-2002. Dostupné z URL: <<http://home.zcu.cz/~poupa/dcf77.html>>
- [5] RFC868 *Time protocol*[online]. Dostupné z URL: <<http://tools.ietf.org/html/rfc868>>
- [6] RFC4330 *Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI* [online].. Leden 2006. Dostupné z URL: <<https://tools.ietf.org/html/rfc4330>>.
- [7] RFC5905 *Network Time Protocol Version 4: Protocol and Algorithms Specification*[online]. Červen 2010. Dostupné z URL: <<https://tools.ietf.org/html/rfc5905>>
- [8] RFC5906 *Network Time Protocol Version 4: Autokey Specification*[online]. Červen 2010 Dostupné z URL: <<http://tools.ietf.org/html/rfc5906>>
- [9] MILLS, David L. *Computer network time synchronization: the Network Time Protocol* , FL: CRC/Taylor & Francis, 2006, xvii, 286 p. ISBN 9780849358050
- [10] *Service Name and Transport Protocol Port Number Registry*[online]. Dostupné z URL: <<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>> (Citováno dne. 16. 12. 2014)
- [11] The National Institute of Standards and Technology (NIST) *NIST Time Scale Data Archive* [online]. Naposledy aktualizováno: 28.11.2014 [cit. 30. 11. 2014] Dostupné z URL: <www.nist.gov/pml/div688/grp50/leapsecond.cfm>

- [12] Český metrologický instiut *Státní etalon frekvence a času* [cit. 24 05 2015]
Dostupné z URL: <<https://www.cmi.cz/statni%20etalon%20frekvence%20a%20casu>>
- [13] IEEE 802.3af-2003 *Data Terminal Equipment (DTE) Power via Media Dependent Interface (MDI)* Červen 2003 Dostupné z URL: <<http://www.lan-power.com/pdf/802.3af-2003.pdf>>

SEZNAM ZKRATEK

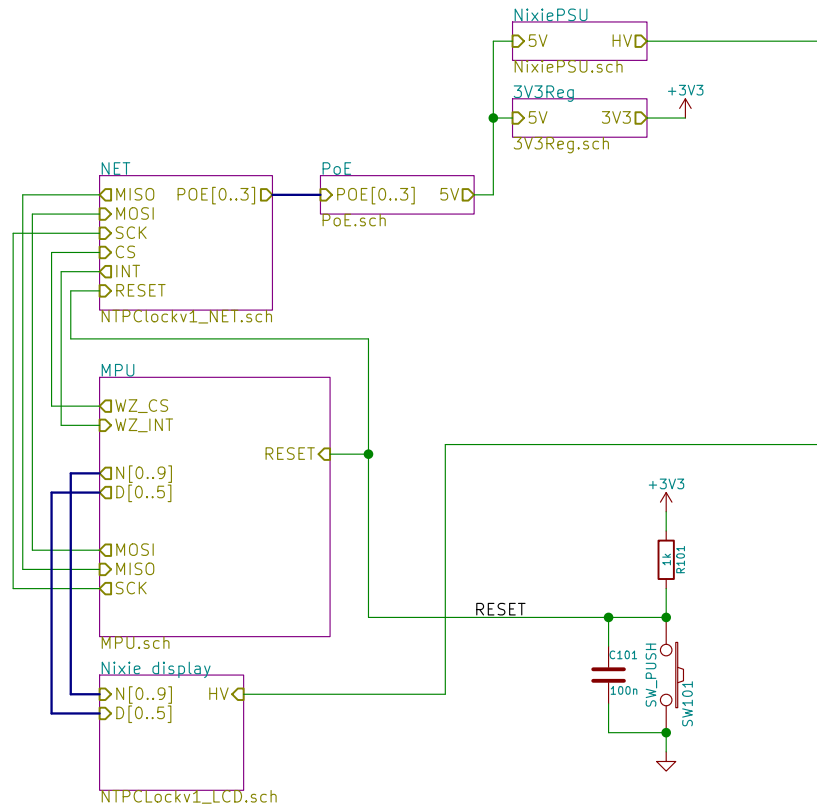
NTP	Network Time Protocol - protokol pro synchronizaci času
UTC	Universal Time Coordinated - koordinovaný světový čas založený na atomových hodinách, nástupce GMT
GMT	Greenwich Mean Time - Greenwichský střední čas - čas na nultém poledníku Země
UDP	User Datagram Protocol - protokol vrstvy L4 (dle ISO/OSI modelu); nezaručuje doručení datagramů
IANA	Internet Assigned Numbers Authority - organizace přidělující adresy a čísla pro komunikaci v Internetu
ČMI	Český metrologický institut

SEZNAM PŘÍLOH

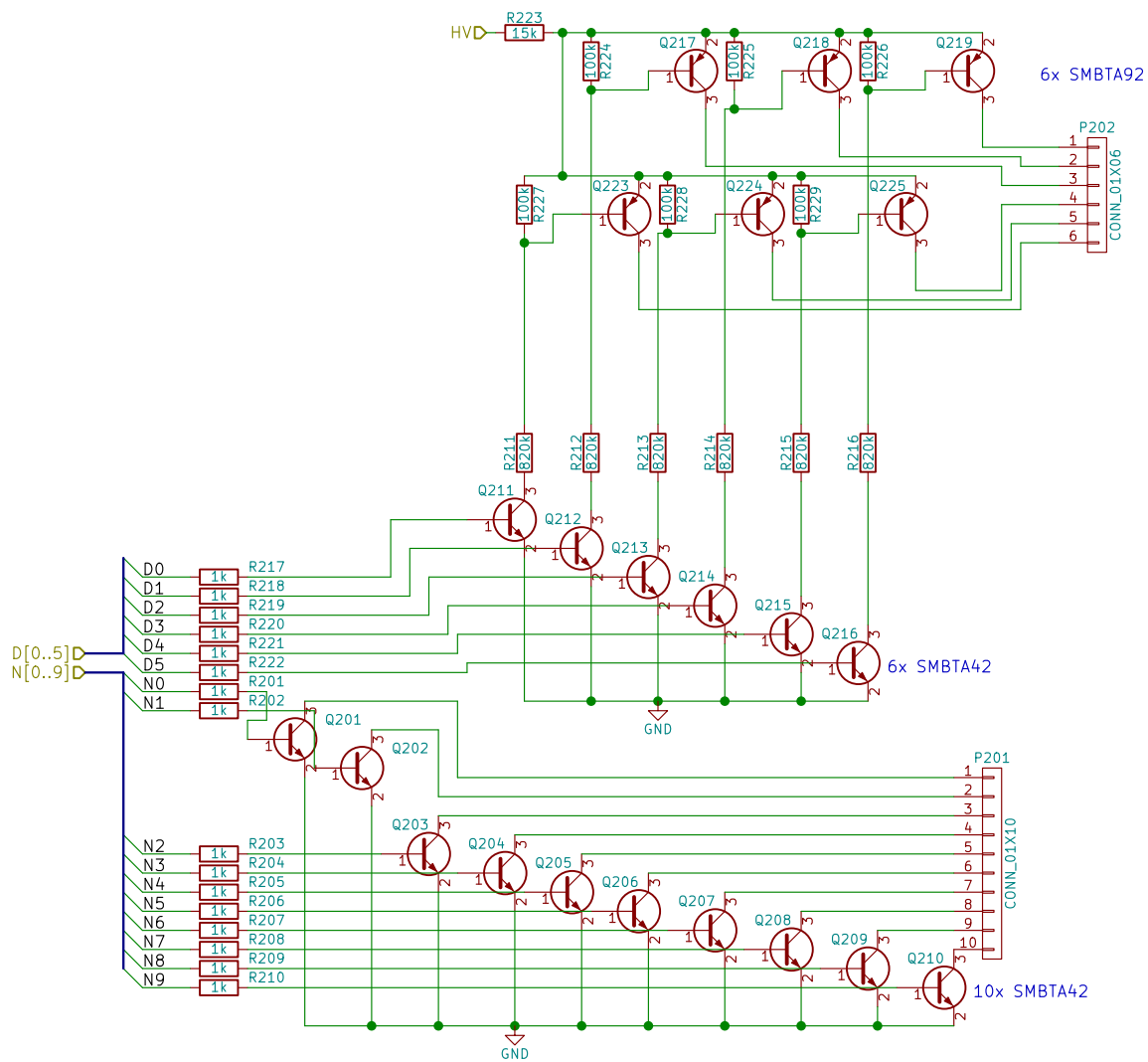
A Schéma zapojení	46
B Obsah přiloženého CD	55

A SCHÉMA ZAPOJENÍ

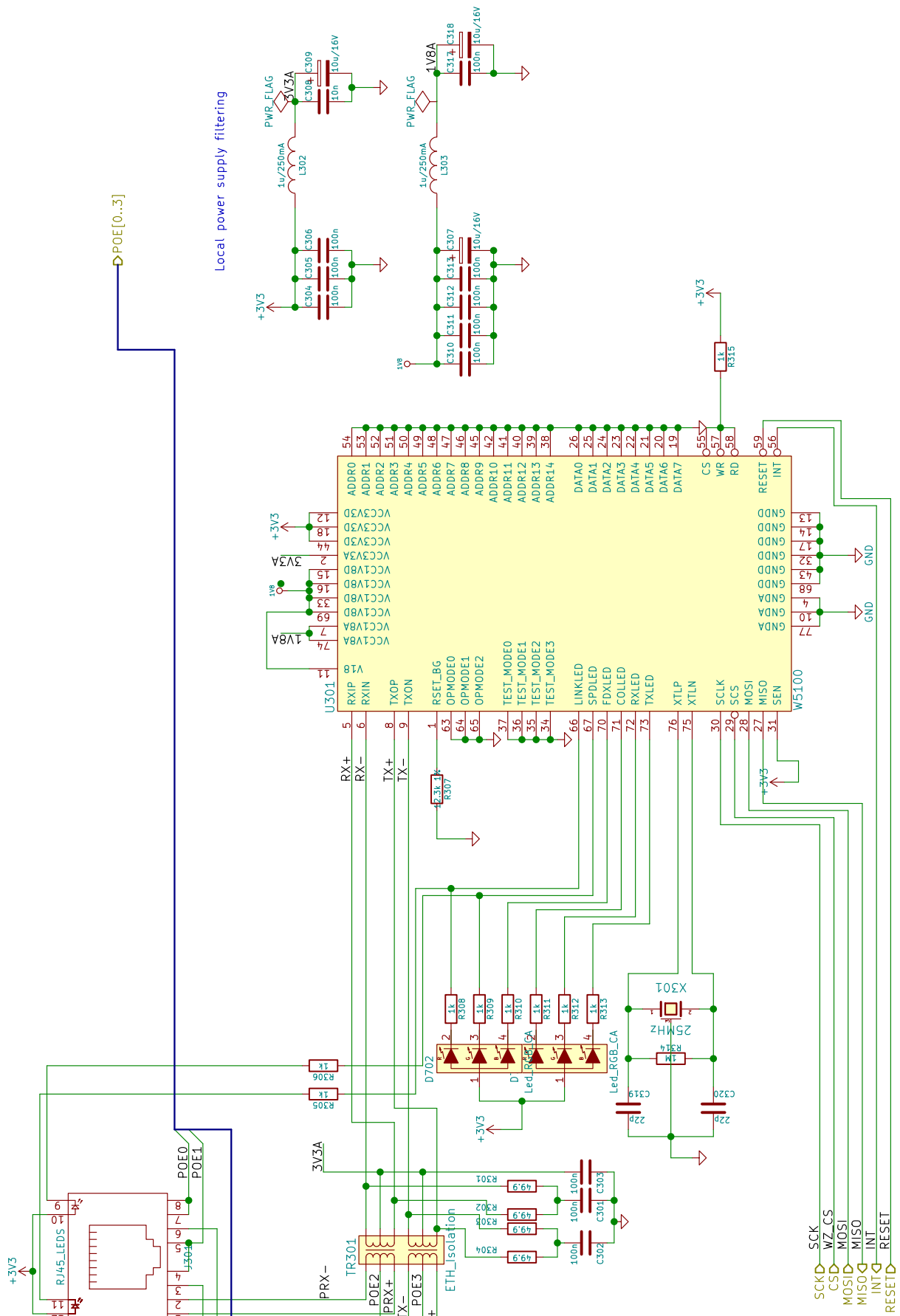
Pozn.: Schémata jsou zakreslena pomocí programu KiCAD.



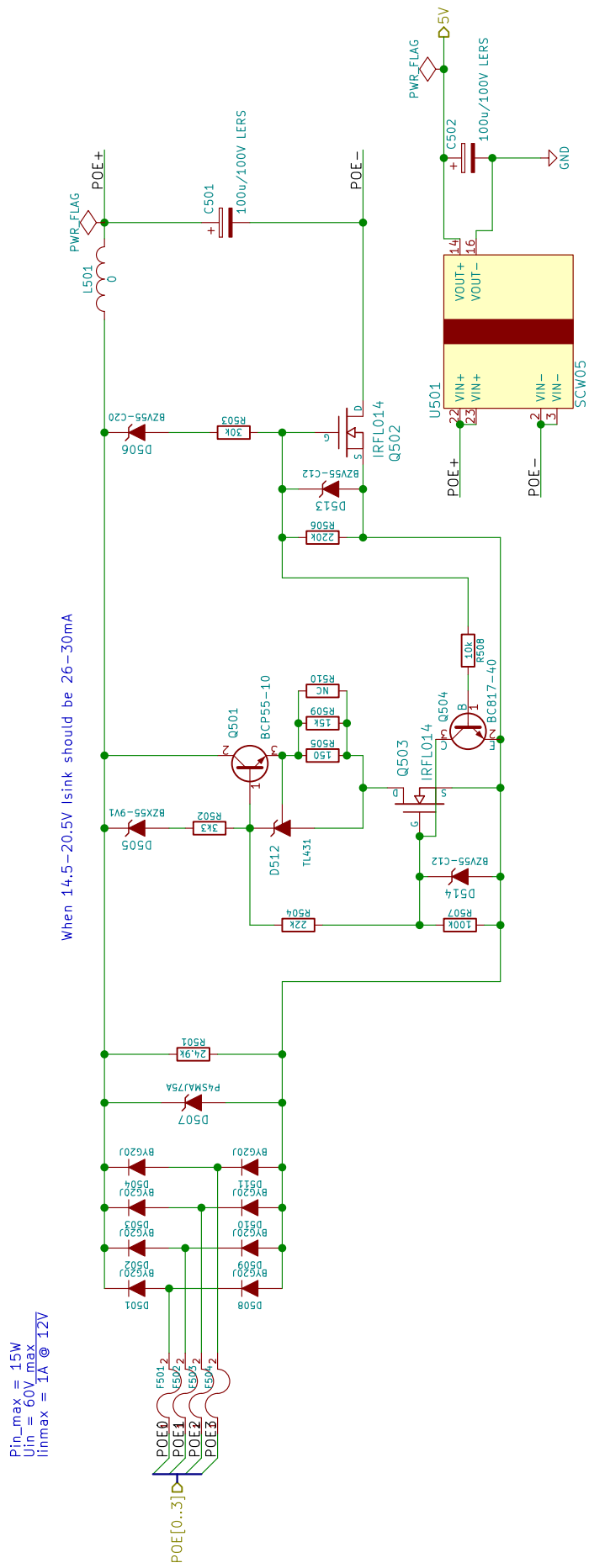
Obr. A.1: Rozdělení schématu na bloky



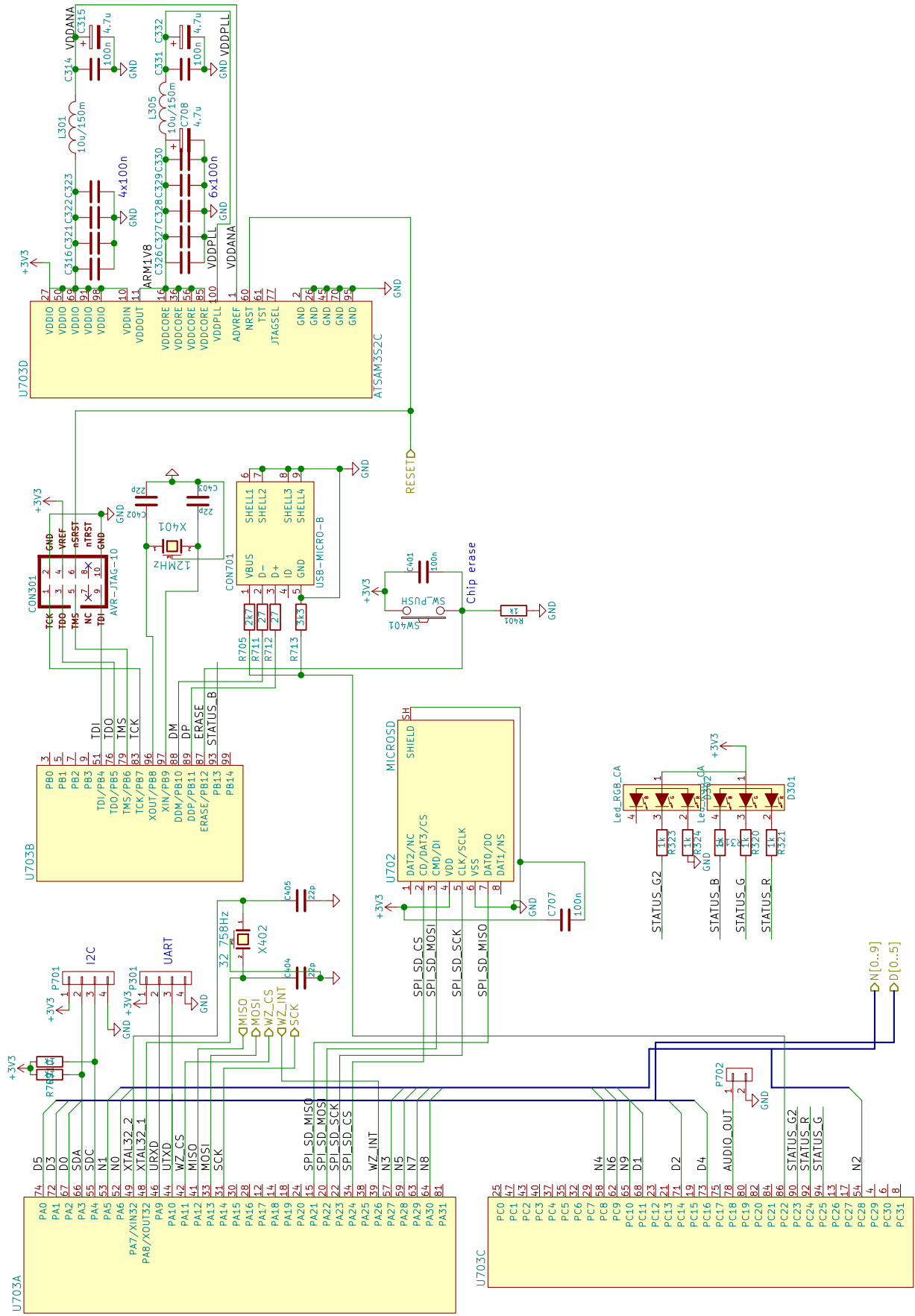
Obr. A.2: Schéma multiplexu pro digitronový displej



Obr. A.3: Schéma síťového rozhraní s čipem WizNet W5100



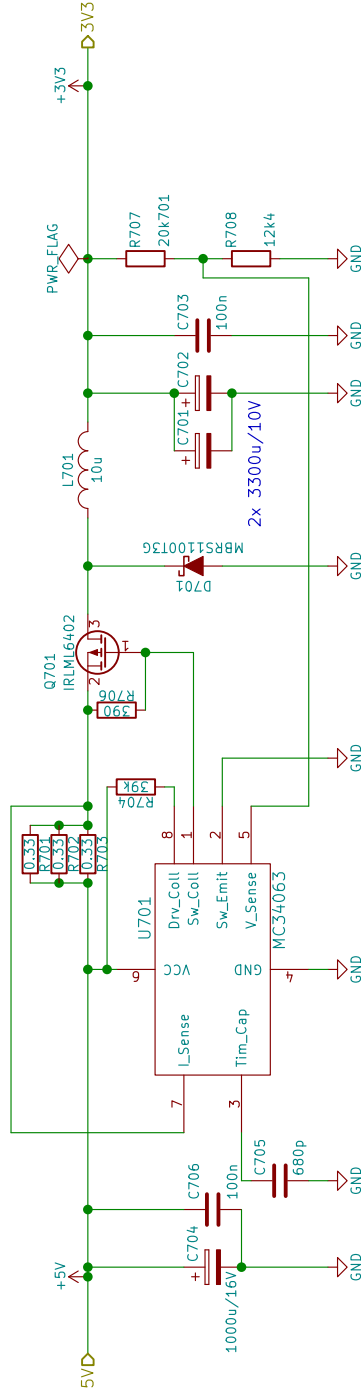
Obr. A.4: Schéma PoE interface



Obr. A.5: Schéma zapojení ARM procesoru

$$\begin{aligned}
 V_{out} &= 3.3V & V_{in} &= 5V \\
 I_{out} &= 1.5A & V_d &= 0.5V \\
 P_{out} &= 4.95W & V_{tr} &= 0.2V \\
 t_{on}/t_{off} &= \frac{(V_{out} + V_d)}{3.3 + 0.5} \cdot \frac{1}{4.5 - 0.1} = \frac{3.8}{3.3} = 1.1 = 3.5 \\
 (t_{on} + t_{off}) &= 1/f = 1/50k = 20\mu s \\
 t_{off} &= (t_{on} + t_{off}) / (t_{on}/t_{off}) + 1 = 20\mu s / 4.5 = 4.4 \mu s \\
 t_{off} &= (t_{on} + t_{off}) - t_{off} = 20\mu s - 4.4\mu s = 15.6 \mu s \\
 C_t &= 40\mu \cdot t_{on} = 40\mu \cdot 15.6\mu = 624 \text{ pF} \\
 L_{pk}(sw) &= 2 \cdot I_{out} = 1.5 \cdot 2 = 3 \text{ A} \\
 R_{sc} &= 0.3 / L_{pk}(sw) = 0.3 / 3 = 0.1 \text{ Ohm} \\
 L_{min} &= (V_{in} - V_{tr} - V_{out}) / \frac{L_{pk}(sw)}{3} = \frac{15.6 \mu}{3} = 7.8\mu H
 \end{aligned}$$

$$\begin{aligned}
 P_{in}(approx) &= L_{pk}(sw) / 2 \cdot t_{on} / (t_{off} + t_{on}) \cdot U_{in} = 3 / 2 \cdot 15.6 / 20 \cdot 5 = 5.85 \\
 i &= P_{out} / P_{in}(approx) = 4.95 / 5.85 = 85\%
 \end{aligned}$$

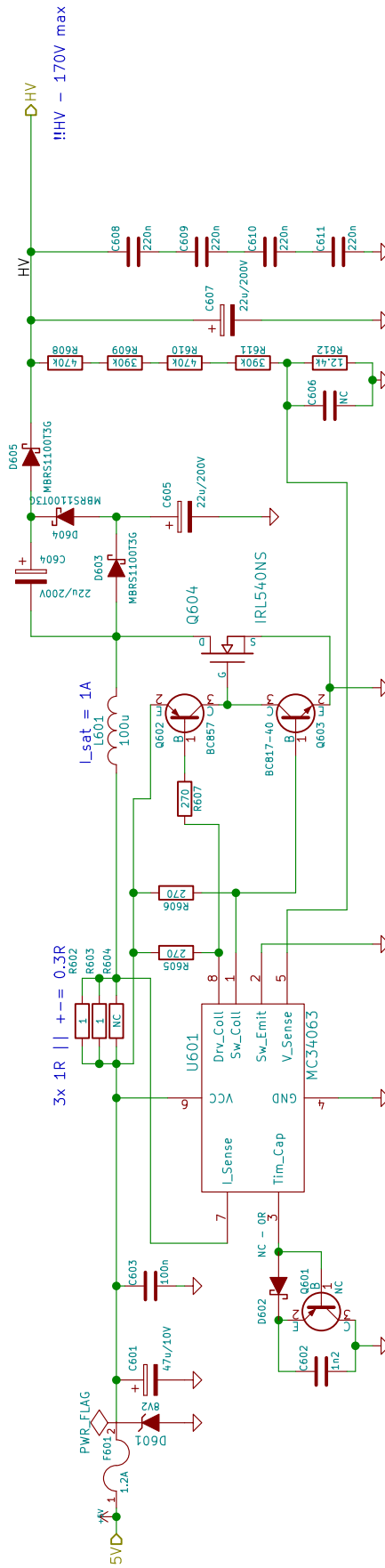


Obr. A.6: Schéma 3V3 stabilizátoru včetně výpočtů hodnot

$$\begin{aligned}
 V_{out} &= 190 / 2 = 95V & V_f &= 0.7V & f &= 50kHz \\
 V_{in(min)} &= 4.5V & V_{sat} &= 0.5V & I_{out(max)} &= 20 \text{ mA} \\
 t_{on}/t_{off} &= V_{out} + V_f - V_{in(min)} / V_{in(min)} - V_{sat} \\
 &= 95 + 0.7 - 4.5 / 4.5 - 0.5 = 91.2 / 4 = 22.8 \\
 (t_{on} + t_{off}) &= 1/f = 1/50\,000 = 20\mu s \\
 t_{off} &= (t_{on} + t_{off}) / (t_{on}/t_{off}) + 1 \\
 &= 20\mu / 23.8 = 0.84\mu s \\
 t_{on} &= (t_{on} + t_{off}) - t_{off} = 20 - 0.84 = 19.16 \mu s \\
 I_{pk(sw)} &= 2 * I_{out(max)} * ((t_{on}/t_{off}) + 1) \\
 &= 2 * 20 \text{ mA} * 23.8 = 952 \text{ mA} \\
 C_t &= 40\mu * t_{on} = 40\mu * 19.16\mu = 766.4 \text{ pF}
 \end{aligned}$$

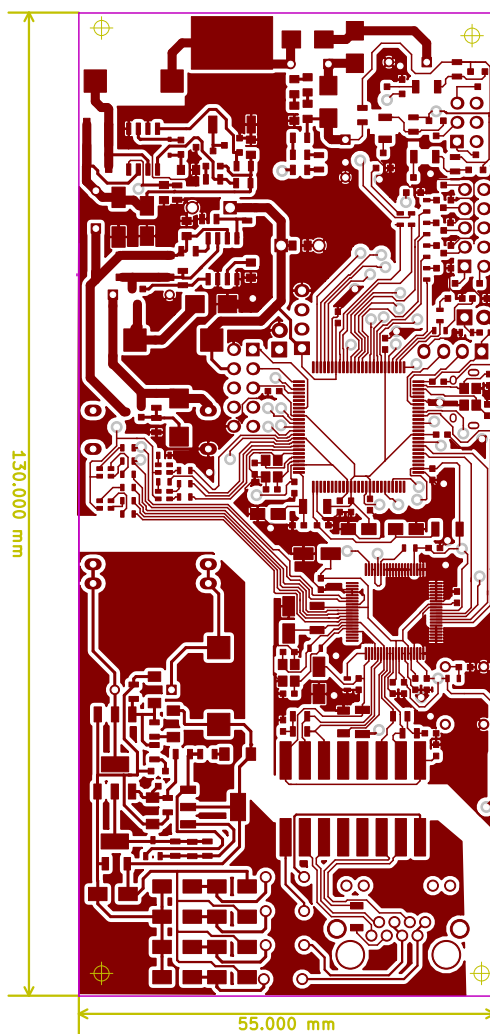
$$\begin{aligned}
 R_{sc} &= 0.3 / I_{pk(sw)} = 0.3 / 0.952 = 0.3 \text{ (Ohm)} \\
 L_{(min)} &= ((V_{in(min)} - V_{sat}) / I_{pk(sw)}) * t_{on(max)} \\
 &= ((4.5 - 0.5) / 0.952) * 19.16\mu s = 4.202 * 19.16 \mu s = 80.5 \mu H \\
 C_{Lo} &= 9 * I_{out} * t_{on} / V_{ripple(pp)} = \\
 &= 9 * 20 \text{ mA} * 19.16\mu / 3 = 1149 \text{ nF} \\
 P_{in} &= t_{on} / (t_{on} + t_{off}) * I_{pk(sw)} * 0.5 * U_{in} \\
 &= 19.16 / 20 * 0.952 * 0.5 * 5 = 0.456 * 5 = 2.2 \text{ W} \\
 P_{out} &= V_{out} * I_{out(max)} = 95 * 20 \text{ mA} = 1.9 \text{ W} \\
 n &= P_{out} / P_{in} = 1.9 / 2.2 = 0.86
 \end{aligned}$$

C604,C605,C607 MUST be low ESR – ex. Nippon chemicon KXG series

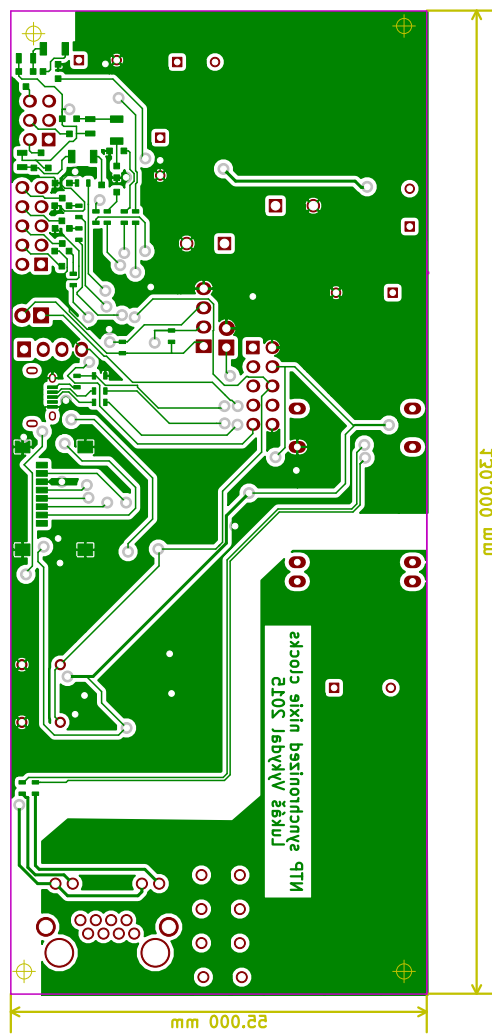


Q604 MUST be Logic Level MOSFET

Obr. A.7: Schéma měniče pro napájení digitronů včetně výpočtů hodnot



Obr. A.8: Deska pločného spoje, vrchní strana



Obr. A.9: Deska plošného spoje, spodní strana

B OBSAH PŘILOŽENÉHO CD

Přiložené CD obsahuje:

- Celkový text této práce ve formátu PDF a zdrojové soubory(formát \LaTeX)
- Zdrojový kód firmwaru pro mikroprocesor v jazyce C
- Návrh hardwaru ve formátu pro program KiCAD včetně návrhu DPS