

GYMNÁZIUM JANA KEPLERA

ZÁVĚREČNÁ PRÁCE  
INFORMATIKA

---

**DCF77 NTP server**  
*server poskytující přesný čas*

---

*Autor:*  
MACHÁČEK TOMÁŠ

*Vedoucí práce:*  
Bc. EMIL MILER

March 24, 2022

Prohlašuji, že jsem tuto práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, v platném znění.

*Dne:*

---

*Podpis:*

---

# Obsah

<b>1</b>	<b>Abstrakt</b>	<b>3</b>
<b>2</b>	<b>Návrh antény</b>	<b>4</b>
2.1	Teorie . . . . .	4
2.1.1	Smyčková anténa . . . . .	4
2.2	Konstrukce . . . . .	5
2.2.1	Program pro výpočet antény . . . . .	5
<b>3</b>	<b>Řídicí jednotka antény</b>	<b>6</b>
3.1	Teorie . . . . .	6
3.1.1	Amplitudová modulace . . . . .	6
3.1.2	DCF77 . . . . .	6
3.1.3	RS-485 komunikace . . . . .	7
3.2	Zpracování anténní jednotky . . . . .	7
3.2.1	Demodulace DCF77 signálu . . . . .	7
3.2.2	Převod na RS-485 sběrnici . . . . .	8
3.2.3	Systém pro vyhledávání signálu . . . . .	8
<b>4</b>	<b>Hlavní jednotka</b>	<b>10</b>
4.1	Teorie . . . . .	10
4.1.1	Network time protocol . . . . .	10
4.2	Hardware . . . . .	10
4.2.1	RPi HAT . . . . .	11
4.2.2	Zprovoznění DCF77 jako stratum 1 NTP serveru . . . . .	12
4.2.3	Ethernetové připojení a USB-C napájení . . . . .	13
<b>5</b>	<b>Nasazení a obsluha</b>	<b>14</b>
5.1	Nasazení zařízení na lokální síť . . . . .	14
5.2	Status na displeji . . . . .	15

# Abstrakt

asamblace Závěrečná práce pojednává o NTP serveru, který je synchronizovaný s přijímaným časem. Čas je poskytován německým vysílačem DCF77 v Mainflingenu.

První část se zabývá návrhem antény, která je schopna přijímat signál o kmitočtu 77.5 kHz.

V druhé části práce se rozebírá demodulaci signálu a převod na průmyslovou RS-485 sběrnici. Vše je součástí tzv. anténní jednotky, obsahuje anténu, demodulační obvod a systém kontrolující kvalitu signálu, což je jedna ze dvou hlavních komponent projektu.

Třetí část se týká hlavní jednotky, ta je zodpovědná za zpracování DCF77 signálu a hostování NTP serveru. Jádrem je Raspberry Zero 2, na které je navržený na zakázku vyrobený HAT. Také rozebírá softwarevou stránku hlavní jednotky, instalaci NTP. Též komplikace vzniklé při sestavování.

Poslední pasáž se zabývá nasazením zařízení do provozu a posléze obsluhou zařízení.

# Návrh antény

## 2.1 Teorie

Anténa je zařízení, které slouží k vysílání či přijímání elektromagnetického záření. Každý vodič, jímž prochází střídavý elektrický proud, je vlastně anténa. Geometrickým tvarováním vodiče a přidáváním jistých materiálů (ferity) se upravuje účinnost antény, dle vhodných parametrů.

Každá anténa má své základní parametry, mezi hlavní patří zisk, rezonanční frekvence, šířka přijímaného pásma a vyzařovací úhel. Podle použití antény se tyto parametry mění. Avšak hlavním parametrem, který nejvíce ovlivňuje vlastnosti, je fyzický tvar antény.

Proto dělíme antény na dipólové, monopólové, složené (Yagi-Uda anténa), smyčkové, kónické a clonové. Každá kategorie představuje jiné spektrum použití. Dipólové se používají pro vysílání TV signálu. Monopólové se používají při příjmu rádiových stanic, vlnové délky řádu desítek centimetrů. Složené jsou vysoce účinné pro širokopásmový příjem, TV i rádia. Smyčkovým anténám se budeme ještě věnovat. Kónické jsou vysoce účinné při vysokých frekvencích. Clonové jsou ideální pro dlouhý dosah.

### 2.1.1 Smyčková anténa

Smyčkové antény jsou velice jednoduché a všestranné. Jejich tvar je rozmanitý, například zaujímají tvar čtvercovitý, trojúhelníkový, kruhový a elipsovitý. Kvůli jednoduchému návrhu a analýze jsou nejvíce rozšířené mezi radioamatéry.

Klasifikují se dvou hlavními skupinami, elektricky malé a velké. Elektricky malé antény jsou ty, jejichž celková délka (počet závitů znásobený obvodem antény) je menší než desetina vlnové délky. Zatímco velké antény svojí délkou odpovídají vlnové délce rezonanční frekvence. Nejvíce se smyčkové antény používají.

Elektricky malé smyčkové antény mají obvykle menší radiační odpor než ztrátový odpor, tedy jsou špatné zářiče, používají se výhradně na příjem signálu. Nejvíce se používají při radiovému příjmu.

Radiační odpor lze zvýšit, tím pádem zvýšit zisk antény, elektrickou délkou antény nebo použít jádro z materiálu o vysoké permeabilitě, nejčastěji ferit. Anténa, která je tvořená cívkou navinutou na feritovém jádře, se nazývá feritová smyčková anténa.

## 2.2 Konstrukce

Pro tento projekt byla zvolena feritová smyčková anténa. Hlavními důvody, proč byla zvolena tento typ antény je kompaktnost antény, velikost řádu centimetrů, a zisk při dané rezonanční frekvenci. Praktická konstrukce je též jednoduchá. Jedná se vlastně o cívku, která je navinutá na feritové tyčince. S tím úzce souvisí požadovaná cena antény, která je nízká. Nevýhodou bylo shánění feritové tyčinky.

Proto byla nejdříve pořízena tyčinka podle ní byla vypočítána cívka. Pro výrobu cívky byl použit enamelový drát o průměru 0.2mm. Na polohu cívky, počet závitů a další parametry byl navržen program, o něm později. Cívka se nesmí na tyčce pohybovat, proto je zalitá epoxidem.

Správná funkčnost antény byla zjištěna pomocí následující sestavy. Na generátor funkcí byl připojen volný vodič. Takže jsme vytvořili jednoduchou anténu. Anténa byla připojena na předzesilovač a poté na osciloskop. Měněním frekvence na generátoru jsme sledovali úroveň napětí na výstupu zesilovače. Největší úroveň jsme dosáhli na frekvenci 77.564 kHz, což je v pořádku, neboť v demodulačním obvodu je použit vysoce účinný krystalový filtr.

### 2.2.1 Program pro výpočet antény

Program na výpočet antény se skládá z dvou hlavních částí, kalkulace modelu antény a poté zpracování dat do grafů.

Základem je výpočet radičního -  $R_r$ , ohmického -  $R_L$  a ztrátového odporu -  $R_f$ .

$$\begin{aligned} R_r &= 20\pi^2 \left(\frac{o}{\lambda}\right)^4 \mu_{eff}^2 N^2 \\ R_L &= \frac{l}{\sigma \pi d \delta} \\ R_f &= 2\pi f \mu_{eff} \frac{\mu_i}{\mu} \mu_0 N^2 \frac{S}{L_r} \end{aligned} \quad (2.1)$$

Kde  $o$  je účinný obvod antény,  $N$  je počet závitů cívky,  $d$  je průměr vodiče,  $f$  je rezonanční frekvence,  $S$  je účinný povrch antény a  $L_r$  je délka antény.

Pro výpočet pozice cívky na feritové tyčce byly použity Nagaokovy rovnice, které nejbližší odpovídají reálnému modelu antény.

Dále program spočítá rezonanční kondenzátor k dané anténě, jakost antény. Zde je nutno podotknout, že jakost úzce souvisí s ziskem a šířkou přijímaného pásma. Čím užší přijímané pásmo, tím vyšší zisk, avšak anténa je více náchylná na změny tlaku, teplot atd. Právě proto generuje program grafy, které vystihují závislosti díky nimž je snazší nalézt optimální parametry pro danou anténu.

Na konec program ještě vygeneruje tabulku, v níž jsou zapsány jednotlivé konfigurace antén.

# Řidicí jednotka antény

## 3.1 Teorie

### 3.1.1 Amplitudová modulace

Amplitudová modulace, zkráceně AM, je nejjednodušší a nejstarší typ modulace. V závislosti na změně modulačního signálu se mění amplituda nosného signálu. Nosná vlna má řádově vyšší frekvenci než modulovaný signál. Fáze a frekvence se u této modulace nemění.

#### Matematický popis

Pokud nosnou vlnu vyjádříme jako  $n(t) = N \sin(\Omega t)$ , kde  $N$  je amplituda a  $\Omega$  je úhlová frekvence, a jednoduchý harmonický signál  $m(t) = M \sin(\omega t + \phi)$ , kde  $\phi$  je fázový posun vůči nosné  $n(t)$ , pak AM vznikne složením přidáním  $m(t)$  k amplitudě nosné  $N$ . Dostáváme tedy

$$y(t) = (N + M \sin(\omega t + \phi)) \sin(\Omega t) \quad (3.1)$$

### 3.1.2 DCF77

Označení DCF77 se užívá pro německý dlouhovlnný rádiový vysílač a také jedinou stanici, jíž poskytuje. Nosná frekvence DCF77 je 77500 Hz. Antény stanice se nacházejí v Mainflingu. Pravidelné a nepřetržité vysílání začalo v roce 1970.

#### Vysílač

Vysílač má výkon 50 kW, avšak skutečný vyzářený výkon je přibližně 25 kW. K vysílání je použita všesměrová anténa s kapacitním nástavcem, hlavní anténa je vysoká 150 m, přičemž při údržbách se používá 200 m vysoká záložní. Dosah vysílače je 1500 až 2000 km. V České republice fungoval do roku 1995 velice podobný vysílač OMA.

#### Kódování DCF77

Kódování časové informace je prováděno pulzně šířkovou modulací (základní AM), poklesem amplitudy nosné na 25 % na začátku každé sekundy. Pakliže je puls dlouhý 100ms jedná se logickou 0, 200ms dlouhý je logická 1. Aktualizace přenášené informace je prováděna každou minutu.

Přenášejí se tyto informace:

- čas platný pro následující minutu
- datum platné pro následující minutu
- číslo dne v týdnu
- hlášení změny časové zóny 1 hodinu předem (informace o přechodu ze standardního na letní čas nebo opačně)
- časová zóna (je-li aktivní CET nebo CEST)
- hlášení přestupné sekundy 1 hodinu předem
- provoz na normální a rezervní anténu
- zabezpečení přenosu – několik paritních bitů
- od roku 2007 i předpověď počasí

### 3.1.3 RS-485 komunikace

Jedná se o typ komunikace, která se především používá v průmyslových prostředích. Standard 485 je navržen tak, aby umožňoval vytvoření dvou vodičového poloduplexního vícebodového sériového spojení. Vlastně se jedná o značně napětově posílenou RS-232, což je totéž co RS-485, avšak dosah sběrnice je až 1200 m, oproti 20 m.

Komunikační linka je tvořena dvěma vodiči (označené A a B), na obou koncích jsou kvůli zvýšení odolnosti proti rušení  $120\Omega$  rezistory. Maximální přenosová rychlost je 10 Mib/s.

Přenos dat je uskutečněn pomocí 7 nebo 8bitových rámců se startbitem a poté jedním či více stopbity. Vysílač by měl na výstupu při logické 1 (klidový stav linky) generovat na vodiči A napětí  $-2\text{ V}$ , na vodiči B  $+2\text{ V}$ , při logické 0 by měl na vodiči A generovat  $+2\text{ V}$ , na vodiči B  $-2\text{ V}$ .

## 3.2 Zpracování anténní jednotky

Mezi hlavní požadavky patřila i kompaktnost, to znamená, že byl vytvořený, na zakázku, plošný spoj, na němž jsou uloženy všechny komponenty. Dělí se na tři základní sektory, první je zodpovědný za demodulaci signálu, druhý převádí demodulovaný signál na RS-485 sběrnici a poslední vyhodnocuje jakost signálu. Každému sektoru je přiřazen jeden integrovaný obvod.

### 3.2.1 Demodulace DCF77 signálu

Pro tento účel byl zvolen U4224B integrovaný obvod. Jedná se o přijímací obvod, který disponuje demodulací od 40 do 80 kHz, podle zvoleného krystalu. Dále má velmi vysokou citlivost, takže i v oblatech s vysokým rušením či slabým signálem, je možné anténu dobře naladit.



Mohl být použito i zapojení typu superhat. Oscilátor naladěný na frekvenci 77 kHz by byl přiveden na vstup vyváženého směšovače, kde společně s DCF77 signálem (77.5 kHz), by vytvořily signál o rozdílovém kmitočtu 500 Hz. Tento signál by byl přiveden na vstup operačního zesilovače, který by fungoval jako aktivní pásmová propust s šířkou pásma 50 Hz. Jelikož by obvod nebyl stabilní bylo by také nutné vybavit obvod AVC (aktivním vyvážením citlivosti).

Obvod výše popsany by bylo možné sestavit pomocí diskrétních součástek, ale byl by komplikovaný. Situaci navíc komplikuje fakt, že dnes hodně spínaných zdrojů generují kmitočty, které vytvářejí harmonické násobky, a ty ruší příjem. Proto šířka pásma 50 Hz je prostě moc široká. Zúžení pásma vedlo k použití tzv. krystalového filtru. Implementace krystalového filtru vedla na integrovaný obvod.

Na výstupu integrovaného je tedy demodulovaný DCF77 signál. Načež kvůli zvýšení stability a ochrany dekodéru byl k obvodu zahrnut tvarovač signálu. Tedy obvod, který se skládá z tranzistoru, který chrání dekodér a Schmittův klopný obvod na definování napěťové úrovně. Tento signál je vedený na vstup převodníku RS-485.

### 3.2.2 Převod na RS-485 sběrnici

Pro dobrý příjem signálu je nutné, aby byla anténa na vyvýšeném místě, bez rušení. To znamená, že napájecí vodiče, ale i datové linky, musejí být alespoň 10m dlouhé. Napájení není problém, ale signál vedený klasickou TTL komunikací by byl ohrožen, norma TTL deklaruje dosah 2-3 metry. Tím pádem bylo nutné nalézt jinou normu.

Potom byla zvolena norma RS-232, ta má dosah 20m. Ale potom se přešlo na RS-485, ta má jednak o 1 km větší dosah a používá stejný typ kódování jako 232. A to z důvodu jednak stejné pořizovací ceny a vyšší napěťová úroveň zajišťuje značně vyšší odolnost vůči rušení.

Pro transkódování signálu byl použit integrovaný obvod SN75176, nepatří mezi rychlé převodníky, avšak na naši aplikaci je postačující. Komunikace probíhá jednostranně halfduplexem.

Abychom ochránili jak anténní jednotku, jsou za obvodem zapojeny trisily a teplem vratné pojistky proti přepětí a nadproudu, který se může indukovat ve vodičích. Nakonec je signál zaveden na vstup Ethernet konektoru. Ten byl použit proto, že má pojistku a oproti ostatním průmyslovým konektorům je velice levný. A navíc UTP kabel využívá kroucenou linku, která je záhodná u 485 komunikace.

### 3.2.3 Systém pro vyhledávání signálu

Abychom nemuseli při ladění antény čekat dvě minuty na načtení celého bufferu a potom zjišťovali, zda je správný. Nutno podotknout, že anténa je směrová, takže záleží na směru normálového vektoru vůči zdroji, což je anténa DCF77. Což ještě ztěžuje situaci ladění. Další, rychlejší, způsob ladění vyžaduje osciloskop. Proto bylo nutné vybavit obvod zařízením, které by v reálném čase sledovalo přijímaný signál a vyhodnocovalo jeho kvalitu.

Srdcem je mikrokontrolér PIC, na který je nahraný program, který analyzuje signál z přijímače. Dále jsou na PIC připojeny dvě LED, které fungují jako signalizace. Jedna je zelená, značící, že je obvod napájen a druhá, oranžová, signalizuje kvalitu signálu. A to následujícím způsobem:

*Pokud dioda nesvítí, potom je signál zcela vadný. Jakmile se dioda rozbliká, znamená to, že byl přijat validní bit, pakliže se dioda rozbliká rychleji (s vyšší frekvencí), byl přijat další validní bit. Tento postup pokračuje, až dioda svítí, neboli anténa je naladěná. S klesající frekvencí se naopak anténa rozladuje. Takže svítící dioda je zárukou dobrého signálu.*

Program funguje tak, že kontroluje správnou délku signálu, frekvenci, intervaly mezi jednotlivými pulsy, inverzní délku pulzů a minutový puls (prázdný). Pakliže všechny parametry jsou splněny je bit zaznamenán jako korektní. Tuto proceduru provádí s každým bitem a to v reálném čase, takže uživatel má aktuální přehled o kvalitě přijímaného signálu a může nalézt místo s nejlepším příjmem.

Deska je osazena pinovou lištou, která slouží k flashování programu, takže je velice snadné aktualizovat program v PICu.

# Hlavní jednotka

## 4.1 Teorie

### 4.1.1 Network time protocol

NTP je protokol pro synchronizaci vnitřních hodin počítačů po paketové síti s proměnným zpožděním. Tento protokol zajišťuje, aby všechny počítače v síti měly stejný a přesný čas. Byl obzvláště navržen tak, aby odolával následku proměnlivého zpoždění v doručování paketů.

Počítač, který chce synchronizovat své hodiny, pošle pár dotazů několika NTP serverům a ty mu v odpovědi pošlou svůj, přesný čas. Klient z odpovědí nejprve vyloučí servery se zřejmě nesmyslným časem (s odchylkou 1000 sekund a více). Poté ponechá skupinu serverů s největším společným průnikem.

NTP je jeden z nejstarších dosud používaných IP protokolů. NTP původně navrhl Dave Mills z univerzity v Delaware a stále jej, spolu se skupinou dobrovolníků, udržuje. Současná verze je NTP verze 4, kterou popisuje RFC 5905.

NTP démon je uživatelský proces, který na stroji běží trvale. Většina protokolu a inteligence je implementována v tomto procesu. Pro dosažení nejlepšího výkonu je důležité, aby jádro operačního systému umělo řídit čas fázovým závěsem, místo aby přesný čas do systémových hodin dosazoval NTP démon přímo. Všechny dnešní verze Linuxu fázový závěs implementují.

NTP používá hierarchický systém „strata hodin“, kde systémy se stratem 1 jsou synchronizovány s přesnými externími hodinami jako třeba GPS nebo jiné hodiny řízené rádiovým signálem (v Česku obvykle DCF77). NTP systémy strata 2 odvozují svůj čas od jednoho nebo více systémů se stratem 1 atd. To zabraňuje vzniku cyklu v grafu synchronizujících se strojů. Stratum systému leží v rozsahu 1 až 14; stratum 0 mají samotné referenční hodiny připojené k nejpřesnějšímu serveru; stratum 15 má počítač, který se v důsledku výpadku sítě nemůže synchronizovat s zdrojem času, nebo se synchronizuje po výpadku spojení.

## 4.2 Hardware

V této části se budeme zabývat, hardwarovou stránkou hlavní jednotky. Bylo nutné vyřešit dvě záležitosti, za prvé jakou platformu zvolit pro běh NTP serveru a za druhé jak konvertovat zpět RS-485 signál. Jelikož provoz NTP serveru není hardwarově náročný, bylo zvoleno Raspberry ZERO 2. To přineslo řadu výhod, okamžitě se vyřešil druhý problém, všechny přídavné obvody budou součástí tzv. HATu. A navíc je celá sestava velice kompaktní.

### 4.2.1 RPi HAT

HAT je označení pro PCB, které je namontováno na, v našem případě jednodeskový počítač. Přičemž rozšiřuje funkčnost desky, přidává vlastnosti. Raspberry disponují 2x20 pinovou lištou, GPIO, která je vhodná pro komunikaci s HATem. Rozměry HATu jsou shodné s Raspberry, je to z toho důvodu, aby byl HAT pomocí sloupků spojen s raspberry, a tím bylo zajištěno pevné spojení desek.

Hlavní funkce HATu je jasná, má za úkol převádět RS-485 komunikaci zpět na TTL logickou úroveň. Ale potom byl na desku ještě přidán ethernetový převodník. Iniciálně se totiž plánovalo použít raspberry zero, to se neuskutečnilo, více bude řečeno později.

#### Komunikace s anténou

Je použito naprosto stejné zapojení jako na řídicí jednotce antény, to znamená i stejný integrovaný obvod. Akorát je zapojený v režimu receiver. Výstup je zaveden na UART na headeru raspberry.

#### Zprovoznění UART komunikace

Při zprovoznění serveru vyvstal problém. Na samé piny (10 a 12) je připojena sběrnice bluetooth. A přitom raspberry má několik UART zařízení, konkrétně AMA0, USB0 a S0. Pro naše účely je vhodné pouze AMA0, jelikož jako jediný disponuje nezávislým UART převodníkem, mimo CPU. Ostatní jsou závislé na aktuální taktovací frekvenci CPU, která se mění, a to je nežádoucí u přesného vyčítání dat.

Takže jako první se deaktivoval kompletně bluetooth, přes `raspi-config`. Ve stejném prostředí byl povolen AMA0 serial port.

```
/boot/config.txt

[all]

enable_uart=1
dtoverlay=disable-bt
```

Poté byla ověřena komunikace pomocí `# screen /dev/ttyAMA0 9600`. Na konzoli by se měli zobrazovat znaky (jedno jaké) s vteřinovou periodou.

### 4.2.2 Zprovoznění DCF77 jako stratum 1 NTP serveru

Potom, co jsme zajistili funkční vyčítání dat z antény správným UART rozhraním, se můžeme přesunout na samotnou instalaci NTP serveru.

#### Instalace s podporou RAW DCF

Prvním krokem je nainstaloování NTP, avšak pro naše účely je nutné podpora DCF77. Instalaci provedeme dle následujících kroků.

```
$ wget http://www.eecis.udel.edu/~ntp/ntp_spool/
ntp4/ntp-4.2/ntp-4.2.8p15.tar.gz
$ tar xzf ntp-4.2.8p15.tar.gz
$ cd ntp-4.2.8p15/
$ ./configure --enable-RAWDCF --prefix=/usr
$ make
$ sudo systemctl stop ntp.service
$ sudo make install:
$ echo "ntp hold" | sudo dpkg --set-selections
```

Poslední příkaz slouží k zakázání aktualizací NTP. Verzi NTP ověříme přes `ntpq --version`.

#### Nastavení strata 1

Ještě před tím, než nastavíme referenční hodiny pro NTP, přidáme uživatele `user` do skupiny `tty`.

```
$ sudo adduser user tty
$ cat /etc/group | grep tty
```

Nyní zeditujeme wrapper kód NTP, tak, aby server užíval DCF jako zdroj stratum 1. Referenční hodiny `/dev/refclock-0` jsou systémovým linkem na `/dev/ttyAMA0`.

```
#for using the DCF77 receiver add this
if [ ! -L /dev/refclock-0 ]; then
    ln -s /dev/ttyAMA0 /dev/refclock-0
fi

#commented out this
#if [ -e /run/ntp.conf.dhcp ]; then
    #NTPD_OPTS="$NTPD_OPTS -c /run/ntp.conf.dhcp"
#fi
```

Nastartujeme daemon `sudo systemctl start ntp.service`. Můžeme zkontolovat přítomnost symbolického linku.

```
$ ls -ahl /dev/ref*  
/dev/refclock-0 -> /dev/ttyAMA0
```

Dále upravíme soubor `/etc/ntp.conf`.

```
ntp.conf
```

Konečně restartujeme NTP daemon a zkontrolujeme aktivitu NTP serveru, a to přes `ntpq -p`. Správný příjem lze jednoduše ověřit tak, že hodnota *reach* je 377.

### Oprava časové diletace = fudgetime

Díky nedokonalosti přenosu a časovým prodlevám vzniklé konverzí, je záhodné nastavit kompenzační časovou konstantu, která bude vyrovnávat časový schodek. Celkový fudgetime je dán součtem předkonfigurovaným a reálného, ten je v našem případě 592 ms. Předkonfigurovaný fudgetime zjistíme následovně (zde 292 ms).

```
$ ntpq -c cv  
...  
... fudgetime1=292.000,  
...
```

### 4.2.3 Ethernetové připojení a USB-C napájení

Tyto dva hardwarové prvky se nepodařilo zprovoznit. Ethernet konektor slouží k navázání stabilního připojení. Problém spočíval v tom, že se zapomnělo připojit jednu nožičku integrovaného obvodu ethernetového převodníku na +3.3V, jednalo se o napájení TX. Bypass se povedl, obvod fungoval bez problémů. Avšak kvůli veliké křehkosti a nestabilitě spoje byl čip odpájen a změněn návrh plošného spoje, který se nestihl znovu popat.

S napájecím konektorem byl nenapravitelný problém. Vlastně hlavní význam integrovat napájecí konektor na desku byla špatná orientace napájecího konektoru na RPi. A navíc USB-C napájení je v dnešní době nejuniverzálnější konektor. Problém spočíval v špatném footprintu. Vyřešilo se to tak, že se na napájecí linky RPi byly napájeny vodiče, které se přivedly na USB mini konektor, je robustnější než micro.

# Nasazení a obsluha

V závěrečné kapitole si podrobně popíšeme, jak zařízení nasadit. V podstatě, co s zařízením dělat, když se nám dostane do rukou.

## 5.1 Nasazení zařízení na lokální síť

### 1. Připojení

Jako první musíme zařízení, resp. hlavní jednotku, připojit k lokální síti. Provedeme to headless způsobem, takže první vyjmeme SD kartu z RPi. K SD kartě se dostaneme tak, že povolíme dva krajní M3 šrouby na zadní straně hlavní jednotky. Vysuneme opatrně uvolněný obal. Nikoliv se nesnažíme oddělit dvě části úplně, postačí jenom odklopit odšroubovanou část a uvidíme RPi navrchu. Poté vytáhneme SD kartu, možná budou překážet šrouby - odstraňte je proto zcela.

Vsuneme SD kartu do počítače a vytvoříme soubor (pokud již existuje, pak editujeme) `/etc/wpa_supplicant/wpa_supplicant.conf` a do něj zapíšeme.

```
network={
    ssid="SSID"
    psk="password"
    key_mgmt=WPA-PSK
}
```

Poté vrátíme SD kartu do zařízení přiklopíme obal, zašroubujeme, dbáme, abychom neskříply žádné vodiče. Zapojíme zařízení na napájení. Abychom zjistili, jakou adresu má zařízení můžeme použít například `nmap -n 192.168.0.0/24`. Zařízení bude mít pravděpodobně jako jediné otevřený jenom ssh port. Poté se jednoduše připojíme přes ssh jako uživatel `ntpccli` `ssh ntpcli@192.168.X.X`. Heslo je `ntp`. A nebo se zkusíme připojit přes `ssh ntpcli@raspberrypi.local`.

## 2. Status antény

Nyní se zaměříme na NTP server. Jako první ověříme, že je nastartovaný NTP daemon.

```
$ sudo systemctl status ntp
$ status
```

Poslední příkaz používáme k zjištění stavu, ten by měl iniciálně vypadat takto

```
ntpccli@raspberrypi:~ $ status
associd=0 status=0000 no events, clk_unspec,
device='RAW DCF77 CODE (TimeBrick)', timecode="", poll=3, noreply=0,
badformat=0, baddata=0, fudgetime1=877.000, stratum=0, refid=DCFa,
flags=0, refclock_time="<UNDEFINED>", refclock_status="",
refclock_format='RAW DCF77 Timecode',
refclock_states="*NOMINAL: 00:02:48 (100.00%); running time: 00:02:48"
```

a když je přijímán korektní signál a je správně dekodován (obvykle do tří minut po zapnutí - záleží na kvalitě signálu), potom takto:

```
ntpccli@raspberrypi:~ $ status
associd=0 status=0020 2 events, clk_unspec,
device='RAW DCF77 CODE (TimeBrick)',
timecode="---#---###---#-----M-S12---1---P12---2P---4---2---412---2---2---P",
poll=8, noreply=0, badformat=2, baddata=0, fudgetime1=877.000, stratum=0,
refid=DCFa, flags=0,
refclock_time="e5e76b81.000000000 Thu, Mar 24 2022 22:13:21.000",
refclock_status="TIME CODE; (LEAP INDICATION; CALLBIT)",
refclock_format='RAW DCF77 Timecode',
```

Pokud trvá načtení správného bufferu dlouho, tak je slabý signál, což status antény neobjeví. Buď lze počkat, NTP server nevyžaduje synchronizaci každou minutu, postačí bohatě jedna synchronizace za 12 hodin, a nebo vyhledat místo s lepším příjmem.

Pokud chceme vidět aktuální dekodovaný čas, můžeme využít funkci `t`. Ta spustí `watch` se sekundovou aktualizací. Vidíme-li **Error 666: Check NTP status.**, znamená to, že hodiny nepřijímají. Pakliže se objeví aktuální čas, hodiny přijímají.

## 3. Nastavení na lokální síť

### 5.2 Status na displeji