

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra mikroelektroniky

Časový normál pro centralizovanou správu elektronických hodin

Martin Pilík

Komunikace, Multimédia a Elektronika

Leden 2018

Vedoucí práce: Ing. Vladimír Janíček, Ph.D.



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Pilík** Jméno: **Martin** Osobní číslo: **434898**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra mikroelektroniky**
Studijní program: **Komunikace, multimédia a elektronika**
Studijní obor: **Aplikovaná elektronika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Časový normál pro centralizovanou správu elektronických hodin

Název bakalářské práce anglicky:

Time Standard for Centralized Control of Distant Clock System

Pokyny pro vypracování:

- 1) Seznamte se se standardy dálkového řízení elektronických hodin.
- 2) Navrhněte koncept řídicí jednotky pro obsluhu hodinové linky.
- 3) Pro aktualizaci času využijte služeb DCF, GPS, NTP. Rešte zálohu stavu hodin a jejich nouzový provoz při výpadku napájení.
- 4) Zařízení realizujte a ověřte jeho vlastnosti.

Seznam doporučené literatury:

- 1) Remote Clock Calibration Via GPS, Lukac C.F., ASIN: B00CJJCO7C
- 2) Expert Network Time Protocol: An Experience in Time with NTP (Expert's Voice), Rybaczyk P., ISBN-13: 978-1590594841

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Vladimír Janiček, Ph.D., katedra mikroelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **15.02.2017** Termín odevzdání bakalářské práce: **26.05.2017**

Platnost zadání bakalářské práce: **10.09.2018**

Ing. Vladimír Janiček, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování / Prohlášení

Na tomto místě bych rád poděkoval zejména vedoucímu své práce panu Ing. Vladimíru Janíčkoví, Ph.D. za trpělivost a ochotu při vedení mé bakalářské práce. Bez jeho cenných rad a znalostí by tato práce nikdy nemohla vzniknout. Dále bych chtěl poděkovat firmě ELTRON s.r.o. za finanční podporu projektu a technické zázemí, které jsem mohl využívat.

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

V Praze dne

.....

Abstrakt / Abstract

Práce se zabývá návrhem a následnou realizací centrální hodinové ústředny, která po zisku časového normálu vzdáleně řídí elektronické hodiny. Pro co možná nejširší možnosti aplikací jsou k zisku časového normálu využity tři nezávislé zdroje (DCF77, GPS a NTP server). Pro přesné řízení elektronických hodin jsou jako výstup použity tři nezávislé linky.

Pro realizaci je využit jednodeskový počítač BeagleBone Black, k němuž je připojena navržená deska plošných spojů, což společně tvoří hardware ústředny. Pro konkrétní aplikaci je hodinová ústředna vybavena potřebným softwarem zvoleným podle přání koncového uživatele.

Klíčová slova: NTP; DCF; GPS; časový normál; elektronické hodiny; Beaglebone Black.

The thesis deals with the design and subsequent realization of a master clock system capable of controlling electronic time clock after receiving standard time. Three independent sources of the standard time (DCF77, GPS and NTP server) are used in order to provide the widest possible options for application. There are three independent two-wire lines used as an output ensuring precise control of the electronic time clock.

The core of this project is a single-board computer BeagleBone Black with the designed printed circuit board connected to it. Together they form hardware of the master clock system. For a use in a particular application, the master clock system is equipped with a software chosen by the end user.

Keywords: NTP; DCF; GPS; standard time; distant clock system; Beaglebone Black.

Obsah /

1 Úvod	1
2 Parametry hodinové ústředny	3
2.1 Řízení hodinové linky	3
2.1.1 Ovládání analogových hodin	4
2.1.2 Ovládání digitálních hodin	5
2.2 Zdroje časového normálu	6
2.2.1 DCF77	6
2.2.2 GPS	8
2.2.3 NTP	9
2.3 Komunikace s uživatelem	9
2.3.1 Dotykový displej	10
2.3.2 Webové rozhraní	10
2.4 Datové sběrnice	10
2.5 Spínaný výstup	10
2.6 Nouzový provoz	10
2.6.1 Výpadek napájení	11
2.6.2 Výpadek zdroje časového normálu	11
2.7 Možnosti využití	11
2.8 Porovnání s konkurencí	12
3 Hardware hodinové ústředny	13
3.1 Převod mezi úrovněmi napětí 3,3 V a 5 V	13
3.2 Příjem signálu DCF77	14
3.3 Příjem signálu GPS	14
3.4 Řízení hodinových linek	15
3.5 RS232	17
3.6 RS485	17
3.7 Spínání relé	17
3.8 DS3232	17
3.9 Signalizační LED	18
3.10 Napájení	18
3.11 Displej, konektory	18
3.12 Propojení desky plošných spojů a BeagleBone Black	19
4 BeagleBone Black	20
4.1 Parametry	20
4.2 Srovnání s podobnými produkty	21
4.3 Operační systém	22
4.4 Nastavení	22
4.4.1 Instalace operačního systému	22
4.4.2 Konfigurace operačního systému	23
4.4.3 Konfigurace GPIO a jiných periférií	23
5 Software hodinové ústředny	25
5.1 Synchronizace času	25
5.1.1 Další nastavení BeagleBone Black	25
5.2 Obsluha hodinové linky	26
5.2.1 Ovládání pinů GPIO z programu	27
5.2.2 Začátek programu	27
5.2.3 Řízení hodinové linky	27
5.2.4 Řízení dobíhání hodin	28
5.3 Výsledný software	29
6 Výsledný produkt	31
6.1 Volitelné konfigurace	31
6.2 Ověření funkčnosti	32
7 Závěr	33
A Schéma zapojení hodinové ústředny	35
B Ukázka desky plošných spojů ...	37
C Tabulky konfigurací funkcí konektorů P8 a P9	39
D Ukázka vytvořeného Device Tree	41
E Ukázka vytvořeného Device Tree	42
Literatura	44

Tabulky / Obrázky

4.1. Vybrané parametry Beagle-Bone Black	21
6.1. Možné konfigurace hodinové ústředny	32
2.1. Analogové hodiny, příklad	4
2.2. Digitální hodiny, příklad	6
2.3. Dosah vysílání DCF77	7
2.4. Kódovací schéma vysílání DCF77	7
2.5. Ilustrace systému GPS	8
2.6. Ukázka vysílané zprávy NMEA ..	9
3.1. Blokový diagram hodinové ústředny	13
3.2. Převod mezi úrovněmi napětí 3,3 V a 5 V	14
3.3. MAX3232 pro GPS a RS232 ..	14
3.4. LED signalizující signál GPS-PPS	15
3.5. Napájení GPS a jeho reset	15
3.6. Vnitřní zapojení integrovaného obvodu TB6568KQ	16
3.7. Zapojení obvodu pro řízení hodinových linek	16
3.8. Zapojení obvodu pro sběrnici RS485	17
3.9. Zapojení obvodu pro výstupní spínaný obvod	17
3.10. Integrovaný obvod DS3232	18
3.11. Signalizační LED	18
3.12. Propojení signálů mezi Cape a BeagleBone Black	19
3.13. Vyrobená deska plošných spojů hodinové ústředny	19
4.1. BeagleBone Black	20
4.2. Tabulka pro nastavení funkce pinů GPIO	24
5.1. Vývojový diagram softwaru	30

Kapitola 1

Úvod

Centralizovaná správa elektronických hodin je využívána na řadě míst. Bývá instalovaná v kancelářských a administrativních budovách, ve výrobních závodech, nemocnicích či v různých sportovních centrech. Své uplatnění nachází také ve školách, kde zajišťuje pravidelné školní zvonění. Nepostradatelná je na nádražích, letištích a jiných dopravních terminálech. Důvod je zřejmý. Přesný čas je pro daný prostor klíčový a neméně důležitá je jeho synchronizace v objektu.

Cílem této bakalářské práce je navrhnout a vyrobit hodinovou ústřednu pro vzdálené řízení elektronických hodin. Téma práce bylo zadáno od externí firmy ELTRON s.r.o., která se mimo jiné zabývá právě výrobou hodinových ústředen nebo digitálních elektronických hodin. Přínos pro firmu ELTRON s.r.o. spočívá v nové koncepci výrobku. Dosud se hodinové ústředny vyráběly za použití mikrokontrolérů. V této práci jsou nahrazeny moderním jednodeskovým počítačem BeagleBone Black, který otevírá nové možnosti realizace. Zařízení vytvořené v této práci je prototypem výrobku, na kterém je možné ověřit vhodnost použitého zapojení a funkčnost jednotlivých částí návrhu.

V práci je vytvářen univerzální hardwarový základ společný pro různé konfigurace zařízení zvolené podle přání zákazníka. Navržená hodinová ústředna umožňuje příjem časového normálu ze tří různých zdrojů – z radiového vysílání DCF77, pomocí jednoho z NTP serverů nebo komunikací s družicovým systémem GPS. Výstupem ústředny mohou být až tři nezávislé hodinové linky. Dále je zařízení vybaveno průmyslovými sběrnicemi RS232 a RS485 nebo spínaným výstupem. Zákazník má na výběr také v oblasti obsluhy. Zvolit si může buď ovládání pomocí dotykového displeje, nebo vzdáleně přes webové rozhraní. Cílem je vytvořit systém s co nejnižšími nároky na obsluhu, ideálně pak systém zcela autonomní. Zásadně záleží na typu použitých elektronických hodin a na konkrétním použití hodinové ústředny.

V první části práce jsou teoreticky popsány vlastnosti hodinové ústředny a její funkce. Jsou zde uvedeny základní informace o jednotlivých zdrojích časového normálu a je popsáno řízení elektronických hodin. V této části práce je také analyzován nouzový provoz zařízení po výpadku napájení nebo po výpadku jednoho ze zdrojů časového normálu. V závěru kapitoly je věnován prostor konkurenčním výrobkům a jejich porovnání s výrobkem navrhovaným v této práci.

Následující kapitola práce se věnuje hardwaru hodinové ústředny. Popisuje zapojení jednotlivých funkčních bloků a je doplněna obrazovou přílohou zobrazující detaily návrhu, konkrétní zapojení i vyrobenou a osazenou desku.

Práce se dále zaměřuje na jednodeskový počítač BeagleBone Black. Jsou zde popsány jeho základní parametry a je porovnán s podobným počítačem Raspberry Pi. Převážná část této kapitoly se zabývá konfigurací systému BeagleBone Black tak, aby vyhovoval potřebám hodinové ústředny. Nechybí ani návod ke konfiguraci pinů GPIO.

V následující kapitole je popsán software a základní způsoby programové obsluhy chodu hodinové ústředny. Je vysvětlen proces, jakým dochází k synchronizaci času mezi hodinovou ústřednou a zdrojem časového normálu.

Poslední kapitola je věnována výslednému produktu, tedy již hotové hodinové ústředně. Jsou zde shrnuty možnosti konfigurace zařízení rozdělené na napevno implementované součásti a výbavu volitelnou dle přání zákazníka. Na závěr je popsán proces, kterým proběhlo ověření funkčnosti hodinové ústředny.

Kapitola 2

Parametry hodinové ústředny

V této kapitole se dočteme obecné informace o navržené hodinové ústředně. Jsou zde popsány periferie a funkce, které ústředna může vykonávat. Dále možnosti použití a krátké porovnání s konkurencí. Důležité je pochopení, jak fungují jednotlivé typy hodin a jakým způsobem se tyto hodiny řídí.

Centralizovaný způsob řízení hodin není žádná novinka. V různých typech budov se již využívá několik desetiletí. Hodinové ústředny také vyrábí mnoho výrobců. Problémem je absence normy, která by určovala jednotný způsob, jakým se hodiny řídí. Každý výrobce tak má vlastní způsob přenosu nebo kódování přenosu informace o přesném čase. Řešení jednotného času v budovách tak bývá dodáváno jako kompletní produkt hodinové ústředny spolu s elektronickými hodinami. Způsob řízení však mívá společné prvky, které budou popsány v kapitole 2.1.

Přínos pro zadavatele práce firmu ELTRON s.r.o. je v novém konceptu realizace. Místo dříve využívaných mikrokontrolérů je tato ústředna poprvé realizována za pomoci jednodeskového linuxového počítače BeagleBone Black, který nabízí mnoho výhod. Kromě vysokého výkonu je to široké spektrum připojitelných periférií a v kontextu maloobjemové výroby také úspora nákladů. Z hlediska komplexnosti, kterou BeagleBone Black nabízí, je také celý proces návrhu a realizace jednodušší. Ač se stále jedná o zákazkovou výrobu na míru, koncept řešení tohoto prototypu by měl nabízet jednotnou platformu pro velké množství různě konfigurovaných aplikací bez nutnosti úpravy hardwaru. Konkrétní aplikace by se měnila pouze osazenými moduly a nahraným softwarem.

2.1 Řízení hodinové linky

V dnešní době neexistuje žádný oficiální standard nebo norma pro centrální řízení elektronických hodin. Z tohoto důvodu je prakticky nemožné vybudovat univerzální ústřednu, která by byla schopna ovládat veškeré elektronické hodiny napříč výrobci. Z průzkumu vyráběných hodin je však zřejmé, že způsob jejich řízení nese společné rysy. U analogových hodin je situace jednodušší, ve většině případů se jedná jen o jednoduchý krokový motor bez vlastní logiky a elektroniky. Řízení digitálních hodin je komplikovanější. Obecně platí, že se do hodin posílají bipolární pulzy o určité délce a určité amplitudě. Amplituda pulzů bývá nejčastěji 24 V, setkáme se ale také s hodinami, které vyžadují 12 V, 36 V nebo 60 V. Délka pulzů a jejich střídající se polarita zároveň slouží jako ochrana proti zákmitům na lince a proti rušení.

Výstupem hodinové ústředny jsou tři samostatné hodinové linky. Jedná se o dvou vodičové vedení, ke kterému se paralelně připojují jednotlivé hodiny. Předpokladem úspěšné instalace celého systému jednotného času je tak logicky rozvedení kabelů k místům, kde budou připojeny jednotlivé hodiny. Průměr vedených kabelů nemusí být příliš velký, maximální uvažovaný proud na lince jsou 3 A, ostatní výrobci často uvažují násobně méně.

2.1.1 Ovládání analogových hodin

Analogové hodiny jsou na ovládání jednoduché. Jedná se prakticky pouze o krokový motor, který se s příchodem dostatečně širokého pulzu posune o příslušný krok. Tento pulz by měl být alespoň 200 ms široký. Uvažujme nyní, že jsou hodiny řízeny 24 V bipolárními pulzy. V této situaci se z liché minuty na sudou posunou s příchodem pulzu +24 V a ze sudé minuty na lichou minutu se posunou s příchodem pulzu -24 V. To vyžaduje správné připojení hodin k lince, jinak by mohlo dojít ke zpoždění o jednu minutu ihned při spuštění do provozu. Některé analogové hodiny jsou vybaveny funkcí přetočit se po přijetí určitého tvaru pulzu (pulzů) na půlnoc (poledne). Bohužel pouze zanedbatelné množství analogových hodin je vybaveno touto funkcí. Velkou výhodou analogových hodin je možnost napájení přímo z dvou vodičového vedení a odpadá tak nutnost jejich vlastního napájení.



Obrázek 2.1. Analogové hodiny, příklad [1]

Jelikož analogové hodiny neposkytují zpětně žádnou informaci o svém okamžitém stavu, nastává problém s jejich nastavováním na konkrétní čas a s připojováním nových zařízení k lince. Tento problém lze řešit několika způsoby.

- První a nejméně praktický způsob je nastavit ručně na všech hodinách určitý čas (např. půlnoc) a až skutečný čas bude odpovídat této hodnotě, hodinovou linku spustit. První uvedení linky do provozu nepředstavuje velký problém, nicméně v situaci, kdy je potřeba připojit další zařízení, je nutné nastavit všechna zařízení znovu, nebo je nutno čekat 12 hodin, než bude skutečný čas znovu odpovídat času na lince.
- Abychom tento problém eliminovali, bude ústředna vybavena funkcí, která nastavování hodin zjednoduší. Podmínkou však je, že všechny hodiny na lince musí být nastaveny na stejný čas. Tento čas je libovolný. V této situaci se pomocí ovládací konzole ústředně zadá čas, na který jsou nastaveny hodiny na lince, a ta spočte počet

kroků (pulzů), které je nutné hodinám poslat, aby se synchronizovaly se skutečným časem.

- Některé vyráběné hodiny jsou schopny se při určitém tvaru pulzu automaticky nastavit na půlnoc (poledne). Ústředna pak jen odešle příslušný počet pulzů, čímž hodiny nastaví na správný čas. Tímto způsobem odpadá nutnost mít všechny hodiny na lince nastavené na stejný čas. Bohužel touto funkcí není většina zařízení vybavena a nelze s ní obecně počítat.

Ač je manipulace s analogovými hodinami méně praktická a vyžaduje obsluhu, má jejich používání i některé výhody. Jelikož hodiny nepotřebují vlastní napájecí napětí a k provozu stačí pulzy na lince, je velmi jednoduché řešit jejich záložní provoz. Při výpadku elektrické energie stačí mít připojený záložní zdroj k samotné ústředně a provoz hodin nebude přerušen. Jednoduchost řízení také vede k větší kompatibilitě mezi výrobky jednotlivých výrobců.

■ 2.1.2 Ovládání digitálních hodin

Složitější ovládání často využívají digitální hodiny. Tím jsou také dány širší možnosti jejich použití. Dle typu použitých hodin mohou kromě hodin, minut a případně vteřin zobrazovat také datum, časové pásmo, teplotu, vlhkost vzduchu a spoustu dalších informací. Nutno dodat, že veličiny jako teplota nebo vlhkost vzduchu si měří a následně zobrazují samy.

Digitální hodiny obsahují vlastní elektroniku, která v nejjednodušším případě řídí pouze zobrazení na displeji. Takové hodiny se pak stejně jako hodiny analogové řídí minutovými polarizovanými pulzy a prvotní nastavení času je nutné udělat ručně. Většina digitálních hodin je ovšem složitějších. Hodinám se každou minutu posílá celá informace o aktuálním čase a případně i datu. Princip kódování tohoto přenosu je odvislý od výrobce konkrétního systému. Kódování přenosu má však podobné principy a několik podmínek:

- musí být vždy nulová střední hodnota přenosu,
- impulsy musí být natolik krátké, aby na ně nereagovaly jiné prvky (např. krokové motory analogových hodin),
- musí být odolné proti rušení a
- harmonická složka přenosu by neměla zasahovat do pásma 77,5 kHz, kde se vysílá signál DCF77.

Protože je linka sdílena s analogovými hodinami, nemůže vysílání kódované zprávy pro digitální hodiny začít v libovolný okamžik. Musí být zajištěno, aby se vysílaná zpráva nekryla s minutovými nebo půlminutovými impulzy na lince. Každý bit se přenáší dvojicí pulzů opačné polarity dlouhých 20 ms. Log „1“ má nejprve kladný pulz a poté záporný, log „0“ má naopak první pulz záporný a druhý kladný. Často se přenáší celý byte – osmice bitů doplněná o jeden paritní bit. Dále vše závisí na zvoleném protokolu. Postupně jsou odesílány informace o stavu hodin, minut a dalších informací v BDC kódu. Použité hodiny musí být naprogramované tak, aby přijímanému signálu rozuměly a správně ho zobrazily.



Obrázek 2.2. Digitální hodiny, příklad [2]

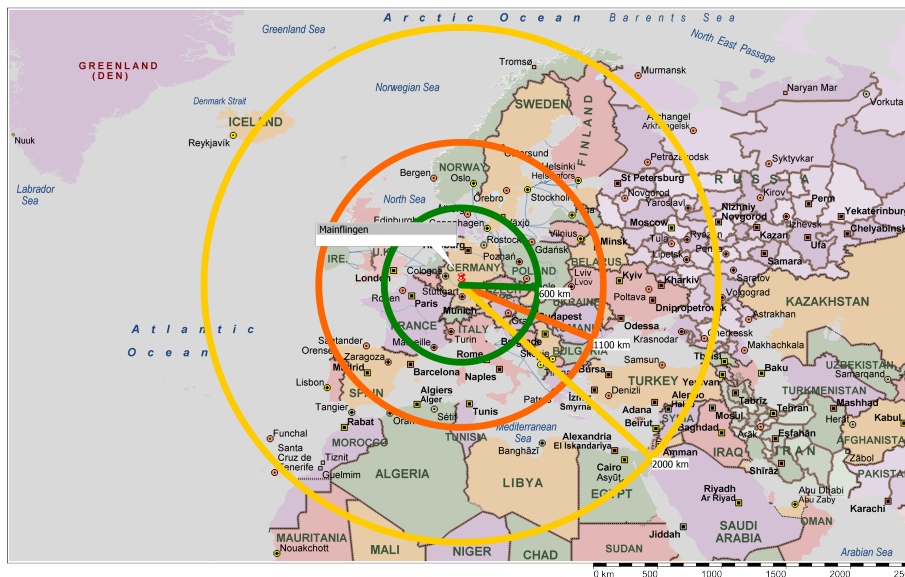
Digitální hodiny mají nespornou výhodu v tom, že je lze kdykoliv připojit k lince a ony se samy synchronizují s časovým normálem v ústředně. Velkou nevýhodou je však to, že způsob posílání časové informace není jednotný a každý výrobce má svůj vlastní. Výrobci navíc kódování tohoto přenosu považují za vlastní know-how, který tudíž není veřejně přístupný. To vede k uzavřenosti celého systému a bez znalosti konkrétního komunikačního protokolu není možné digitální hodiny cizích výrobců ovládat.

2.2 Zdroje časového normálu

Pro přesné řízení elektronických hodin je nejprve potřeba získat přesný čas. Ten je označen jako časový normál a lze ho získat z několika zdrojů. V našich geografických podmínkách lze uvažovat tři zdroje časového normálu. Dva z nich jsou globální a jeden použitelný pouze v části Evropy.

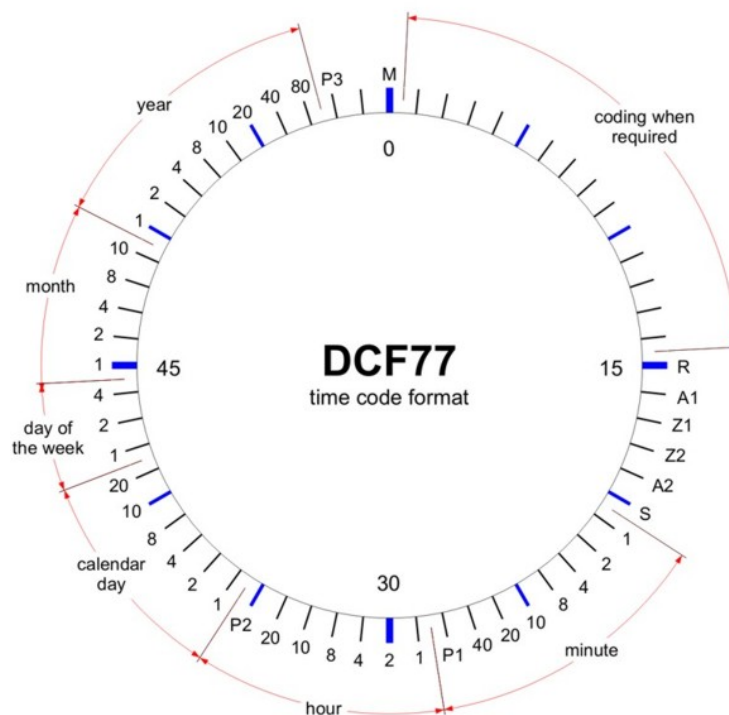
2.2.1 DCF77

DCF77 je dlouhovlnné radiové vysílání na kmitočtu 77,5 kHz, které v sobě nese informaci o přesném čase. Vysílací stanice je umístěna v Mainflingenu nedaleko Frankfurtu nad Mohanem v Německu. Vysílací výkon je 50 kW. Vysílání probíhá pravidelně a nepřetržitě od roku 1970. Je garantováno, že minimálně 99,7 procent roku bude vysílání k dispozici a v naprosté většině případů je celková doba přerušení za jeden rok v řádu jednotek hodin. Výpadky bývají způsobené zejména špatným počasím. Dosah vysílače je teoreticky až 2000 km (viz obrázek 2.3), to však platí za ideálních podmínek šíření. Pro Českou Republiku je dosah vysílače dostatečný. [3],[4]



Obrázek 2.3. Mapa pokrytí signálem DCF77 [5]

Modulování přenášené informace je prováděno pulzně šířkovou modulací poklesem amplitudy nosné na 25 procent při začátku každé sekundy. Tento pokles odpadá při 59. sekundě každé minuty, čemuž se říká minutová značka. Pokles trvá buď 100 ms, což představuje log 0, nebo 200 ms, což představuje log 1. [3] [4]



Obrázek 2.4. Kódovací schéma vysílání DCF77 [6]

Jak je vidět na obrázku 2.4, není vysílána pouze informace o čase. Vysílání je rozděleno do několika sekcí. [4]

- První část minuty je určena k vysílání informací o počasí a civilním nebezpečí.

- Druhá část určuje počet minut.
- Třetí část určuje počet hodin.
- Čtvrtá část určuje den v měsíci.
- Pátá část určuje den v týdnu.
- Šestá část určuje měsíc v roce.
- Sedmá část určuje rok.

Ve vysílání jsou také uloženy speciální znaky. [4]

- M – minutová značka (vždy log 0)
- R – použitá anténa (normální anténa log 0, náhradní anténa log 1)
- A1 – oznámení změny z SEČ na SELČ nebo naopak (log 1 je změna, ukazuje se 1 hodinu před změnou)
- Z1 a Z2 – časové znaky (označují aktuální časové pásmo)
- A2 – oznámení přestupné sekundy (log 1 je změna)
- S – startovací bit kódované časové informace (vždy log 1)
- P1 P3 – kontrolní bity (sudá parita)

Pomocí vhodného přijímače signálu je možné každou minutu získat přesný čas, který následně používá tato ústředna.

■ 2.2.2 GPS

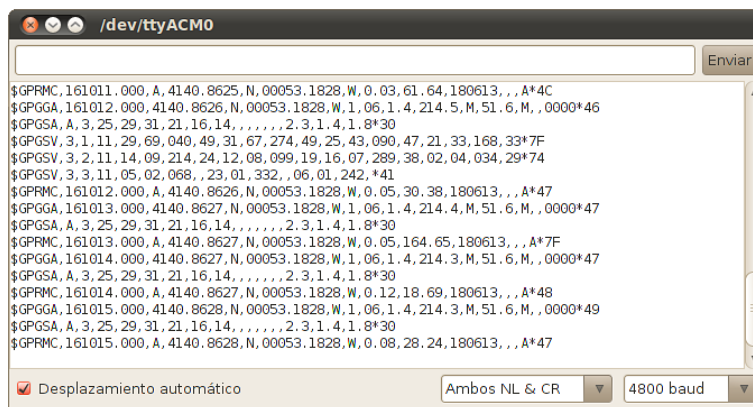
GPS je zkratka pro Global Positioning System – globální systém pro určování polohy. Systém funguje na principu komunikace přijímače s družicemi na oběžné dráze Země. Původně se jednalo pouze o vojenský projekt americké armády, v 80. letech 20. století se ale tento družicový systém otevřel veřejnosti. Díky tomu jsme kdekoliv na Zemi (nebo nad ní) schopni určit polohu přijímače s přesností v jednotkách metrů spolu s nadmořskou výškou a navíc s přesným časem. Jelikož je každá družice vybavena několika atomovými hodinami, systém GPS poskytuje čas s přesností na jednotky nanosekund. Přesný čas je pro celý systém naprosto klíčový a zajišťuje jeho správnou synchronizaci. Navíc vzhledem k množství družic systém GPS funguje prakticky bez přerušení a na celém světě. [7]



Obrázek 2.5. Ilustrace systému GPS [8]

V GPS je vysíláno velké množství informací. Pro zjištění přesného času je důležitý protokol NMEA, který obsahuje několik vět. Tyto věty obsahují různé informace. Časové

informace najdeme ve zprávách GGA, GLL, RMC a ZDA. Například zpráva ZDA slouží pro vysílání UTC času (světový čas). Obsahuje UTC čas, den, měsíc, rok, místní časové pásmo, místní minutové pásmo a kontrolní součet. Tyto zprávy jsou vysílány každou vteřinu a je možné z nich získat přesný čas pro další synchronizaci. Způsob zápisu jednotlivých zpráv NMEA je znázorněn na obrázku 2.6. [9]



```

$GPRMC,161011.000,A,4140.8625,N,00053.1828,W,0.03,61.64,180613,,A*4C
$GPGGA,161012.000,4140.8626,N,00053.1828,W,1.06,1.4,214.5,M,51.6,M,,0000*46
$GPGSA,A,3,25,29,31,21,16,14,,,,,,,,,2,3,1,4,1,8*30
$GPGSV,3,1,11,29,69,040,49,31,67,274,49,25,43,090,47,21,33,168,33*7F
$GPGSV,3,2,11,14,09,214,24,12,08,099,19,16,07,289,38,02,04,034,29*74
$GPGSV,3,3,11,05,02,068,,23,01,332,,06,01,242,*41
$GPRMC,161012.000,A,4140.8626,N,00053.1828,W,0.05,30.38,180613,,A*47
$GPRMC,161013.000,4140.8627,N,00053.1828,W,1.06,1.4,214.4,M,51.6,M,,0000*47
$GPGSA,A,3,25,29,31,21,16,14,,,,,,,,,2,3,1,4,1,8*30
$GPRMC,161013.000,A,4140.8627,N,00053.1828,W,0.05,164.65,180613,,A*7F
$GPGGA,161014.000,4140.8627,N,00053.1828,W,1.06,1.4,214.3,M,51.6,M,,0000*47
$GPGSA,A,3,25,29,31,21,16,14,,,,,,,,,2,3,1,4,1,8*30
$GPRMC,161014.000,A,4140.8627,N,00053.1828,W,0.12,18.69,180613,,A*48
$GPGGA,161015.000,4140.8628,N,00053.1828,W,1.06,1.4,214.3,M,51.6,M,,0000*49
$GPGSA,A,3,25,29,31,21,16,14,,,,,,,,,2,3,1,4,1,8*30
$GPRMC,161015.000,A,4140.8628,N,00053.1828,W,0.08,28.24,180613,,A*47

```

Obrázek 2.6. Znázornění zápisu zpráv NMEA [10]

2.2.3 NTP

NTP neboli Network Time Protocol byl navržen pro synchronizaci vnitřních hodin počítačů po paketových sítích. Účel je zřejmý, nabídnout všem počítačům stejný a přesný čas. NTP umožňuje udržovat čas s přesností menší než deset milisekund. Komunikace probíhá po internetu a je tedy závislá na kvalitě a stálosti internetového připojení. Například nesymetrické internetové připojení vede k přesnosti pouze na stovky milisekund. Samotná komunikace probíhá tak, že klient (zařízení, počítač) pošle dotaz na NTP server, který mu v odpovědi sdělí údaje o svém přesném čase. NTP servery jsou samy v první fázi synchronizovány externím zdrojem času (atomové hodiny, GPS, nebo rádiem vysílaný čas). Dnes existuje i zjednodušená verze SNTP, ta však nepočítá se zpožděním a rozptylem paketů a není schopná pamatovat si předchozí komunikaci. Z těchto důvodů nemůže svůj čas dále sdílet a sama se stát NTP serverem. V dnešní době je na internetu dostupné velké množství NTP serverů, ze kterých je možné čerpat přesný čas. Za výhodu i nevýhodu systému lze považovat nutnost připojení zařízení k internetu nebo alespoň k počítačové síti. [11],[12]

2.3 Komunikace s uživatelem

Z důvodů popsaných v kapitole 2.1 o způsobech řízení elektronických hodin je nutná základní obsluha systému. Tu nejčastěji zajišťuje správce daného objektu. Obsluha není náročná, systém je z velké části autonomní. Pokud bychom uvažovali pouze použití digitálních hodin, je systém zcela autonomní. Interakce s uživatelem je nutná při spuštění linky, připojování nových hodin k lince, nebo při výpadku napájení a jiných havarijních situacích. Jediné, co se od obsluhy žádá, je přepnutí systému do servisního módu, zadání nastaveného času na lince (hodiny musejí být všechny nastaveny na stejný čas) a pro zpětné uvedení do provozu odejít ze servisního módu. Vhodné způsoby ovládání jsou dva a je na zákazníkovi, který si vybere a který bude lépe vyhovovat umístění aplikace. Navržená hodinová ústředna je připravena pro použití obou způsobů.

■ 2.3.1 Dotykový displej

První možností je ovládat systém pomocí dotykového displeje. Ten umožňuje zobrazení velkého množství informací a je jednoduchý na ovládání. Uživatel pomocí dotykového panelu přepne systém do servisního módu a zadá čas na konkrétní hodinové lince.

Nevýhodou tohoto způsobu je nutnost osobní interakce uživatele. V případě umístění hodinové ústředny do hůře přístupných prostor může být tento způsob obsluhy nepohodlný. Naopak v dobře dostupných prostorách se jedná o jednoduchý a uživatelsky přívětivý způsob obsluhy.

■ 2.3.2 Webové rozhraní

Druhou možností je vzdálené ovládání přes webové rozhraní. Uživatel se na počítači umístěném ve stejné síti jako hodinová ústředna vzdáleně připojí k webové aplikaci spuštěné z ústředny. Tato aplikace nabízí stejné možnosti ovládání jako dotykový displej.

Výhodou tohoto způsobu je, že uživatel vše potřebné zařídí od stolu ve své kanceláři. Ne vždy je však ústředna připojena k počítačové síti, tudíž tento způsob není vždy možný. Je na zákazníkovi, aby se rozhodl, která možnost mu více vyhovuje nebo která je vůbec proveditelná. Ovládání přes webové rozhraní je však přívětivá a levnější varianta.

■ 2.4 Datové sběrnice

Do této hodinové ústředny jsou zapojeny také dvě rozšiřující datové sběrnice. Jedná se o standardní průmyslově hojně využívané sběrnice RS232 a RS485. V této fázi není umožněn jejich současný provoz a uživatel si musí vybrat buď jednu nebo druhou. Přítomnost těchto sběrnic výrazně rozšiřuje komunikační potenciál zařízení. Tímto způsobem je možné šířit přesný čas k dalším aktivním zařízením, například ke spínacím obvodům pro ovládání zvonků či osvětlení nebo k různým docházkovým systémům. Pomocí sběrnice RS485 je například možné řídit některé druhy digitálních elektronických hodin. Důvodem pro přítomnost sběrnic v zařízení je případná možnost komunikace s ostatními prvky v celém systému.

■ 2.5 Spínaný výstup

Ústředna je také vybavena jedním spínaným výstupem. V případě zájmu zákazníka by ve finálním výrobku těchto spínaných výstupů mohlo být více. Výstup je udělaný obecně a je na uživateli, co k němu připojí. Může se jednat například o školní zvonky, osvětlení místností nebo venkovních prostorů či spuštění jiné signalizace. Na přání zákazníka by se relé spínalo v určitých časech. Tyto časy je možné nastavovat v komunikačním rozhraní. Přítomnost spínaných výstupů zásadně rozšiřuje využitelnost zařízení.

■ 2.6 Nouzový provoz

Při provozu hodinové ústředny může dojít k jeho narušení. Toto narušení mohou způsobit dva typy výpadků. Prvním typem je výpadek napájení, druhý představuje výpadek zdrojů časového normálu.

■ 2.6.1 Výpadek napájení

Horší důsledky má výpadek napájení, který ústřednu zcela odstavuje z provozu. Přestává fungovat jak příjem časového normálu, tak řízení hodinových linek. Ochranu představuje použití záložního napájení po dobu trvání výpadku. Je však důležité uvědomit si, jaký důsledek záložní provoz má. Hodinová ústředna je dále synchronizována za pomoci jednoho ze zdrojů časového normálu. Přijímače GPS a DCF jsou napájeny přímo z ústředny, fungují tudíž bez problémů. Synchronizaci s NTP serverem není možné zaručit, neboť je závislá na připojení k internetu. Vlivem výpadku napájení může být internetové připojení nedostupné. Problém také nastává na hodinové lince. Analogové hodiny jsou napájeny přímo z ní, a tak není problém s jejich provozem. Digitální hodiny ale potřebují vlastní napájení, tudíž s výpadkem napájení přestanou fungovat. Pro tento případ jsou některé typy digitálních hodin vybaveny záložní baterií pro nouzový provoz procesoru, který i při výpadku napájení sleduje hodinovou linku. V tomto případě však v rámci úspory energie nezobrazuje čas na displayi.

V této souvislosti záleží na konkrétní aplikaci hodinové ústředny. Například v prostorách, kde neexistuje žádný záložní zdroj napájení a k synchronizaci času je použit NTP server, nedává zálohování napájení velký smysl. V místě, kde se používají jen analogové hodiny a časový normál se bere z GPS nebo DCF, naopak stačí zálohovat napájení pouze hodinové ústředny a celý systém je dále funkční. Z těchto důvodů nebyl k ústředně instalován žádný akumulátor, který by zajišťoval nouzový chod. Pokud bude mít zákazník zájem, může být na úrovni připojení do sítě s napětím 230 V umístěn záložní zdroj UPS, který zajistí chod ústředny.

■ 2.6.2 Výpadek zdroje časového normálu

Pokud budou k ústředně připojeny všechny tři zdroje časového normálu, je velmi malá pravděpodobnost, že by došlo k výpadkům všech tří zdrojů najednou. Zákazník si ale může zvolit levnější řešení, kdy by čas získával pouze jedním z těchto způsobů. Pro tento případ je možnost ústřednu osadit přesným, teplotně kompenzovaným krystalem, který by řídil po dobu výpadku distribuci času s dostatečnou přesností.

■ 2.7 Možnosti využití

Primárním účelem této hodinové ústředny je centralizované řízení elektronických hodin, které zajistí automatický a bezporuchový provoz. Podmínkou provozu je infrastruktura v podobě dvou vodičové linky přivedené od ústředny k jednotlivým hodinám. Vzhledem ke konstrukci se třemi nezávislými hodinovými linkami a maximálním proudem na lince 3 A je možné připojit velké množství zařízení (dle typu a použitých hodinových strojků u analogových hodin). Neměl by být problém pokrýt velké prostory s velkým množstvím hodin. Dále lze využít programovatelný spínaný výstup, typickým využitím je školní zvonění. Pokud by zařízení bylo připojeno k internetu, lze ho také využívat jako NTP server, který by dále šířil přesný čas z GPS.

Využití se nabízí ve velkých objektech jako jsou školy, nemocnice, výrobní haly, kancelářské budovy nebo jiná administrativní centra. Uplatnění lze najít také na různých sportovištích (např. golfové hřiště), velkou kapitolou jsou dopravní terminály jako nádraží nebo letiště.

2.8 Porovnání s konkurencí

Podobné produkty jako je tato hodinová ústředna nabízí několik firem. K porovnání jsem zvolil výrobky tří společností. Mým cílem bylo udělat konkurenceschopný výrobek, který by bylo možné upravit co nejvíce na míru podle přání zákazníka.

Velkým hráčem na trhu je společnost Elektročas s.r.o., která nabízí kromě hodinových ústředn také velké množství elektronických hodin. Jejich Hlavní hodiny EH71 umožňují ovládní obou typů hodin. Mají nastavitelnou délku minutových impulsů a jednu hodinovou linku (model EH72 má linky dvě). Maximální proud na lince je 0,45 A. Synchronizace času probíhá za pomoci signálu DCF77. K této hodinové ústředně je možné dokoupit hodiny signální, které se připojí na linku a ovládají různé spínané obvody. [13]

Další společností vyrábějící hodinové ústředny je společnost BRNOTRON, v.o.s.. Tato společnost nabízí velké množství různých konfigurací. Jejich hlavní hodiny mívají jednu nebo dvě hodinové linky s maximálním proudem až 1,3 A. Jednotlivá zařízení se liší zejména v počtu a použití spínaných obvodů. Pro synchronizaci je opět použito DCF77. [14]

Komplexní řešení nabízí společnost Visitech. Ta k synchronizaci hodin využívá kromě drátového připojení i bezdrátové řízení hodin DHF. Navíc nabízí jako zdroj přesného času všechny tři v této práci uvažované způsoby. Zajímavé je řešení, kdy výrobce určitý druh digitálních hodin připojuje do ethernetové sítě, čímž kromě jejich synchronizace umožňuje i jejich napájení pomocí POE (Power over Ethernet). [15]

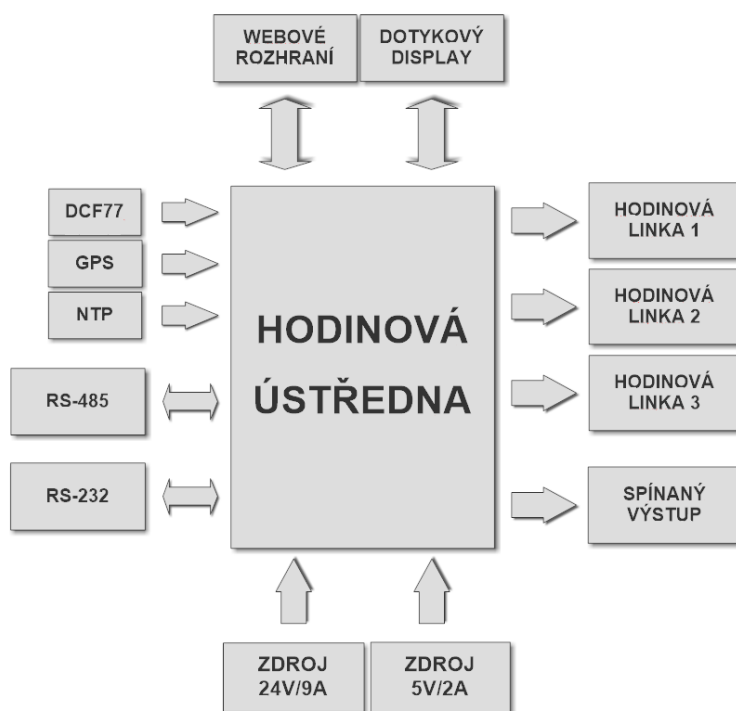
Hodinová ústředna navržená v této práci má ve srovnání s konkurencí lépe řešené výstupy hodinových linek, kdy uživatel může využívat tři nezávislé linky s maximálním proudem na každé z nich až 3 A. Zároveň je přítomna možnost připojení signalizačních zařízení na spínaný výstup. Výhodou je také přítomnost sběrnic RS232 a RS485. V některých aplikacích nemusí být vždy dostupný signál DCF77, proto je velkou výhodou možnost výběru jiných zdrojů časového normálu. Chybí naopak bezdrátové řízení elektronických hodin nebo možnost řídit hodiny pomocí ethernetové sítě.

Kapitola 3

Hardware hodinové ústředny

Návrh hardwaru hodinové ústředny byl vytvořen za pomoci vývojového prostředí pro tvorbu schémat a návrh designu desek plošných spojů EAGLE od společnosti AUTODESK. Jeho použití mi umožnila studentská licence, kterou společnost AUTODESK nabízí.

Návrh začal vytvořením blokového diagramu, který je zobrazen na obrázku 3.1. Tím se ujasnilo, jaké funkční bloky bude ústředna potřebovat a přešlo se k tvorbě schématu zapojení. Kompletní schéma zapojení je k zobrazení v příloze A.



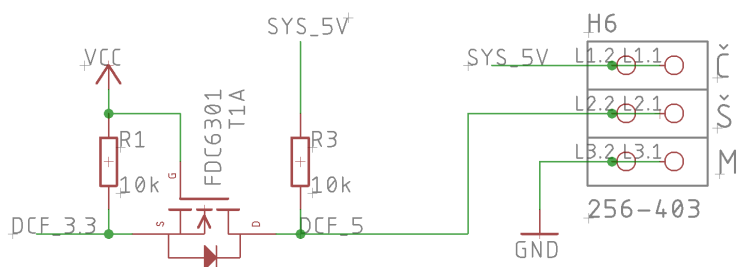
Obrázek 3.1. Blokový diagram hodinové ústředny

Poté bylo možné přejít k vlastnímu designu desky plošných spojů. Bylo nutné dát pozor na vzájemnou kompatibilitu mezi mnou navrženou deskou a počítačem BeagleBone Black. Výsledný návrh desky plošných spojů je možné vidět v příloze B.

Nyní si rozebereme jednotlivé bloky návrhu.

3.1 Převod mezi úrovněmi napětí 3,3 V a 5 V

Pro převod logických úrovní mezi 3,3 V a 5 V je využit integrovaný obvod *FDC6301*, který nabízí jednoduchou obousměrnou konverzi využitím unipolárního tranzistoru MOSFET a dvou pullup rezistorů, jak je patrné na obrázku 3.2.



Obrázek 3.2. Převod mezi úrovněmi napětí 3,3 V a 5 V – ukázka na signálu DCF

Pokud ani jedna strana nekomunikuje, pull-up rezistory zařídí, že se na obou stranách tranzistoru nachází logická jednička. Rozdíl napětí mezi U_S a U_G je tedy roven nule a tranzistor je zavřený.

Pokud strana s nižším napětím spojí linku se zemí a vytvoří se na ní logická nula, rozdíl napětí mezi U_S a U_G se zvýší a tranzistor se otevře. Otevřením tranzistoru se logická nula objeví i na straně s vyšším napětím.

Pokud strana s vyšším napětím spojí linku se zemí, dioda mezi *source* a *drain* kanály tranzistoru způsobí, že se zvětší rozdíl napětí mezi kanály a tranzistor se otevře. Otevřením tranzistoru se logická nula přenesse i na stranu nižšího napětí.

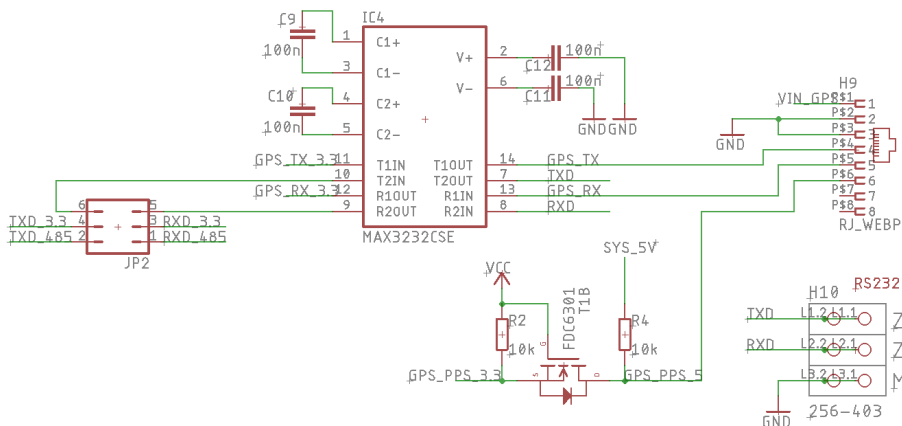
Vzhledem k tomu, že BeagleBone Black pracuje s 3,3 V logikou a většina připojených periférií pracuje s 5 V logikou, je tento způsob převodu použit hned několikrát.

3.2 Příjem signálu DCF77

Příjem signálu z modulu DCF77 stačí pouze převést z 5 V na úroveň 3,3 V a připojit k pinu procesoru. Na konektoru pro modul DCF77 je připraveno napájení modulu 5 V a společná zem GND, viz obrázek 3.2.

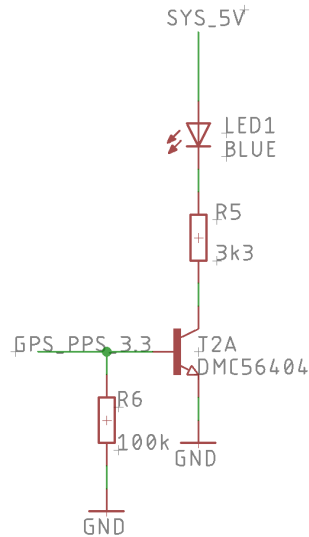
3.3 Příjem signálu GPS

Pro příjem signálu GPS je využit přijímač GPS 16x LVS od společnosti Garmin. Výhodou je zabudovaná anténa ve vodotěsném krytu. Přijímač vykazuje vynikající citlivost. Je osazen 8pinovým konektorem typu RJ, který je možné připojit přímo do konektoru v navrženém obvodu, který současně obstarává napájení modulu. Pro převod úrovní linky RS232 a signálů Rx/D/TxD 3,3 V je využit integrovaný obvod MAX3232.



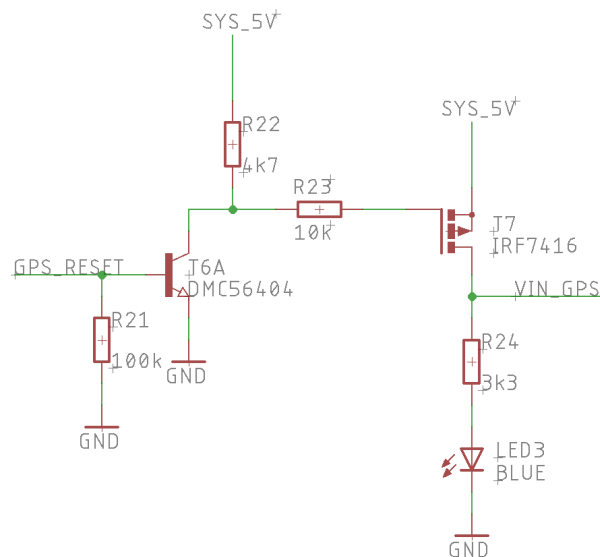
Obrázek 3.3. Využití integrovaného obvodu MAX3232 pro signál GPS a sběrnici RS232

Pro přesné řízení času se u systému GPS používá signál PPS. Ten je přijímán každou sekundu s velkou přesností. V procesoru systému může být využit k přerušení a tím časování procesů uvnitř operačního systému. Příjem signálu PPS je indikován na signalizační LED1, viz obrázek 3.4.



Obrázek 3.4. LED signalizující signál GPS-PPS

Napájení GPS přijímače je povoleno řídicím systémem. Toto řešení umožňuje v případě problémů s přijímačem provést jeho reset a počkat na jeho odezvu. Zapínání a vypínání je realizováno pomocí MOSFET tranzistoru s kanálem typu P. Otevřením tranzistoru T6A dojde k poklesu napětí mezi U_G a U_S a otevření tranzistoru T7. Přítomnost napájecího napětí pro GPS přijímač je indikována pomocí LED3. Zapojení je zobrazeno na obrázku 3.5.

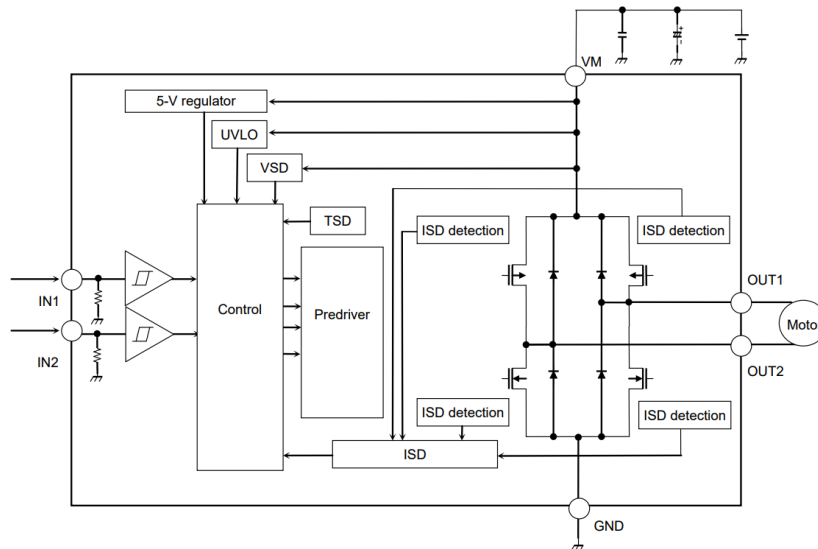


Obrázek 3.5. Napájení GPS a jeho reset

3.4 Řízení hodinových linek

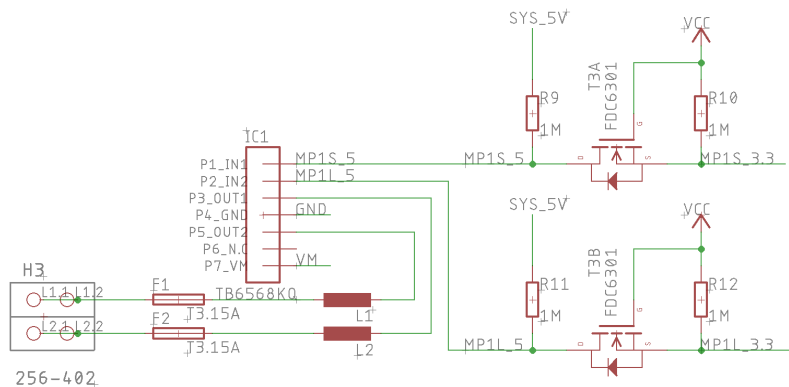
Pro řízení hodinových linek je využit integrovaný obvod TB6568KQ od společnosti Toshiba. Tento obvod obsahuje můstkové zapojení čtyř tranzistorů a umožňuje velmi

jednoduché řízení pomocí dvou logických vstupů. Obvod je tedy řízen nízkým napětím z procesoru a na výstupech spíná napájecí napětí. Obvod zároveň chrání oba výstupy před současným sepnutím, obsahuje ochranu proti zkratu a přetížení. Napájecí napětí je v rozmezí od 10 V do 45 V, výstupní proud může být až 3 A. Vnitřní zapojení tohoto obvodu je zobrazeno na obrázku 3.6. [16]



Obrázek 3.6. Vnitřní zapojení integrovaného obvodu TB6568KQ [16]

K výstupům linky jsou připojeny pojistky jako ochrana před vysokým proudem a tlumivky, které linku chrání před rušením.



Obrázek 3.7. Zapojení obvodu pro řízení hodinových linek

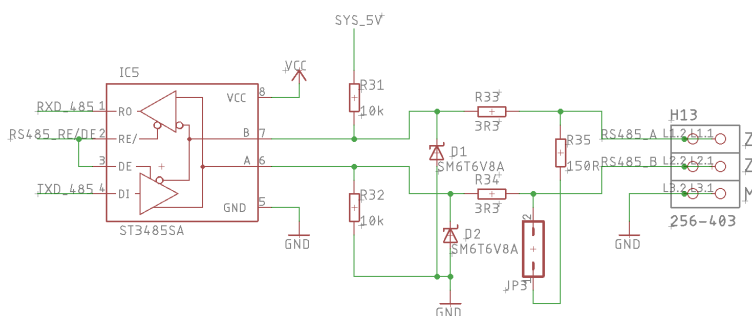
U jedné z hodinových linek muselo dojít ke změně hodnot pull-up rezistorů u převodu mezi úrovněmi 5 V a 3,3 V (viz obrázek 3.7). V původní konfiguraci s odpory 10 k Ω docházelo k problémům se spuštěním BeagleBone Black. Po analýze problému bylo zjištěno, že piny procesoru 45 a 46 na konektoru P8 jsou používány při bootování systému a při spouštění by neměly být připojeny k žádné logické úrovni. Problém se podařilo vyřešit osazením pull-up rezistorů hodnoty 1 M Ω . Větší úbytek napětí na rezistoru umožní bootování systému a převodník úrovní stále funguje. Při dalším návrhu se tyto signály přesunou mimo piny využívané při bootování.

3.5 RS232

Na obrázku 3.3 je patrné vyvedení sběrnice RS232. Pomocí jumperu si uživatel zvolí mezi použitím sběrnice RS232 nebo RS485. U sběrnice RS232 dochází pouze k převodu úrovní signálů za pomoci integrovaného obvodu MAX3232. Převedený signál je pak přiveden rovnou na konektor.

3.6 RS485

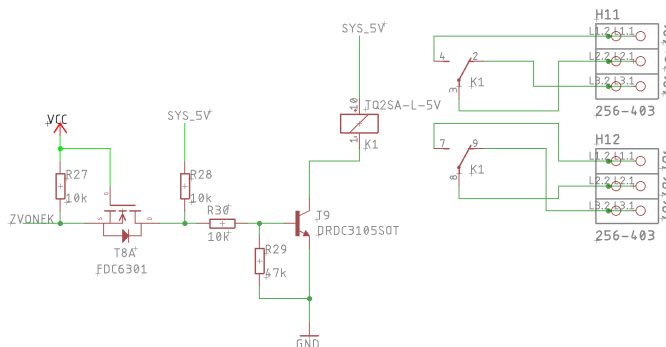
Sběrnice RS485 je vytvořena ze standardu RS232 pomocí převodníku úrovně - obvodu ST3485. Poloduplexní dvou vodičové sériové vedení signálů A a B je převedeno na RxD a TxD 3,3 V. Směr komunikace je řízen signálem RS485-RE/DE přímo z řídicí jednotky. Linku je možné zatížit rezistorem o hodnotě 150 Ω pomocí jumperu. Jako ochrana proti zkratu jsou použity transily.



Obrázek 3.8. Zapojení obvodu pro sběrnici RS485

3.7 Spínání relé

K ovládání relé je použit hradlový budič DRDC3105SOT, který je výrobcem navržen pro ovládání cívek s napájením 3–5 V. Výrobce garantuje velmi dobrou izolaci mezi vstupní a výstupní částí a malý ovládací proud. Přesto, jak je vidět na obrázku 3.9, je pro maximální odlehčení výstupního pinu řídicího obvodu zvolen převod mezi 3,3 V a 5 V.

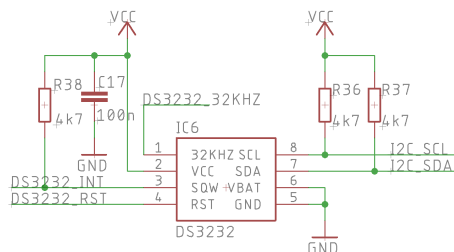


Obrázek 3.9. Zapojení obvodu pro výstupní spínaný obvod

3.8 DS3232

Jako záloha pro případ výpadku zdrojů časového normálu jsou pro správný chod jednotky použity hodiny reálného času typu DS3232. Tento obvod zaručuje maximální

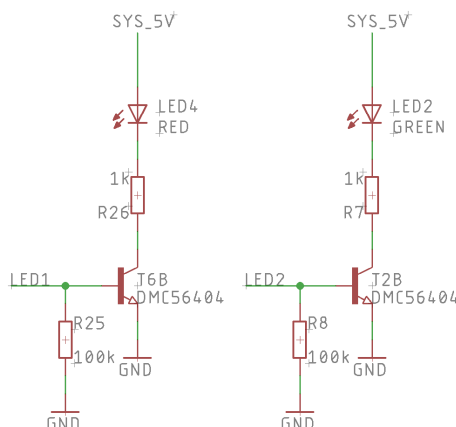
odchylku času 0,432 s/den (při provozní teplotě -40°C až 85°C) a umožní relativně přesný provoz jednotky bez synchronizačního signálu po dobu několika hodin. Obvod je připojen k řídicímu systému sběrnici I2C. Z výstupu obvodu DS3232 je dále využíván signál o kmitočtu 32 kHz určený k dalšímu využití v řídicím systému.



Obrázek 3.10. Zapojení integrovaného obvodu DS3232

3.9 Signalizační LED

Pro zapojení pomocných signalizačních diod je využit integrovaný obvod DMC56404. Jeho malé rozměry spolu s dvojicí tranzistorů umožní bezpečně a s minimální zátěží výstupních pinů počítače BeagleBone Black ovládat LED diody.



Obrázek 3.11. Zapojení obvodu signalizačních LED diod

3.10 Napájení

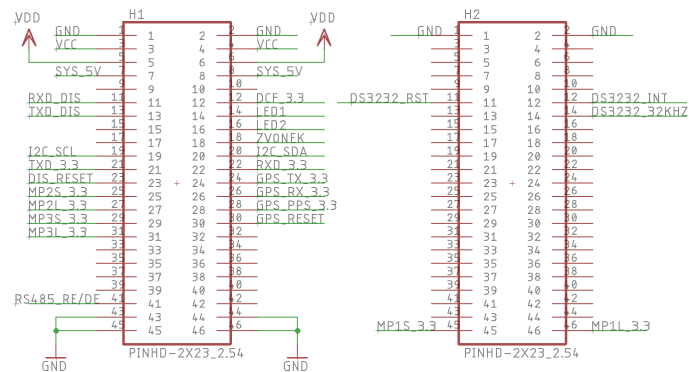
Hodinová ústředna je napájena ze dvou zdrojů napětí. Prvním je zdroj 5 V pro běh celého systému, který by měl být schopen poskytnout proud alespoň 1 A. Druhý zdroj napájí integrovaný obvod TB6568KQ, který dále napájí hodinové linky. Napětí tohoto zdroje je 24 V a při plném běhu hodinových linek může docházet k odběru až 9 A. Konkrétní zdroj bude zvolen podle počtu připojených hodin a jejich proudových nároků.

3.11 Displej, konektory

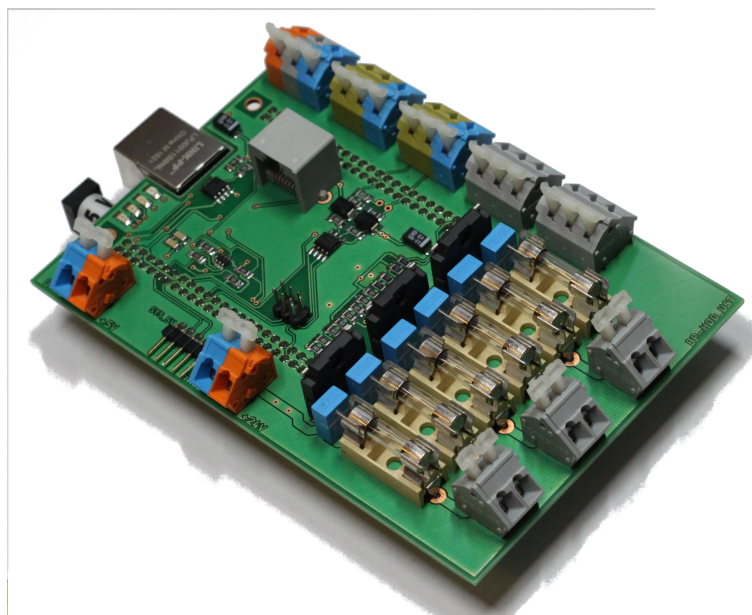
Dotykový displej je připojen jednoduchým konektorem přímo k pinům procesoru. Zároveň je k němu připojeno napájení 5 V a GND. Všechny ostatní použité konektory jsou značky WAGO, model 256-40*. Jedná se o jednoduchý typ konektorů vhodný pro tento koncept.

3.12 Propojení desky plošných spojů a BeagleBone Black

Propojení desky BeagleBone Black s navrženou deskou hodinové ústředny je uskutečněno pomocí dvou 46 pinových oboustranných kolíků, které jsou nasunuty do dutinových lišt na desce počítače. Na obrázku 3.12 si můžeme prohlédnout, na který pin procesoru míří signály hodinové ústředny, na obrázku 3.13 pak vzhled vyrobené a osazené desky hodinové ústředny.



Obrázek 3.12. Propojení signálů mezi Cape a BBB

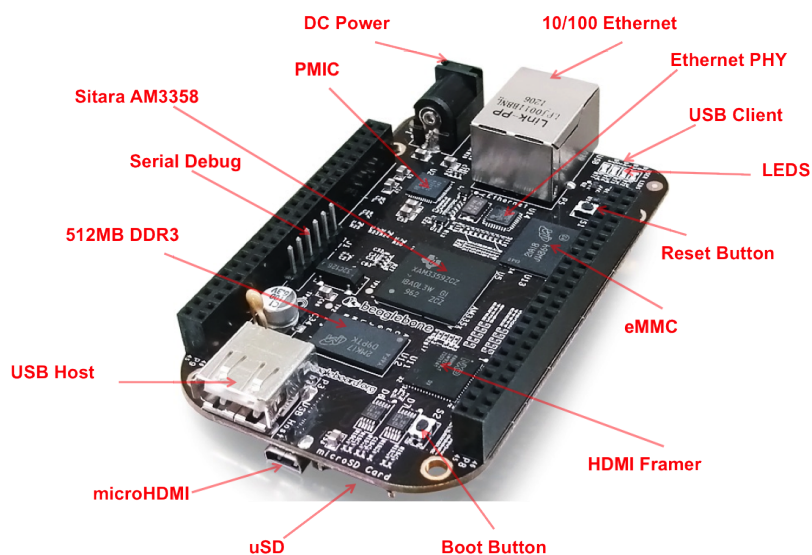


Obrázek 3.13. Vyrobená deska plošných spojů hodinové ústředny

Kapitola 4

BeagleBone Black

Na začátku práce bylo učiněno důležité rozhodnutí, a to na jakém hardwaru bude hodinová ústředna postavena. Dříve se podobné aplikace tvořily za pomoci mikrokontrolérů, dnes lze stavět na jednodeskových počítačích, které vynikají svým výkonem, uživatelskou přívětivostí a širokou podporou jednoduše dostupnou na internetu. Konečné rozhodnutí padlo na počítač BeagleBone Black, nicméně podobných alternativ existuje velké množství.



Obrázek 4.1. BeagleBone Black [17]

BeagleBone Black patří do rodiny open-source jednodeskových počítačů BeagleBoard, za kterými stojí společnost Texas Instruments. Desky z rodiny BeagleBoard jsou zaměřeny jak na výuku programování na školách a pro studenty, tak na vývojáře, kteří na těchto počítačích stavějí své výrobky. Tomu pomáhají zejména dva 46 pinové konektory, ke kterým je možné připojit tzv. Capes – originální nebo vlastní rozšiřující moduly. Díky těmto rozšiřujícím konektorům a velkému množství periférií a sběrnic procesoru jsou možnosti návrhu vlastních zařízení opravdu široké. [18]

4.1 Parametry

V tabulce 4.1 je vypsáno několik základních parametrů BeagleBone Black. Srdcem zařízení je výkonný procesor ARM Cortex-A8, který je součástí integrovaného obvodu SoC (System on Chip) AM3358BZCZ100 od společnosti Texas Instruments. V tomto obvodu se dále nachází multimediální subsystém obsahující např. grafický akcelerátor, paměťový subsystém a programovatelný real-time subsystém (PRU). AM3358 také obsahuje velké množství periferních rozhraní. Většina výstupů procesoru má až osm signálových funkcí, ze kterých lze vybírat. Ne všechny periférie však můžou být spuštěné najednou a

uživatel musí dát pozor, aby se vzájemně nekryly. K tomu pomáhá utilita firmy Texas Instruments Pin Mux, kde je možné ověřit vlastní konfiguraci. Existuje široká škála možností. K dispozici jsou 4 sady obecných I/O pinů (GPIO), až 6x UART, 3x I2C, 2x SPI. Dvakrát je přítomna i sběrnice CAN a lze vyvést Ethernetový port s rychlostí až 1GBs. Ne všechny periferie však lze použít zároveň. Tabulka s použitelnými módy pinů na konektorech P8 a P9 je uvedena v příloze C. [18],[19]

Processor	Sitara AM3358BZCZ100 1GHz, 2000 MIPS
SDRAM Paměť	512MB DDR3L 800MHz
Flash paměť	4GB, 8bit Embedded MMC
Napájení	miniUSB, DC Jack nebo 5VDC z Headeru
HS USB 2.0 Host Port	přes USB1,500mA LS/FS/HS
HS USB 2.0 Client Port	přes USB0, klientský mód přes miniUSB
Ethernet	RJ45, 10/100
SD/MMC konektor	microSD , 3.3V
Výstupní napětí	5V, 3.3V , VDD-ADC(1.8V)
Periferie	např. 65x GPIO, 4x UART, 2x I2C, 2x SPI, 2x PRU

Tabulka 4.1. Vybrané parametry BeagleBone Black [19].

Důležitá je možnost využití rozšiřujících hardwarových modulů Capes. Ty se připojují pomocí dvou 46 pinových konektorů a dochází k překrytí původního BeagleBone Black. Uživatel může zvolit již vytvořené Capes, které jsou vyrobeny komunitou BeagleBoard, nebo si sestavit vlastní. Dnes jsou k dispozici desítky již vytvořených Capes, jedná se o různé moduly pro displeje, Bluetooth moduly, moduly řízení a další. Vlastní Capes musí brát ohled na možnosti procesoru, přesto má návrhář široké pole působnosti a na základě BeagleBone Black může vytvořit spoustu aplikací. Při správné konfiguraci lze současně připojit až 4 rozšiřující desky najednou. [19]

4.2 Srovnání s podobnými produkty

Podobných zařízení, jakým je BeagleBone Black, je na trhu více. Za průkopníka takových zařízení se považují vývojové kity Arduino, které svým pojetím dovolovaly uživateli tvořit nejrůznější aplikace. Také je lze využít jako platformu pro výuku programování, u studentů se těší velké oblibě. Podobným směrem jako Arduino se vydaly společnosti jako například STMicroelectronics nebo Microchip Technology, které začaly vyrábět vlastní vývojové kity často kompatibilní s deskami Arduino. Čerstvý vítr do této oblasti přinesl jednodeskový počítač Raspberry Pi, který si vydobyl neotřesitelnou pozici v tomto segmentu. Dnes je aktuální jeho třetí derivát Raspberry Pi 3, se kterým srovnám zvolený počítač BeagleBone Black.

Oba přístroje jsou na podobné úrovni, ale každý vyniká v jiné oblasti. Raspberry Pi má výrazně vyšší grafický výkon, tudíž je vhodný na veškeré multimediální aplikace. Jeho velkou výhodou je obrovská komunita uživatelů a dostupná podpora na internetu. BeagleBone má podporu také snadno dostupnou, ve srovnání s Raspberry Pi jde však o zlomek informací. BeagleBone Black naopak vyniká v množnostech dalších rozšíření. Možnosti různých sběrnic a pinů GPIO jsou výrazně širší než u Raspberry Pi, a tak se BeagleBone stává vhodnější pro embedded aplikace, kdy je možné sestavit složitější zařízení. Osobně na mě Raspberry Pi působí méně profesionálně a pro komerční účely mi přijde vhodnější BeagleBone, ač je nepatrně dražší. [20] [19]

4.3 Operační systém

Softwarovým základem každého počítače je operační systém. Pro BeagleBone Black lze použít operační systémy s jádrem Linux nebo s jádrem Windows CE. Troufnu si tvrdit, že drtivá většina uživatelů si zvolí operační systém Linux, neboť licence Windows jsou zpoplatněny. V této práci se zabývám pouze operačním systémem Linux.

Linux je operační systém unixového typu s volně dostupnými zdrojovými kódy (open-source). To je v oblasti návrhu velká výhoda, neboť uživatel může na systému cokoli změnit a upravit ho tak podle svých přání. Linux umožňuje multitasking (běh více úloh najednou) a připojení více uživatelů k systému. Operační systém je šířen v distribucích. Pro potřeby počítače BeagleBone Black je vhodné využít některou již sestavenou distribuci pro malé počítače. Do této skupiny patří hned několik distribucí: [21],[18]

- Distribuce Android je populární operační systém určený zejména pro mobilní telefony nebo tablety. Existuje verze určená pro BeagleBone Black, ta je však bez oficiální podpory, neboť ji vytvářela jen skupina fanoušků. [22]
- Distribuce Angstrom je určena pro malá embedded zařízení. Jedná se o uživatelsky přívětivý operační systém oblíbený pro své diagnostické vlastnosti. [23]
- Distribuce Debian je u desek BeagleBone nejčastěji používaná. Vyniká svou stabilitou a uživatelskou přívětivostí. Díky své rozšířenosti má Debian velmi širokou a jednoduše dostupnou podporu. Za výhodu lze považovat i předinstalované grafické prostředí. V případě potřeby je možné provést upgrade distribuce Debian na distribuci Machinekit. Tato distribuce není tak obecná jako Debian a je využívána pro řízení strojů, díky přítomnosti prostředí pro řízení v reálném čase. [24]

Po zvážení pro a proti jednotlivých výše uvedených distribucí jsem se rozhodl použít distribuci Debian.

4.4 Nastavení

Pro správné fungování hodinové ústředny je nejprve nutné BeagleBone Black správně nakonfigurovat. Tato část práce může sloužit jako obecný návod pro práci s BeagleBone Black.

4.4.1 Instalace operačního systému

Z oficiálních internetových stránek výrobce¹ je nutné stáhnout obraz operačního systému Debian, který je určen pro zařízení BeagleBone Black. Z tohoto souboru vytvoříme bootovatelnou SD paměťovou kartu, která musí mít kapacitu nejméně 4 GB. Ke správnému naformátování paměťové karty slouží například utilita Etcher, která je volně dostupná na internetu². Bootování systému z paměťové karty provedeme následujícím způsobem: do vypnuté desky BeagleBone Black vsuneme paměťovou kartu a během připojení napájení k desce budeme mít stisknuté tlačítko *BOOT BUTTON* (viz. obrázek 4.1). [25]

Pro komunikaci s BeagleBone Black je možné využít protokolu SSH. K tomu musíme spojit desku s počítačem pomocí Ethernetového nebo USB kabelu a stáhnout si utilitu, které slouží jako SSH klient — například program Putty³. Zařízení je dostupné na IP adrese *192.168.7.2*. Po připojení zařízení je vyžadováno přihlášení. Základní uživatelské

¹ <http://beagleboard.org/latest-images>

² <https://etcher.io/>

³ <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

jméno je *debian* a základní heslo je *temppwd*. Jakmile je uživatel úspěšně přihlášen, může ověřit verzi svého operačního systému pomocí příkazu *uname -a*. [18][25]

Některé činnosti v systému vyžadují vyšší oprávnění. V takovém případě je potřeba přihlásit se jako tzv. superuživatel. K tomu slouží příkaz *sudo su*. Opět je vyžadováno heslo, které je opět *temppwd*.

4.4.2 Konfigurace operačního systému

V některých případech je důležité přenastavit konfiguraci spouštění operačního systému. Ta se definuje v souboru */boot/uEnv.txt*. Zde můžeme upravit několik zásadních aspektů, soubor totiž určuje, jak se BeagleBone Black spustí a nakonfiguruje při bootování samotného systému. Tento soubor je možné otevřít v kterémkoli textovém editoru. Pro vypálení obrazu operačního systému z paměťové karty do integrované paměti eMMC je potřeba v souboru *uEnv.txt* odstranit znak označující komentář před následujícím řádkem:

```
##enable BBB: eMMC Flasher:
#cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh
```

a v souboru zanechat zápis:

```
##enable BBB: eMMC Flasher:
cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh
```

Pomocí příkazu *sudo reboot* pak restartujeme systém. Ten při následném zapnutí začne vypalovat obraz z paměťové karty do integrované eMMC paměti. Tento proces může trvat několik desítek minut.

Protože jsou některé piny procesoru alokované pro určitou funkci (např. pro obsluhu HDMI), může docházet ke konfliktu s vlastním využitím těchto pinů. Přehled funkcí jednotlivých pinů je k dispozici v tabulkách v příloze C. Alokované funkce je možné deaktivovat v souboru *uEnv.txt*. Tam je již připravená sekvence řádků, kdy každý z nich ovládá jinou alokovanou funkci a pomocí odstranění znaku komentáře u příslušného řádku konkrétní funkci deaktivujeme:

```
#disable_uboot_overlay_emmc=1
disable_uboot_overlay_video=1
disable_uboot_overlay_audio=1
#disable_uboot_overlay_wireless=1
#disable_uboot_overlay_adc=1
```

V případě mé práce bylo nutné deaktivovat funkci pro obsluhu video (HDMI) a audio pinů, protože je využívám jako standardní GPIO piny. Řádky s příslušnou deaktivací jsem tak nechal bez znaku komentáře.

4.4.3 Konfigurace GPIO a jiných periférií

Konfigurovat jednotlivé piny GPIO lze buď přímo z operačního systému, nebo již při bootování za pomoci tzv. Device Tree. Každý pin je označen jedinečným číslem, které lze vypočítat podle následujícího vzorce:

$$GPIO_{xyy} = (x \cdot 32) + yy = GPIO_{zz} \quad (1)$$

$GPIO_{xyy}$ je označení GPIO pinu dle výrobce, $GPIO_{zz}$ je označení GPIO pinu v souborovém systému operačního systému. Například pin na pozici konektoru P9-14 je označen jako $GPIO1[18]$. Po provedení uvedeného výpočtu dostaneme tvar $GPIO50$. V tomto ohledu je velmi nápomocna tabulka v příloze C. [18]

Pokud zůstaneme u pinu *GPIO50* a budeme ho chtít nakonfigurovat např. jako výstupní pin s logickou úrovní *log 0*, budeme postupovat podle následující sekvence příkazů napsaných do příkazové řádky:

```
echo 50 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio50/direction
echo 0 > /sys/class/gpio/gpio50/value
```

První příkaz nám vyexportuje tento pin a dovolí nám jeho používání. Druhý příkaz určí, zda se jedná o vstup nebo výstup pomocí argumentů *in / out*. Třetí příkaz pak určuje logickou hodnotu výstupu. [18]

Další možností, jak správně nakonfigurovat používané piny, a jedinou možností, jak konfigurovat další periferie (např. datové sběrnice), je sestavení souboru zvaného Device Tree. Jedná se o textový soubor, který popisuje hardwarovou konfiguraci procesoru. Obsahuje seznam konfigurací všech pinů a periférií pro správnou funkčnost vytvořené Cape, která je připojena k desce BeagleBone Black. Konfigurace pinu se definuje podle jeho adresy a 7 bitového kódu, který je převeden do hexadecimální soustavy. Následující tabulka ukazuje význam 7bitové kódu:

GPIO Settings				
Bit 6	Bit 5	Bit 4	Bit 3	Bit 2,1,0
Slew Control	Receiver Active	Pullup/Pulldown	Enable Pullup/Pulldown	Mux Mode
0 Fast	0 Disable	0 Pulldown select	0 Enabled	000 Mode 0 to
1 Slow	1 Enable	1 Pullup select	1 Disabled	111 Mode 7

e.g. OUTPUT GPIO(mode7) 0x07 pulldown, 0x17 pullup, 0xf no pullup/down
e.g. INPUT GPIO(mode7) 0x27 pulldown, 0x37 pullup, 0xf no pullup/down

Obrázek 4.2. Tabulka pro nastavení funkce pinů GPIO [18]

Je tedy možné definovat mód, ve kterém pin pracuje, dále zda bude mít pin pull-up rezistor, pull-down rezistor nebo nebude mít žádný rezistor, a zda se bude jednat o vstup nebo výstup. Pro návod, jak správně sestavit Device Tree, čtenáře odkážu na knihu Dereka Molloye [18] a na jeho internetový blog¹, kde je vše přehledně vysvětleno a pochopení celé oblasti usnadní přiložené video.

Sestavený soubor Device Tree s příponou **.dts* je potřeba přeložit do binární verze. K tomu slouží příkaz:

```
dtc -O dtb -o DEVICE-TREE-00A0.dtbo -b 0 -@ DEVICE-TREE.dts
```

Přeložený soubor přesuneme do složky */lib/firmware*. V této složce se kromě našeho nahraného souboru nachází další již připravené Device Tree. Jedná se o soubory Device Tree, které spouští různé periferie, například UART, I2C nebo SPI. Pokud chceme některý z uvedených Device Tree spustit při bootování systému, musíme opět upravit soubor *uEnv.txt*. Tentokrát do souboru zapíšeme, které Device Tree se mají spouštět:

```
uboot_overlay_addr0=/lib/firmware/DEVICE-TREE.dtbo
uboot_overlay_addr1=/lib/firmware/UART1.dtbo
uboot_overlay_addr2=/lib/firmware/I2C2.dtbo
uboot_overlay_addr3=/lib/firmware/<file3>.dtbo
```

Takto pozměněný zápis zajistí, že se při spuštění systému načte náš Device Tree, UART1 a I2C2. Umístění jednotlivých sběrnic je možné zjistit z tabulek v příloze C.

V tento moment máme BeagleBone Black správně nakonfigurovaný a můžeme pokračovat k psaní softwaru.

¹ <http://derekmolloy.ie/gpios-on-the-beaglebone-black-using-device-tree-overlays/>

Kapitola 5

Software hodinové ústředny

V této kapitole je vysvětleno, jak se programově obsluhují základní rutiny pro chod hodinové ústředny. Nejprve jsou demonstrovány hlavní výhody použití linuxového počítače namísto dříve používaného mikrokontroléru. V druhé části kapitoly je popsán program řízení hodinové ústředny.

5.1 Synchronizace času

Velké zjednodušení při realizaci hodinové ústředny nám umožní aplikace běžící na pozadí operačního systému. Pomocí nich je možné synchronizovat systémový čas a ten již pouze volat z programu. Odpadá tím nutnost poměrně složitých přerušení v programu, kdy je nutné zařídit, aby procesor v momentě synchronizace času s jedním ze zdrojů časového normálu nebyl vytížen jinými instrukcemi.

Pro synchronizaci času na pozadí je stěžejní správný chod aplikace Chrony, která se stará o veškeré záležitosti v oblasti času. Tato aplikace umožňuje používat jako zdroj časového normálu všechny tři výše zmiňované možnosti a sama si z nich vybírá v daný moment nejpřesnější zdroj. Možnost synchronizace času pomocí NTP serverů umožňuje Chrony sám od sebe, ke čtení signálu z GPS a DCF77 jsou potřeba další pomocné aplikace, jak bude vysvětleno dále.

Dojde-li k výpadku zdroje časového normálu, přechází Chrony automaticky na jiný použitelný zdroj. V případě výpadku všech zdrojů času najednou aplikace Chrony sama komunikuje s hodinami reálného času DS3232. Aplikace Chrony tedy využívá přesnou časovou referenci po dobu trvání výpadku zdrojů časového normálu a zajišťuje správný chod ústředny. Tím je vyřešena otázka nouzového provozu při výpadku jednoho nebo více zdrojů časového signálu.

5.1.1 Další nastavení BeagleBone Black

Jak již bylo zmíněno, synchronizaci času pomocí NTP serveru umožňuje Chrony v již v základní konfiguraci. Pro správné získání času z GPS a DCF77 je však nutná implementace dalších aplikací běžících na pozadí. Pro GPS jsou to aplikace *GPSTOOLS*, *GPSTOOLS-CLIENTS* a *PPS-TOOLS*, pro DCF77 aplikace *RADIOCLK*. V této práci je popsána pouze instalace aplikací pro obsluhu GPS. Analogicky je možné instalovat také aplikaci pro obsluhu DCF77.

Pro zahájení implementace aplikací pro synchronizaci času je nejprve nutné stáhnout aplikaci Chrony a správně ji nainstalovat. K tomu slouží následující příkazy:

```
git clone git://git.tuxfamily.org/gitroot/chrony/chrony.git
./configure
make
make install
```

V dalším kroku je potřeba zařídit spuštění aplikace již při startu systému úpravou souboru *chronyd.service* a správně upravit konfigurační soubor, ve kterém Chrony zjišťuje, jaké úkony má vykonávat. To se provede podle následující dvojice příkazů:

```
nano /etc/chrony.conf
nano /lib/systemd/system/chronyd.service
```

Při tvorbě konfiguračního souboru je vhodné nastudovat dokumentaci k aplikaci na webových stránkách Chrony¹.

Dále je nutné pomocí následujících příkazů stáhnout již zmíněné aplikace:

```
apt-get install gpsd
apt-get install gpsd-clients
apt-get install pps-tools
```

Následuje vytvoření Device Tree pro obsluhu PPS (Pulse Per Second) a nastavení jeho načítání již při bootování. Návod pro vytváření Device Tree je uveden v kapitole 4.4.3 a Device Tree vytvořený pro obsluhu PPS je k nalezení v příloze D. Spolu s tímto Device Tree je nutné ke spuštění při bootování přidat již připravené Device Tree pro obsluhu sériové linky UART.

Pomocí příkazu `nano /lib/systemd/system/gpsd.service` vytvoříme textový soubor, který zařídí spuštění služby při každém zapnutí. Poslední nutností je úprava souboru `/etc/default/gpsd`, kde do kolonky `DEVICE`: přiřadíme příslušný UART pro čtení dat z GPS.

Výše popsané úpravy zajistí správný chod aplikace Chrony a uvedených aplikací pro obsluhu GPS. Ten je možné ověřit pomocí následujících příkazů:

```
systemctl status chronyd/gpsd //ukáže stav aplikace v pozadí
cgps /dev/ttyS1 //vypíše veškerá data přijímaná z GPS
grep ocp /proc/interrupts //vypíše přerušování PPS
ppstest /dev/pps1 //vypíše příchozí sekundy PPS
chronyc
--> sources //vypíše zdroje časového normálu pro synchronizaci
--> tracking //vypíše podrobné údaje o přijímaném signálu
```

Při správné konfiguraci aplikace Chrony a provedení výše uvedených kroků se systémový čas synchronizuje buď pomocí GPS nebo NTP, bez nutnosti dále zasahovat do procesů synchronizace. Analogickým způsobem instalace aplikace `RADIOCLK` lze přidat také synchronizaci pomocí signálu DCF77.

Tento postup řešení má řadu dalších výhod. Aplikace Chrony umožňuje při synchronizaci času pomocí GPS použít hodinovou ústřednu také jako NTP server (nikoli pouze jako NTP klient). Vyžaduje to pouze úpravu konfiguračního souboru. Další výhodou je možnost nastavit v operačním systému Debian místní časové pásmo, kdy se přijímaný čas UTC automaticky převádí do místního času. Tím odpadá starost o řízení přepínání mezi letním a zimním časem. Nastavení místního časového pásma se provede pomocí příkazu:

```
dpkg-reconfigure tzdata
```

5.2 Obsluha hodinové linky

V této kapitole jsou popsány pouze základní rutiny obsluhy, neboť výsledná konfigurace hodinové ústředny zatím není známa. Po odevzdání práce v lednu 2018 se uskutečnilo jednání mezi společností ELTRON s.r.o. a koncovým zákazníkem, kde se dohodnou na podrobnosti výsledné konfigurace hodinové ústředny. Program je tak omezen na zadávání údajů od uživatele pomocí konzole operačního systému Debian. Program dále řídí hodinové linky s funkcí automatického nastavení času na lince.

¹ <https://chrony.tuxfamily.org/documentation.html/>

■ 5.2.1 Ovládní pinů GPIO z programu

V celém kódu je použito ovládní pinů GPIO z programu. Z následujících řádků je patrné, jakým způsobem dochází k jejich ovládní. Nejprve je uvedena inicializace pinů GPIO, v dalším kroku pak nastavování hodnoty pinů GPIO:

```
int LED1_fd = -1;
static void gpio_init(void)
{
    int res, fd;
    char buf[255];
    // enable all GPIO banks
    res = chmod("/sys/class/gpio/export", S_IWGRP);
    fd = open("/sys/class/gpio/export", O_WRONLY);
    res = write(fd, "50", 3);
    close(fd);
    // set pin direction
    fd = open("/sys/class/gpio/gpio50/direction", O_WRONLY);
    write(fd, "out", 3);close(fd);
    // init pin write VALUE descriptors
    LED1_fd = open("/sys/class/gpio/gpio50/value", O_WRONLY);
}
```

```
void gpio_set(int fd, int value)
{
    if(value == 0)
        write(fd, "0", 1);
    else
        write(fd, "1", 1);
}
gpio_init();
gpio_set(LED1_fd, 1); //příklad využití
```

■ 5.2.2 Začátek programu

K zadání času, na který je nastavena hodinová linka, dochází pouze na začátku programu z konzole operačního systému Debian. Po získání informací o stavu linky program začíná nekonečný cyklus. Na začátku každého cyklu program získá lokální čas systému, který je již synchronizovaný s časovým normálem, a ihned ho porovná s časem nastaveným na lince. Zde uvedená činnost programu je zapsaná takto:

```
int main(int argc, char** argv)
... //deklarace proměnných
printf("pocet hodin na lince:");
scanf("%d", &lm_hour); //získiná dat o počtu hodin nastavených na lince
printf("pocet minut na lince:");
scanf("%d", &lm_min); //získiná dat o počtu minut nastavených na lince
while (1) {
    time_t t = time(NULL);
    struct tm tm = *localtime(&t);
    if (lm_hour == tm.tm_hour && lm_min == tm.tm_min)
```

■ 5.2.3 Řízení hodinové linky

Podmínka na konci kódu uvedeného v kapitole 5.2.2 dělí program na dvě části. Pokud je čas nastavený na lince stejný jako reálný čas v daný okamžik (tzn. že hodiny ukazují

správný čas), program začne kontrolovat, zda již nastala 59. vteřina minuty. V 59. vteřině minuty se posílá buď sudý nebo lichý pulz podle toho, která minuta právě běží, a čas na hodinách se tak posouvá o jednu minutu. V každé jiné vteřině minuty program projde až na konec stále se opakujícího cyklu, opět načítá systémový čas a porovnává ho s časem na hodinové lince.

```

if (lm_hour == tm.tm_hour && lm_min == tm.tm_min) //počáteční podmínka
{
if (tm.tm_sec == 59) //při 59. vteřině se postupuje dále
{ //při jiných vteřinách se jde na začátek programu
if (tm.tm_min % 2 == 0) //pro sudé minuty se posílá sudý pulz
{
gpio_set(MP3S_fd, 1);
usleep(1000000);
gpio_set(MP3S_fd, 0);
}
else //pro liché minuty se posílá lichý pulz
{
gpio_set(MP3L_fd, 1);
usleep(1000000);
gpio_set(MP3L_fd, 0);
}
lm_min++; //aktualizace času nastaveného na hod. lince
if (lm_min == 60)
{
lm_hour++;
lm_min = 0;
if (lm_hour == 24)
{
lm_hour = 0;
}
}
}
}
}
}

```

■ 5.2.4 Řízení dobíhání hodin

Pokud podmínka uvedená na konci kódu v kapitole 5.2.2 není splněna, tedy čas nastavený na hodinové lince neodpovídá reálnému času v daný moment, hodiny přejdou do režimu, ve kterém linka dohání reálný čas. Na začátku je uveden příkaz pro převedení do dvanáctihodinového cyklu. Dále je spočítaný počet pulzů, které linka obdržela od půlnoci a počet pulzů, které odpovídají reálnému času od půlnoci. Porovnáním těchto hodnot je zjištěno, kolik pulzů je nutné hodinám poslat, aby opět ukazovaly správný čas. Tento počet pulzů se v následném kroku hodinám dodá. V některých případech je rychlejší nechat linku čekat až se její čas bude rovnat reálnému času, než ji dohánět. Tato situace nastává v případě, kdy je čas na lince o 23 minut nebo méně napřed oproti času reálnému. Doba čekání je v tomto případě kratší než doba, kterou by trvalo dohánění linky. V ostatních případech je výhodnější linkou dohánět skutečný čas.

```

else
{
if (tm.tm_hour > 12){
tm_hod = tm.tm_hour - 12;
}else{

```

```

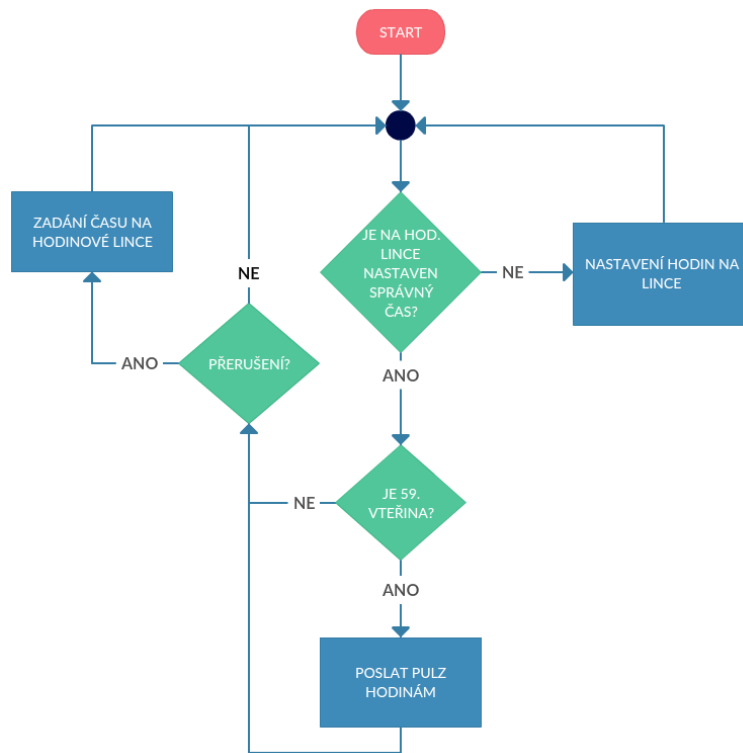
tm_hod = tm.tm_hour;
}if (lm_hour > 12){
lm_hod = lm_hour - 12;
}else{
lm_hod = lm_hour;
}
rcas = ((tm_hod * 60) + tm.tm_min);
lcas = ((lm_hod * 60) + lm_min);
cas_dif = rcas - lcas;
if (cas_dif > 0){
p_pulzu = cas_dif;
}else{
p_pulzu = 720 + cas_dif;
}
if (p_pulzu > 696)
{}
else
{
for (size_t i = 0; i < p_pulzu; i++)
{
if (lm_min % 2 == 0){
gpio_set(MP3S_fd, 1);
usleep(1000000);
gpio_set(MP3S_fd, 0);
}else{
gpio_set(MP3L_fd, 1);
usleep(1000000);
gpio_set(MP3L_fd, 0);
}
lm_min++;
if (lm_min == 60){
lm_hour++;
lm_min = 0;
if (lm_hour == 24){
lm_hour = 0;
}
}
}
usleep(1000000);
}
}
}
}

```

Tímto programem je zařízen autonomní chod ústředny. Pro větší přehlednost napsaných kódů je v příloze E uveden celý program formou screenshotů z vývojového prostředí Visual Studio.

5.3 Výsledný software

Po dohodě o výsledné konfiguraci hodinové ústředny s koncovým zákazníkem bude vyvinut obsluhující software, který bude postaven na univerzálně použitelných základech uvedených v této kapitole. Na obrázku 5.1 je znázorněn průběh takového softwaru.



Obrázek 5.1. Vývojový diagram softwaru

Kapitola 6

Výsledný produkt

6.1 Volitelné konfigurace

Koncept hodinové ústředny umožňuje sestavit různé konfigurace výsledného produktu, aniž by bylo potřeba měnit hardwarový základ. Je pouze na volbě zákazníka, kterou konfiguraci si zvolí. Jedná se o výrobu na zakázku a o volném prodeji se zatím neuvažuje. Při výběru vhodné konfigurace je však nutné přihlídnout k některým omezením. Taková omezení mohou nastat při výběru zdroje či zdrojů časového normálu nebo při volbě ovládání hodinové ústředny.

Prvním limitujícím faktorem je špatná přístupnost zařízení. Pokud bude výrobek umístěn v místě, kam se lze těžko dostat, bude nevhodné zvolit ovládání pomocí dotykového displeje. V takové situaci je vyhovující řešení s ovládáním přes webové rozhraní.

Omezení způsobí také nedostupná internetová síť nebo nepřivedený ethernetový kabel v místě umístění ústředny, a to ze dvou důvodů. Za těchto podmínek není možné systém ovládat skrz webové rozhraní a jediným řešením je dotykový displej. Nepřítomnost internetového připojení také zabraňuje použití NTP serveru jako zdroje časového normálu.

Zbylé dva zdroje časového normálu jsou naopak závislé na dostatečně kvalitním příjmu signálu. Přijímací modul GPS by měl být umístěn v těsné blízkosti okna nebo ideálně mimo budovu s dostatečným výhledem na oblohu. Takto umístěný GPS modul je pak použitelný kdekoliv na světě. Příjem DCF77 je teoreticky možný i uvnitř budov, nicméně pokud budova stíní radiové vlny, může být příjem problematický. Z tohoto důvodu není doporučeno přijímač instalovat do útrob budovy. S rostoucí vzdáleností od vysílače ve Frankfurtu nad Mohanem navíc intenzita signálu slábne a příjem je o to komplikovanější.

Pokud odhlédneme od výše zmíněných omezení, má zákazník možnost poskládat si zařízení odpovídající jím zvolené aplikaci. Na výběr má ze tří různých zdrojů časového normálu, přičemž je možné kombinovat více zdrojů najednou, což zajišťuje ochranu proti výpadku jednoho ze zdrojů. Pokud si zákazník vybere GPS nebo DCF77, bude ústředna dodána i s příslušným přijímačem. Kromě zdroje časového normálu tvoří další nedílnou součást hodinové ústředny výstupní hodinové linky. Ty je podle rozsahu aplikace možné zvolit až tři.

Zcela dobrovolná je možnost implementace dalších nabízených částí. Zákazník může využít spínaného výstupu. V případě potřeby komunikace s dalšími zařízeními je možné ústřednu doplnit sériovými linkami RS232 a RS485. K dispozici je také zálohování přesného času po dobu výpadku pomocí integrovaného obvodu DS3232 s přesným teplotně kompenzovaným krystalem. Možné je i použití hodinové ústředny jako NTP serveru. Možnosti konfigurace jsou shrnuty v tabulce 6.1.

Podle vybrané konfigurace bude deska hodinové ústředny osazena příslušnými součástkami a zařízení bude vybaveno příslušným softwarem. Dle množství využitých funkcí bude také upravena cena výrobku s ohledem na náklady vynaložené při realizaci a na použité technologie. Tvorba ceny však již není v kompetenci autora.

Povinná výbava:	
Zdroje časového normálu:	GPS DCF77 NTP server
Počet výstupních hodinových linek:	max. 3x
Způsob ovládání:	dotykový displej webové rozhraní
Volitelná výbava:	
	spínaný obvod sériová linka RS232 sériová linka RS485 zálohování přesného času (obvod DS3232) NTP server

Tabulka 6.1. Možné konfigurace hodinové ústředny.

6.2 Ověření funkčnosti

Ověření funkčnosti vyrobeného zařízení bylo vykonáno ve dvou krocích. Nejprve byla kontrolována pouze vyrobená deska plošných spojů bez připojení k počítači BeagleBone Black, později již hotové zařízení jako celek. V následujících odstavcích je popsán průběh testování.

Samostatná deska plošných spojů byla připojena k laboratornímu zdroji s napětím 5 V a 9 V. Z dalšího napájecího zdroje bylo odebíráno napětí 3,3 V a pomocí tohoto napětí byly k pinům na konektorech připojovány logické hodnoty stejným způsobem, jako je v běžném provozu připojuje procesor. Tímto způsobem byla ověřena funkčnost výstupů hodinových linek, signalizačních LED diod, přívodu napájení modulu GPS a spínaného výstupu. V rámci tohoto testování nebyla odhalena žádná chyba.

Druhá část ověřování byla provedena ovládním pinů z konzole operačního systému Debian na počítači BeagleBone Black. Postup pro ovládní pinů procesoru GPIO a používání sběrnic je popsán v kapitole 4.4.3. Tímto způsobem bylo opět možné ověřit funkčnost všech výše uvedených částí a především správnou komunikaci vytvořené desky plošných spojů s počítačem BeagleBone Black. V této části ověřování byl objeven konflikt funkcí na pinech 45 a 46 na konektoru P8. Bližší informace k problému i postup jeho řešení je uveden v kapitole 3.4.

Funkčnost hardwaru výsledného zařízení byla ověřena ve dvou fázích a je tedy možné konstatovat, že zařízení bude po implementaci softwaru připraveno k případnému prodeji.

Kapitola 7

Závěr

Cílem mé práce bylo seznámit se se způsobem, jakým je možné vzdáleně řídit elektronické hodiny, a následně provést návrh řídicí jednotky, která by obsluhovala hodinové linky a řídila tak elektronické hodiny. Časový normál měl být získáván pomocí služeb DCF, GPS a NTP serveru. Hlavním cílem této práce bylo celý projekt realizovat dle zadání a prověřit jeho vlastnosti.

Ve druhé kapitole jsem popsal parametry navržené hodinové ústředny. Uvedl jsem zde, jaké typy elektronických hodin existují a popsal jsem možnosti jejich vzdáleného řízení. Pro pochopení kontextu získávání časového normálu byly uvedeny základní informace o jednotlivých službách využívaných k jeho získávání. Také jsem zde teoreticky popsal funkce, jimiž je navržené zařízení vybaveno. V poslední části jsou diskutovány možné oblasti využití a porovnání hodinové ústředny s podobnými výrobky konkurence.

Detaily jednotlivých hardwarových řešení jsem přiblížil v následující kapitole. U každého funkčního bloku ústředny jsem uvedl základní vysvětlení jeho funkce a text doplnil obrazovými přílohami z programu EAGLE, na kterých je možné prohlédnout si podrobnosti zapojení. Na závěr této kapitoly jsem umístil fotografii již vyrobené a osazené desky plošných spojů hodinové ústředny.

Ve čtvrté kapitole práce jsem se zabýval popisem jednodeskového počítače BeagleBone Black. Nejprve jsem zařízení představil a uvedl jeho základní parametry. Následně jsem ho porovnal s konkurenčním výrobkem Raspberry Pi. Dále jsem se vyjádřil k tomu, proč jsem se rozhodl právě pro BeagleBone Black. V praktické části kapitoly jsem popsal instalaci operačního systému Debian do BeagleBone Black a jeho následnou konfiguraci. Vysvětlil jsem také, jak ovládat piny GPIO.

V páté kapitole věnované softwaru jsem přiblížil programovou obsluhu hodinové ústředny. Zabýval jsem se synchronizací času na pozadí pomocí aplikace Chrony, která je zároveň schopná zajistit nouzový chod zařízení. Uvedl jsem praktický návod k instalaci rozšiřujících aplikací pro získávání časového normálu pomocí GPS. V další části jsem rozebral jednotlivé bloky programu pro řízení hodinové ústředny. K tomu jsou přidány části kódů programu pro lepší představu o jeho funkci.

V závěrečné kapitole jsem se věnoval popisu možností konfigurace hodinové ústředny. Do tabulky 6.1 jsem shrnul možnou výbavu výrobku. V poslední části jsem se zabýval ověřením funkčnosti vyrobeného zařízení.

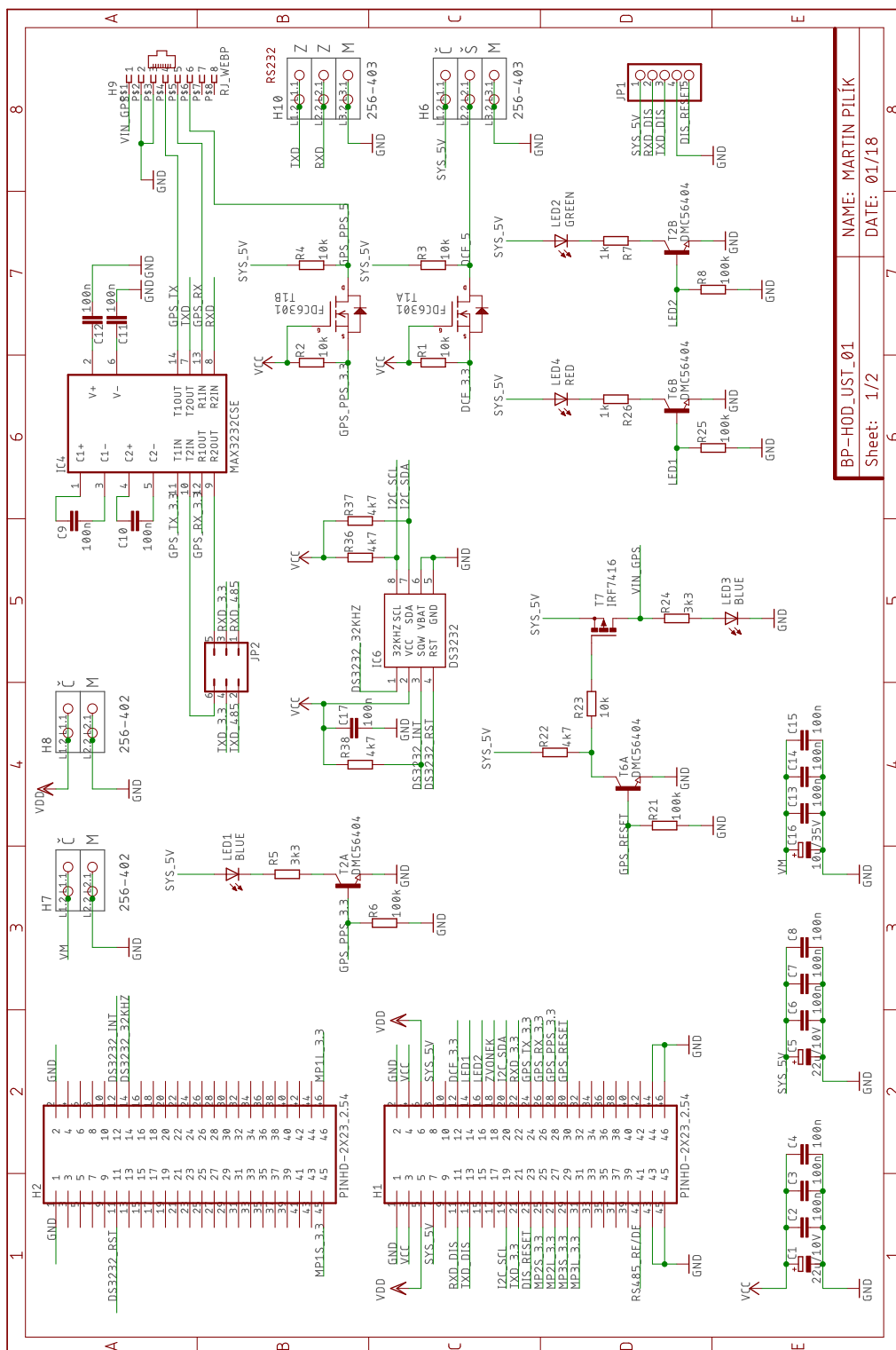
Výsledná hodinová ústředna splňuje hardwarové nároky žádané zadáním a rozšiřuje je o množství dalších funkcí, jako je například možnost ovládnutí školního zvonění nebo komunikace po sériových linkách. Troufám si tvrdit, že se mi podařilo vytvořit použitelné zařízení, které dobře poslouží jako prototyp k finálnímu výrobku. Případné změny se provedou jen v konstrukčních částech návrhu tak, aby zařízení bylo možné uschovat do krabičky a mělo tak potřebnou ochranu proti mechanickému poškození.

Kam se nyní bude projekt vyvíjet odkryje chystané jednání firmy ELTRON s.r.o. s koncovým zákazníkem. Podle jeho požadavků bude sestavena konfigurace hodinové ústředny a upraven příslušný software. Sepsáním práce pro mě projekt nekončí a budu s

firmou nadále spolupracovat. Do budoucna by pro firmu mohlo být zajímavé například navrhnout bezdrátovou synchronizaci elektronických hodin.

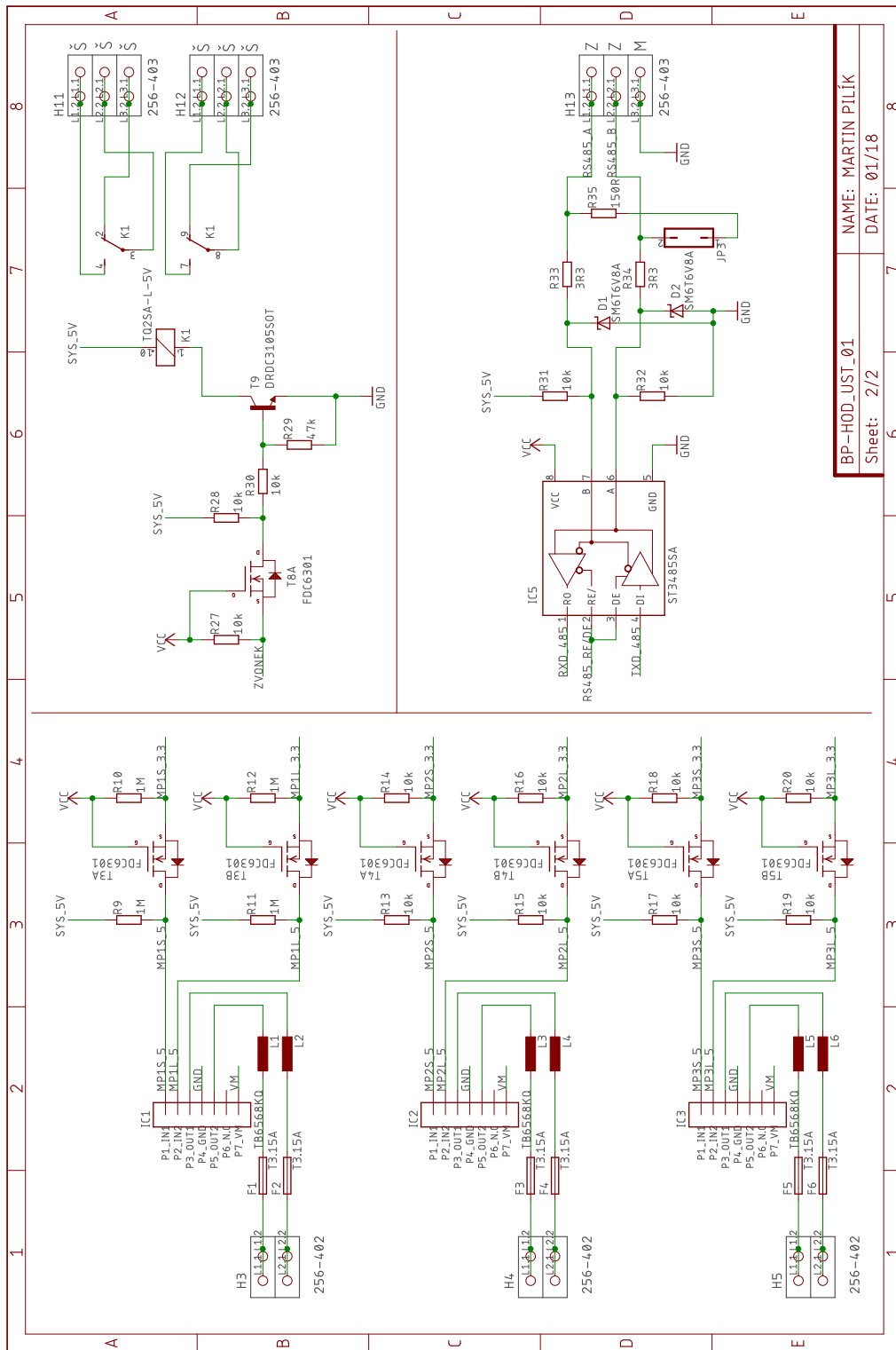
Příloha A

Schéma zapojení hodinové ústředny



BP-HOD_UST_01
 Sheet: 1/2
 NAME: MARTIN PILÍK
 DATE: 01/18

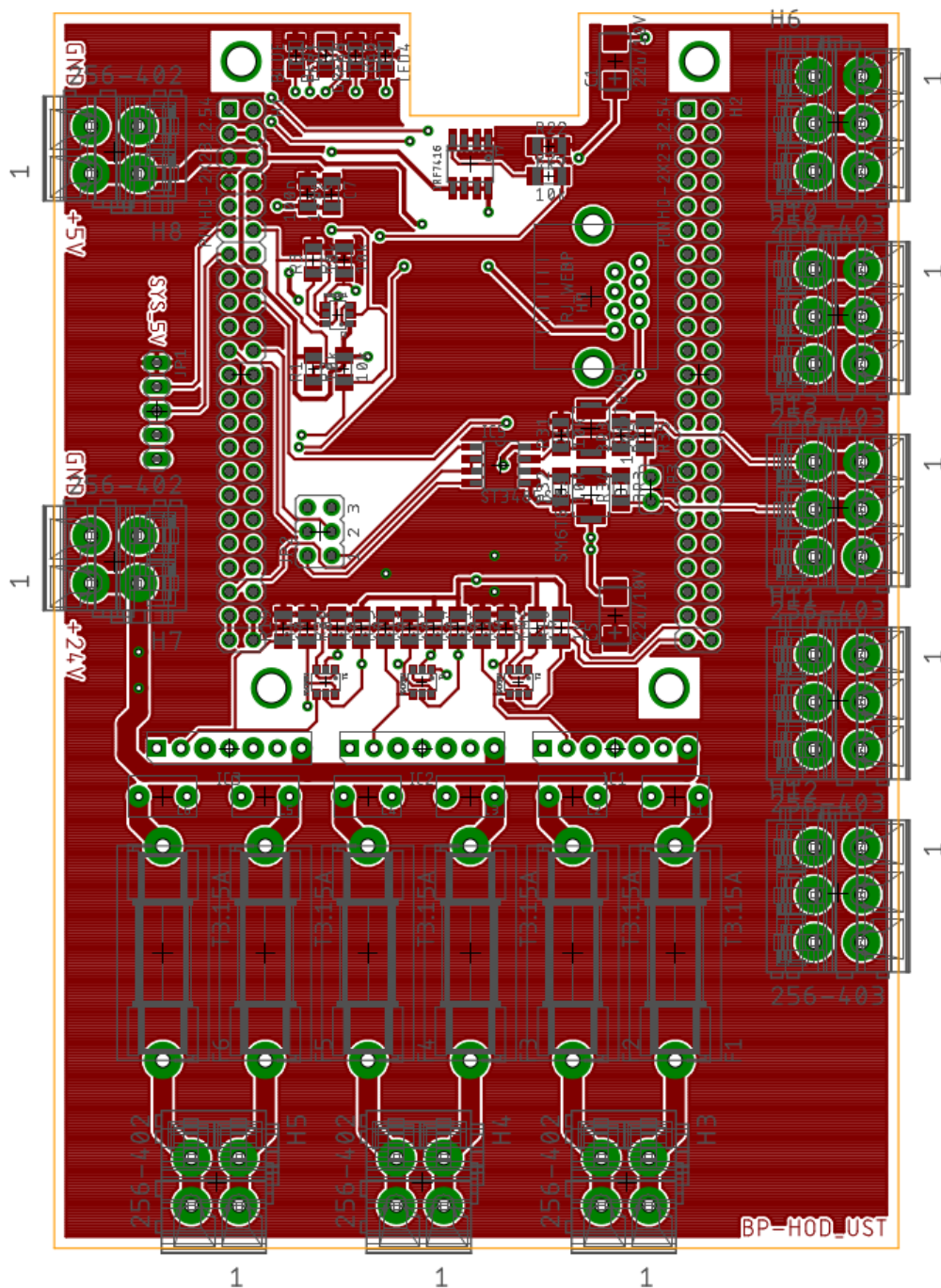
Obrazek A.1. Schéma zapojení hodinové ústředny 1/2



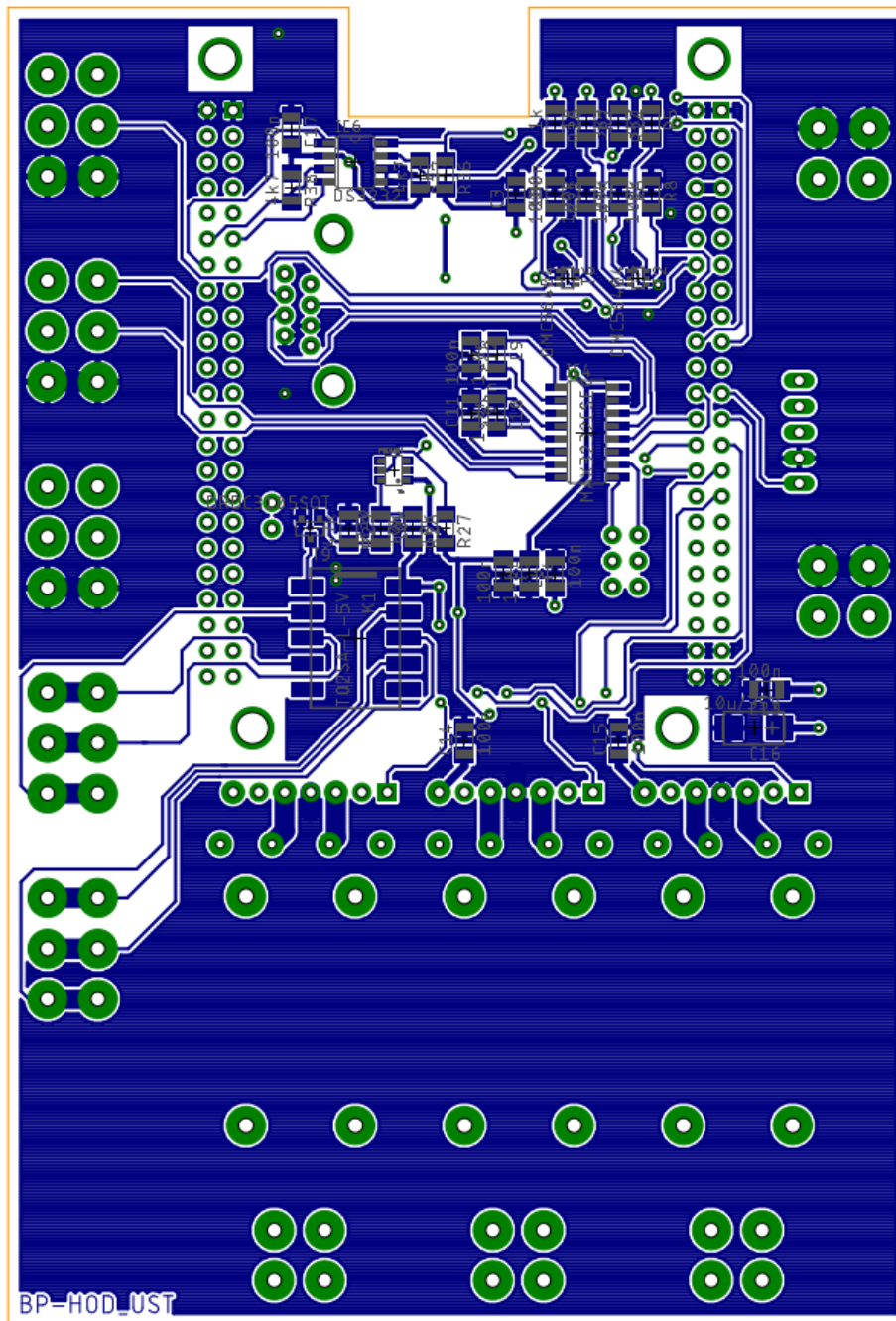
Obrázek A.2. Schéma zapojení hodinové ústředny 2/2

Příloha B

Ukázka desky plošných spojů



Obrázek B.3. Ukázka desky plošných spojů, strana TOP



Obrázek B.4. Ukázka desky plošných spojů, strana BOTTOM

Příloha C

Tabulky konfigurací funkcí konektorů P8 a P9

Pin	SPINS	ADDR	GPIO	Name	Mode6	Mode5	Mode4	Mode3	Mode2	Mode1	Modio0	CPU	Noise
P8_01		Offset from:		DGND									Ground
P8_02	6	0x8180018	38	GPIO1_6	gpio1[6]				mmc1_dat6	mmc1_dat6	gmic_ad6	R9	Allocated emmc2
P8_03	7	0x818001c	39	GPIO1_7	gpio1[7]				mmc1_dat7	mmc1_dat7	gmic_ad7	T9	Allocated emmc2
P8_04	2	0x8080008	34	GPIO1_2	gpio1[2]				mmc1_dat2	mmc1_dat2	gmic_ad2	R8	Allocated emmc2
P8_05	3	0x808000c	35	GPIO1_3	gpio1[3]				mmc1_dat3	mmc1_dat3	gmic_ad3	T8	Allocated emmc2
P8_06	36	0x8800080	66	TIMER4	gpio2[2]				timer4	timer4	gmic_adm_ale	R7	
P8_07	37	0x8840084	67	TIMER7	gpio2[3]				timer7	timer7	gmic_ain_ven	T7	
P8_08	39	0x88c008c	69	TIMER5	gpio2[5]				timer5	timer5	gmic_beln_cle	T6	
P8_09	38	0x8880088	68	TIMER6	gpio2[4]				timer6	timer6	gmic_wen	U6	
P8_10	38	0x8880088	68	TIMER6	gpio2[4]				timer6	timer6	gmic_wen	U6	
P8_11	13	0x8340034	45	GPIO1_13	gpio1[13]		eOEP2B_in	mmc2_dat1	mmc1_dat5	lcd_data18	gmic_ad13	R12	
P8_12	12	0x8300030	44	GPIO1_12	gpio1[12]		E0EP2A_IN	MMC2_DAT0	MMC1_DAT4	LCD_DATA19	gmic_ad12	T12	
P8_13	9	0x8240024	23	EHRPWM2B	gpio2[3]		ehpwm2B	mmc2_dat5	mmc2_dat5	lcd_data22	gmic_ad9	T10	
P8_14	10	0x8280028	26	GPIO0_26	gpio0[26]		ehpwm2_index_in	mmc2_dat6	mmc1_dat2	lcd_data21	gmic_ad10	T11	
P8_15	15	0x82c002c	47	GPIO1_15	gpio1[15]		eOEP2_strobe	mmc2_dat3	mmc1_dat7	lcd_data16	gmic_ad15	U13	
P8_16	14	0x8280028	46	GPIO1_14	gpio1[14]		eOEP2_index	mmc2_dat2	mmc1_dat6	lcd_data17	gmic_ad14	U13	
P8_17	11	0x8200020	27	GPIO0_27	gpio0[27]		ehpwm0_synco	mmc2_dat7	mmc1_dat8	lcd_data20	gmic_ad11	U12	
P8_18	35	0x88c008c	65	GPIO2_1	gpio2[1]		mmc2_clk	mmc2_clk	gmic_wait1	lcd_memory_clk	gmic_clk_muc0	V12	
P8_19	8	0x8200020	22	EHRPWM2A	gpio2[2]		ehpwm2A	mmc2_dat4	mmc1_gnd	lcd_data23	gmic_ad8	U10	
P8_20	33	0x8840084	63	GPIO1_31	gpio1[31]		pr1_pmu_pmu_r00_13	mmc1_cmd	mmc1_cmd	gmic_ba1n	gmic_cmd2	V9	Allocated emmc2
P8_21	32	0x8800080	62	GPIO1_30	gpio1[30]		pr1_pmu_pmu_r00_12	mmc1_clk	mmc1_clk	gmic_cmd1	gmic_cmd1	U9	Allocated emmc2
P8_22	5	0x8140014	37	GPIO1_5	gpio1[5]				mmc1_dat5	mmc1_dat5	gmic_ad5	V8	Allocated emmc2
P8_23	4	0x8100010	36	GPIO1_4	gpio1[4]				mmc1_dat4	mmc1_dat4	gmic_ad4	U8	Allocated emmc2
P8_24	1	0x8040004	33	GPIO1_1	gpio1[1]				mmc1_dat1	mmc1_dat1	gmic_ad1	V7	Allocated emmc2
P8_25	0	0x8000000	32	GPIO1_0	gpio1[0]				mmc1_dat0	mmc1_dat0	gmic_ad0	U7	Allocated emmc2
P8_26	31	0x87c007c	61	GPIO1_29	gpio1[29]						gmic_cmd0	V6	
P8_27	56	0x8d000d0	86	GPIO2_22	gpio2[22]		pr1_pmu_pmu_r00_8		gmic_a6	gmic_a6	lcd_vsync	U5	Allocated HDMI
P8_28	58	0x8d800d8	88	GPIO2_24	gpio2[24]		pr1_pmu_pmu_r00_10		gmic_a10	gmic_a10	lcd_pclk	V5	Allocated HDMI
P8_29	57	0x8d400d4	87	GPIO2_23	gpio2[23]		pr1_pmu_pmu_r00_9		gmic_a9	gmic_a9	lcd_hsync	R5	Allocated HDMI
P8_30	59	0x8dc00dc	89	GPIO2_25	gpio2[25]		pr1_pmu_pmu_r00_11		gmic_a11	gmic_a11	lcd_ac_bias_en	R6	Allocated HDMI
P8_31	54	0x8d800d8	10	UART5_CTSN	gpio0[10]		uart5_ctsn	mmcasp0_avr1	gmic_a18	gmic_a18	lcd_data14	V4	Allocated HDMI
P8_32	55	0x8dc00dc	11	UART5_RTSN	gpio0[11]		uart5_rtsn	mmcasp0_avr2	gmic_a19	gmic_a19	lcd_data15	T5	Allocated HDMI
P8_33	53	0x8d400d4	9	UART4_RTSN	gpio0[9]		uart4_rtsn	mmcasp0_ahcbx	eOEP1_strobe	eOEP1_strobe	lcd_data13	V3	Allocated HDMI
P8_34	51	0x8c000c0	81	UART3_RTSN	gpio2[17]		uart3_rtsn	mmcasp0_for	eOEP1B_in	eOEP1B_in	lcd_data11	U4	Allocated HDMI
P8_35	52	0x8c400c4	8	UART4_CTSN	gpio0[8]		uart4_ctsn	mmcasp0_ahcckr	ehpwm1B	ehpwm1B	lcd_data12	V2	Allocated HDMI
P8_36	50	0x8d800d8	80	UART3_CTSN	gpio2[16]		uart3_ctsn	mmcasp0_avr0	eOEP1A_in	eOEP1A_in	lcd_data10	U3	Allocated HDMI
P8_37	48	0x8d000d0	78	UART5_TXD	gpio2[14]		uart5_txd	mmcasp0_ackx	ehpwm1A	ehpwm1A	lcd_data8	U1	Allocated HDMI
P8_38	49	0x8d400d4	79	UART5_RXD	gpio2[15]		uart5_rxd	mmcasp0_bx	ehpwm0_synco	ehpwm0_synco	lcd_data9	U2	Allocated HDMI
P8_39	46	0x8bc00bc	76	GPIO2_12	gpio2[12]		pr1_pmu_pmu_r00_6		gmic_a6	gmic_a6	lcd_data6	T3	Allocated HDMI
P8_40	47	0x8c000c0	77	GPIO2_13	gpio2[13]		pr1_pmu_pmu_r00_7		gmic_a7	gmic_a7	lcd_data7	T4	Allocated HDMI
P8_41	44	0x8b000b0	74	GPIO2_10	gpio2[10]		pr1_pmu_pmu_r00_4		gmic_a4	gmic_a4	lcd_data4	T1	Allocated HDMI
P8_42	45	0x8b400b4	75	GPIO2_11	gpio2[11]		pr1_pmu_pmu_r00_5		gmic_a5	gmic_a5	lcd_data5	T2	Allocated HDMI
P8_43	42	0x8a800a8	72	GPIO2_8	gpio2[8]		ehpwm2_index_in	ehpwm2_index_in	gmic_a2	gmic_a2	lcd_data2	R3	Allocated HDMI
P8_44	43	0x8ac00ac	73	GPIO2_9	gpio2[9]		ehpwm0_synco	ehpwm0_synco	gmic_a3	gmic_a3	lcd_data3	R4	Allocated HDMI
P8_45	40	0x8a000a0	70	GPIO2_6	gpio2[6]		ehpwm2A	ehpwm2A	gmic_a0	gmic_a0	lcd_data0	R1	Allocated HDMI
P8_46	43	0x8a400a4	71	GPIO2_7	gpio2[7]		ehpwm2B	ehpwm2B	gmic_a1	gmic_a1	lcd_data1	R2	Allocated HDMI

Obrázek C.5. Přehled adres a módů na konektoru P8

Pin	SPINS	ADDR	GPIO	Name	Mod07	Mod06	Mod03	Mod04	Mod02	Mod01	Mod00	CPU	Notes
P9_01		44410000		GND									Ground
P9_02		Offset from:											Ground
P9_03		44410900	DC_3.3V										250mA Max Current
P9_04			DC_3.3V										250mA Max Current
P9_05			VDD_5V										1A Max Current
P9_06			VDD_5V										1A Max Current
P9_07			SYS_5V										250mA Max Current
P9_08			SYS_5V										250mA Max Current
P9_09			PMR_BUTTON										5V Level (pulled up PMIC)
P9_10			SYS_RESETn										5V Level (pulled up PMIC)
P9_11	28	0x670070	30	UART4_RXD	gpio030	uart4_rxd_mux2		mmc1_sdbd	gmic_csm4	mm2_crs	RESET_OUT	A10	
P9_12	30	0x678078	60	GPIO1_28	gpio128	mmc2_sdbd		gmic_dir	gmic_csm6	mm2_coi	gmic_wait0	T17	All GPIOs to 4-6mA output and approx. 8mA on input.
P9_13	29	0x674074	31	UART4_TXD	gpio031	uart4_txd_mux2		mmc2_sdbd	gmic_csm5	mm2_axer1	gmic_wait1n	U18	
P9_14	18	0x648048	50	ETHERPMM1A	gpio110	etherpmm1a_mux1		gmic_a18	gmic_csm5	mm2_axer1	gmic_waitn	U14	
P9_15	18	0x640040	48	GPIO1_18	gpio118	etherpmm1a_mux1		gmic_a18	gmic_csm5	mm2_axer1	gmic_waitn	U14	
P9_16	19	0x64c04c	51	ETHERPMM1B	gpio119	etherpmm1b_mux1		gmic_a19	gmic_csm5	mm2_axer1	gmic_waitn	U14	
P9_17	87	0x65015c	5	I2C1_SCL	gpio05	etherpmm1b_mux1		gmic_a19	gmic_csm5	mm2_axer1	gmic_waitn	U14	
P9_18	86	0x658158	4	I2C1_SDA	gpio04			pr1_uart0_bsd	I2C1_SCL	mm2_swp	sp0_d0	A16	
P9_19	95	0x67017c	13	I2C2_SCL	gpio013			pr1_uart0_rxd	I2C1_SDA	mm2_swp	sp0_d1	B16	
P9_20	94	0x678178	12	I2C2_SDA	gpio012			pr1_uart0_rxd	I2C2_SCL	mm2_swp	sp0_d1	D17	Allocated I2C2
P9_21	85	0x654154	3	UART2_TXD	gpio03	EMU3_mux1		pr1_uart0_rxd	I2C2_SDA	mm2_swp	sp0_d0	D18	Allocated I2C2
P9_22	84	0x650150	2	UART2_RXD	gpio02	EMU2_mux1		pr1_uart0_rxd	I2C2_SDA	mm2_swp	sp0_d0	B17	
P9_23	17	0x644044	49	GPIO1_17	gpio117	etherpmm0_syncro		gmic_a17	I2C2_SDA	mm2_rxdy	uart2_rxd	A17	
P9_24	97	0x684164	15	UART1_TXD	gpio015	pr1_uart0_rxd		pr1_uart0_rxd	I2C1_SCL	mm2_rxdy	uart1_rxd	V14	
P9_25	107	0x68016c	117	GPIO3_21	gpio021	pr1_uart0_rxd		pr1_uart0_rxd	mm2_swp	mm2_swp	uart1_rxd	D15	
P9_26	96	0x680160	14	UART1_RXD	gpio014	pr1_uart0_rxd		pr1_uart0_rxd	mm2_swp	mm2_swp	uart1_rxd	D16	
P9_27	105	0x684164	115	GPIO3_19	gpio019	pr1_uart0_rxd		pr1_uart0_rxd	mm2_swp	mm2_swp	uart1_rxd	D16	
P9_28	103	0x69016c	113	SPH1_CS0	gpio017	pr1_uart0_rxd		pr1_uart0_rxd	mm2_swp	mm2_swp	uart1_rxd	C13	Allocated mcsap0_pins
P9_29	101	0x694164	111	SPH1_D0	gpio015	pr1_uart0_rxd		pr1_uart0_rxd	mm2_swp	mm2_swp	uart1_rxd	C12	Allocated mcsap0_pins
P9_30	102	0x698168	112	SPH1_D1	gpio016	pr1_uart0_rxd		pr1_uart0_rxd	mm2_swp	mm2_swp	uart1_rxd	B13	Allocated mcsap0_pins
P9_31	100	0x690160	110	SPH1_SCLK	gpio014	pr1_uart0_rxd		pr1_uart0_rxd	mm2_swp	mm2_swp	uart1_rxd	D12	Allocated mcsap0_pins
P9_32				VADC								A13	Allocated mcsap0_pins
P9_33				AIN4								CR	1.8V input
P9_34				AGND									Ground for ADC
P9_35				AIN6								AE	1.8V input
P9_36				AIN5								DE	1.8V input
P9_37				AIN2								BT	1.8V input
P9_38				AIN3								A7	1.8V input
P9_39				AIN0								BE	1.8V input
P9_40				AIN1								C7	1.8V input
P9_41A	109	0x6bd164	20	CLKOUT2	gpio020	EMU3_mux0		clkout2	clkout2	clkout2	xdma_event_irq1	D14	Both to P21 of P11
P9_41B		0x680168	116	GPIO3_20	gpio020	pr1_uart0_rxd		clkout2	clkout2	clkout2	xdma_event_irq1	D13	Both to P21 of P11
P9_42A	89	0x684164	7	GPIO0_7	gpio07	xdma_event_irq2		pr1_uart0_rxd	pr1_uart0_rxd	pr1_uart0_rxd	eCAP0_in_PPM0_out	C18	Both to P22 of P11
P9_42B		0x690160	114	GPIO3_18	gpio018	pr1_uart0_rxd		pr1_uart0_rxd	pr1_uart0_rxd	pr1_uart0_rxd	eCAP0_in_PPM0_out	B12	Allocated mcsap0_pins
P9_43				GND									Ground
P9_44				GND									Ground
P9_45				GND									Ground
P9_46	cut		(Mode 7)	GND								CPU	Notes

Obrazek C.6. Přehled adres a módů na konektoru P9

Příloha D

Ukázka vytvořeného Device Tree

```
|/dts-v1/;
|/plugin/;
|
| {
|     compatible = "ti,beaglebone", "ti,beaglebone-black";    /* identification */
|     part-number = "BP-PPS";
|     version = "00A0";
|     exclusive-use = "P9.28", /* pps signal */
|                    "gpio3_17";
|     fragment@0 {
|         target = &am33xx_pinmux;
|         __overlay__ {
|             gps_pps_pins: pinmux_gps_pps_pins {
|                 pinctrl-single,pins = <
|                     0x19c 0x27 /* P9 28 GPIO3_17: MODE7 | INPUT */
|                 >;
|             };
|         };
|     };
|     fragment@1 {
|         target = &ocp;
|         __overlay__ {
|             pps {
|                 compatible = "pps-gpio";
|                 status = "okay";
|                 pinctrl-names = "default";
|                 pinctrl-0 = <&gps_pps_pins>;
|                 gpios = <&gpio3 17 0>;
|             };
|         };
|     };
| };
```

Obrázek D.7. Ukázka vytvořeného Device Tree

Příloha E

Ukázka vytvořeného Device Tree

```
82
83 int main(int argc, char** argv)
84 {
85     gpio_init();
86     gpio_set(gpspwr_fd, 1);
87     char ces1;
88     char ces2;
89     int lm_hour;
90     int lm_min;
91     int lm_hod;
92     int p_pulzu;
93     int tm_hod = 0;
94     int rcas;
95     int lcas;
96     int cas_dif;
97     int tm_gmtoff;
98     printf("pocet hodin na lince:");
99     scanf("%d", &lm_hour);
100    printf("pocet minut na lince:");
101    scanf("%d", &lm_min);
102    while (1) {
103        time_t t = time(NULL);
104        struct tm tm = *localtime(&t);
105        if (lm_hour == tm.tm_hour && lm_min == tm.tm_min)
106        {
107            if (tm.tm_sec == 59)
108            {
109                if (tm.tm_min % 2 == 0)
110                {
111                    gpio_set(MP3S_fd, 1);
112                    gpio_set(LED1_fd, 1);
113                    usleep(1000000);
114                    gpio_set(MP3S_fd, 0);
115                    gpio_set(LED1_fd, 0);
116                }
117                else
118                {
119                    gpio_set(MP3L_fd, 1);
120                    gpio_set(LED2_fd, 1);
121                    usleep(1000000);
122                    gpio_set(MP3L_fd, 0);
123                    gpio_set(LED2_fd, 0);
124                }
125                lm_min++;
126                if (lm_min == 60)
127                {
128                    lm_hour++;
129                    lm_min = 0;
130                    if (lm_hour == 24)
131                    {
132                        lm_hour = 0;
133                    }
134                }
135            }
136        }
137        else
138        {
139            if (tm.tm_hour > 12)
140            {
141                tm_hod = tm.tm_hour - 12;
142            }
143            else
```

Obrázek E.8. Zdrojový kód 1/2

```
143     else
144     {
145         tm_hod = tm.tm_hour;
146     }
147     if (lm_hour > 12)
148     {
149         lm_hod = lm_hour - 12;
150     }
151     else
152     {
153         lm_hod = lm_hour;
154     }
155     rcas = ((tm_hod * 60) + tm.tm_min);
156     lcas = ((lm_hod * 60) + lm_min);
157     cas_dif = rcas - lcas;
158     if (cas_dif > 0)
159     {
160         p_pulzu = cas_dif;
161     }
162     else
163     {
164         p_pulzu = 720 + cas_dif;
165     }
166     if (p_pulzu > 696)
167     {
168     }
169     else
170     {
171         for (size_t i = 0; i < p_pulzu; i++)
172         {
173             if (lm_min % 2 == 0)
174             {
175                 gpio_set(MP3S_fd, 1);
176                 gpio_set(LED1_fd, 1);
177                 usleep(1000000);
178                 gpio_set(MP3S_fd, 0);
179                 gpio_set(LED1_fd, 0);
180             }
181             else
182             {
183                 gpio_set(MP3L_fd, 1);
184                 gpio_set(LED2_fd, 1);
185                 usleep(1000000);
186                 gpio_set(MP3L_fd, 0);
187                 gpio_set(LED2_fd, 0);
188             }
189             lm_min++;
190             if (lm_min == 60)
191             {
192                 lm_hour++;
193                 lm_min = 0;
194                 if (lm_hour == 24)
195                 {
196                     lm_hour = 0;
197                 }
198             }
199             usleep(1000000);
200         }
201     }
202 }
203 }
204 }
```

Obrázek E.9. Zdrojový kód 2/2

Literatura

- [1] Elektročas s.r.o. *Interiérové hodiny řady MK*.
http://www.elektrocas.cz/art_file/MK.pdf.
- [2] Elektročas s.r.o. *Exteriérové digitální hodiny řady ED*.
http://www.elektrocas.cz/art_file/ED-rada.pdf.
- [3] Wikipedia.org. *DCF77*.
<https://en.wikipedia.org/wiki/DCF77>.
- [4] Martin Poupa. *Vysílač časového signálu a normál. frekvence DCF77*. 2012.
<http://home.zcu.cz/~poupa/dcf77.html>.
- [5] Wikipedia.org. *Mapa pokrytí signálem DCF77*.
<https://commons.wikimedia.org/wiki/File:DCF77-Range.png>.
- [6] Hopf GmbH. *DCF77 Time Code Format*.
<http://www.hopf.com/pics/dcf77-time-code-format.jpg>.
- [7] Wikipedia.org. *Global Positioning System*.
https://en.wikipedia.org/wiki/Global_Positioning_System.
- [8] COPS. *GPS satellites flying around Earth*.
https://cops.usdoj.gov/html/dispatch/05-2015/enhancing_cp_through_gps.asp.
- [9] Joe Peter. *NMEA data*.
<http://www.gpsinformation.org/dale/nmea.htmstream>.
- [10] Cooking Hacks. *NMEA sentences coming from the GPS module*.
<https://www.cooking-hacks.com/documentation/tutorials/gps-module-arduino-raspberry-pi-tutorial>.
- [11] Wikipedia.org. *Network Time Protocol*.
https://en.wikipedia.org/wiki/Network_Time_Protocol.
- [12] *NTP - How does it work?*
<http://www.ntp.org/ntpfaq/NTP-s-algo.htmQ-ALGO-BASIC-SYNC>.
- [13] Elektročas s.r.o. *Hlavní hodiny a hodinové ústředny*.
<http://www.elektrocas.cz/KATALOG/Hlavni-hodiny-a-hodinove-ustredny>.
- [14] BRNOTRON. *Hlavní hodiny*.
<http://www.brnotron.cz/hlavni.html>.
- [15] Vistech. *Základní druhy systémů jednotného času*.
<http://www.vistech.cz/produkty/synchronizace-casovych-systemu.htm>.
- [16] Toshiba. *TB6568KQ, datasheet*.
http://www.mouser.com/ds/2/408/TB6568KQ_datasheet_en_20170414-771365.pdf.
- [17] JKRINDER.WORDPRESS.COM. *BeagleBone Black hardware details*.
<http://beagleboard.org/support/bone101>.
- [18] Derek MOLLOY. *Exploring BeagleBone: tools and techniques for building with embedded Linux*. John Wiley Sons, 2015. ISBN 1118935128.

-
- [19] Gerald Coley. *BeagleBone Black System Reference Manual*. 2014.
<https://media.digikey.com/pdf/Data%20Sheets/Circuitco%20Elect/BB-BBLK-000%20Manual.pdf>.
- [20] Michael Leonard. *How to Choose the Right Platform: Raspberry Pi or BeagleBone Black?*
<https://makezine.com/2014/02/25/how-to-choose-the-right-platform-raspberry-pi-or-beaglebone-black/>.
- [21] *Debian GNU/Linux — instalační příručka*.
<https://www.debian.org/releases/wheezy/armel/index.html.cs>.
- [22] elinux.org. *Beagleboard:Android*.
<https://elinux.org/Beagleboard:Android>.
- [23] Jason Krinder. *Angstrom Distribution*.
<https://beagleboard.org/p/koenkooi-myopenid-com/angstrom-distribution-e3c0bb>.
- [24] Machinekit. *Machinekit Documentation*.
<http://www.machinekit.io/docs/>.
- [25] JKRINDER.WORDPRESS.COM. *Getting Started*.
<http://beagleboard.org/getting-started>.