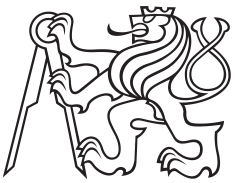
Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra měření

Detektor poruch a degradací

Ondřej Hruška

Vedoucí: doc. Ing. Radislav Šmíd, Ph.D.
Obor: Kybernetika a robotika
Studijní program: Senzory a přístrojová technika
2016

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
Katedra měření

Akademický rok **2015-16**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Ondřej Hruška**

Studijní program: **Kybernetika a robotika**
Obor: **Senzory a přístrojová technika**

Název tématu česky: **Detektor poruch a degradací**

Název tématu anglicky: **Fault and Degradation Detector**

Pokyny pro vypracování:

Navrhněte a realizujte systém pro detekci poruch a degradací zařízení pomocí vyhodnocení tvaru a dalších charakteristik odebíraného proudu, například pomocí frekvenčního spektra. Pro řešení použijte mikrokontrolér STm 32FX. Prozkoumejte možnost využití analýzy přechodových dějů při zapínání a vypínání zařízení. Systém by měl pro ovládání a vyčítání dat používat komunikační rozhraní WiFi a umožňovat několik režimů: přenos dat do nadřazeného PC pro zpracování nebo autonomní provoz s vyhodnocováním dat v mikrokontroléru a zasíláním informace o stavu zařízení.

Seznam odborné literatury:

[1] Nandi, S.; Toliyat, H.A.; Xiaodong Li: "Condition monitoring and fault diagnosis of electrical motors-a review," in Energy Conversion, IEEE Transactions on , vol.20, no.4, pp.719-729, Dec. 2005

[2] Yiu J., The Definitive Guide to the ARM Cortex-M3

[3] Ganssle, J.G.: The Art of Designing Embedded Systems, Elsevier 2000. ISBN: 978-0-7506-9869-6

Vedoucí bakalářské práce: doc. Ing. Radislav Šmíd, Ph.D.

Datum zadání bakalářské práce: 9. prosince 2015

Platnost zadání do[1]: 30. září 2017

L. S.

Doc. Ing. Jan Holub, Ph.D.
vedoucí katedry

Prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 9. 12. 2015

[1] Platnost zadání je omezena na dobu tří následujících semestrů.

Prohlášení

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, 27. května 2016

.....

Poděkování

Rád bych poděkoval doc. Ing. Radislavu Šmídovi, Ph.D za odborné vedení práce, a rovněž doc. Ing. Janu Fischerovi, CSc. za jeho cenné rady a konzultace. Dále chci poděkovat kamarádům z IRC kanálů stm32 a esp-open-rtos v síti Freenode, kteří mi několikrát pomohli s laděním firmwaru.

Abstrakt

Tato práce se zabývá implementací detektoru poruch a degradací síťového spotřebiče pomocí analýzy časového průběhu odebíraného proudu. Cílem je navrhnout a realizovat přístroj ve formě zásuvkového adaptéru, který by bylo možné použít k monitorování připojeného zařízení.

V rámci práce byl realizován prototyp přístroje s procesorem řady STM32 F3 a programovatelným WiFi modulem ESP8266. Zařízení umožňuje pohodlné ovládání a zobrazení grafů spektra a průběhu proudu pomocí webového rozhraní. Dále je podporováno pravidelné hlášení stavu na servery Xively a ThingSpeak.

Klíčová slova: detekce poruch, frekvenční analýza odběru, monitorování, WiFi, ARM, STM32, ESP8266, vestavěný systém, Internet of Things, Xively, ThingSpeak

Vedoucí: doc. Ing. Radislav Šmíd, Ph.D.

Abstract

This work focuses on the design of an AC appliance degradation detector. The goal is to implement a device in the form of a power-plug adapter that could be used to monitor and study the characteristics of the AC current.

A prototype with a STM32 F3 processor and an ESP8266 programmable WiFi module has been realised, together with a custom firmware for both processors, which allows easy access to the measurements and charts using a web browser. The device also supports regular reporting to a Xively or ThingSpeak monitoring server.

Keywords: degradation detection, frequency analysis of AC current, monitoring, WiFi, ARM, STM32, ESP8266, embedded system, Internet of Things, Xively, ThingSpeak

Title translation: Fault and Degradation Detector

Obsah

Obrázky

Kapitola 1

Úvod

V dnešní době jsme svědky rozšiřování automatizace a elektrifikace do nových odvětví, jako jsou například inteligentní budovy a moderní domácnosti, kde stále více činností zajišťují elektrická zařízení. Běžně se v nových objektech používají rekuperátory, ventilátory, klimatizace a další spotřebiče, které nahrazují například otevíratelná okna. Pro spolehlivou a bezpečnou funkci je nezbytné stav těchto zařízení monitorovat a včas je vyměnit, než dojde k úplnému selhání.

Hlavním úkolem této práce je sestrojit přístroj pro diagnostiku a monitorování (především motorem poháněných) síťových spotřebičů pomocí analýzy charakteristik jejich proudového odběru. Bude realizován prototyp tohoto přístroje ve formě zásuvkového adaptéru, který umožní zobrazení aktuálního průběhu a charakteristik, jakož i jejich dlouhodobý záznam pro sledování změn v delším časovém horizontu.

Pro pohodlné měření bez nutnosti spotřebič někam přenášet je výhodné měřicí přístroj vybavit bezdrátovým ovládáním a obsluhu provádět na dálku. K tomu bude použita síť WiFi, která je dnes velmi rozšířená a k připojení stačí obyčejný laptop nebo mobilní telefon. Za tímto účelem budou v práci prozkoumány možnosti populárního WiFi čipu ESP8266, který najde uplatnění v dalších projektech i ve výuce vestavěných systémů.

Kapitola 2

Motivace

2.1 Cíle práce

Hlavním úkolem práce je sestrojit monitorovací zařízení a ověřit na něm možnosti detekce poruch analýzou odběru síťových spotřebičů. Jedná se tedy o hardwarovou realizaci měřícího přístroje a napsání firmwaru, který bude měření provádět a umožní vyčítání výsledků. S tímto přístrojem budou následně provedena měření skutečných spotřebičů za účelem ověření schopnosti detekovat simulované poruchy (zadřená ložiska a podobně).

Konečným cílem by mělo být ucelené zařízení ve formě zásuvkového adaptéru, které bude možné ovládat prostřednictvím WiFi sítě. Kromě pohodlného ovládání přes webového rozhraní by navíc použití WiFi mělo zajistit vyšší bezpečnost, protože obsluha bude zcela bezkontaktní.

Kromě sestrojení měřícího přístroje má práce ještě vedlejší cíl, totiž prozkoumat možnosti programovatelného WiFi modulu s obvodem ESP8266, který bude v realizaci použit.

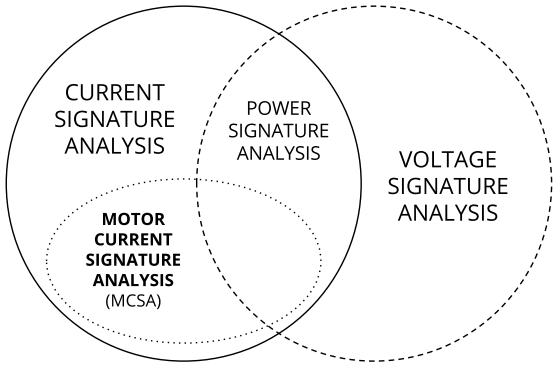
2.2 Rozdělení práce

Práci jsem rozdělil do několika částí, které dohromady povedou k realizaci kompletního měřícího přístroje:

- **Implementace vzorkování a zpracování signálu.** Je potřeba navrhnout způsob analýzy průběhu proudu a metodu detekce změn v jeho charakteristice.
- **Prozkoumání možností WiFi čipu ESP8266**, implementace vlastního firmwaru a návrh protokolu pro jeho připojení k hlavnímu procesoru.
- **Implementace uživatelského rozhraní** a autonomního monitorování. Přístroj nebude mít displej, všechno ovládání tedy musí probíhat prostřednictvím webového rozhraní (s přístupem přes WiFi), indikačních LED a tlačítek.
- **Návrh zapojení a design desky plošných spojů.** Deska by se měla vejít do zásuvkového adaptéru, musí tedy být kompaktní a je třeba dodržet bezpečné izolační vzdálenosti – přístroj bude napájen ze sítě.
- **Měření reálných spotřebičů** a analýza získaného signálu pro detekci poruch. V této části je cílem ověřit, jak dobře přístroj v praxi funguje, a získat ukázky reálných dat.

2.3 Existující implementace

Předmětem práce je monitorování síťového spotřebiče pomocí analýzy časového průběhu odebíraného proudu. Tato diagnostická metoda se obecně nazývá ESA (*Electrical Signature Analysis*), konkrétně CSA (*Current Signature Analysis*). V případě motorů, pro které je tento způsob monitorování zvláště vhodný, se jedná o metodu MCSA – *Motor Current Signature Analysis*. Na obrázku 2.1 jsou znázorněny různé varianty ESA.



Obrázek 2.1: Metody analýzy elektrických charakteristik spotřebiče spadající pod označení ESA; *Zdroj: [1]*

Analýzu odběru lze použít k detekci opotřebení mechanické části motorem poháněných zařízení, jako je například poškození ložisek [2] nebo opotřebení soukolí v převodovce [3]. Nalezl jsem též článek popisující využití MCSA k diagnostice motorem ovládaných ventilů [1], kde včasná detekce poruchy umožní výměnu bez zbytečně dlouhé odstávky.

Metoda MCSA je typicky založena na spektrální analýze proudového odběru, kde se poškození projeví jako posun či změna magnitudy jednotlivých složek. Některé závady je též možné detekovat z prostého zvýšení odběru, pokud má motor jinak konstantní zátěž (například takto lze detekovat opotřebení ložisek ventilátoru). To však není příliš spolehlivou indikací, pokud zvýšení odběru může být způsobeno i jinak.

Kapitola 3

Analýza problému a návrh řešení

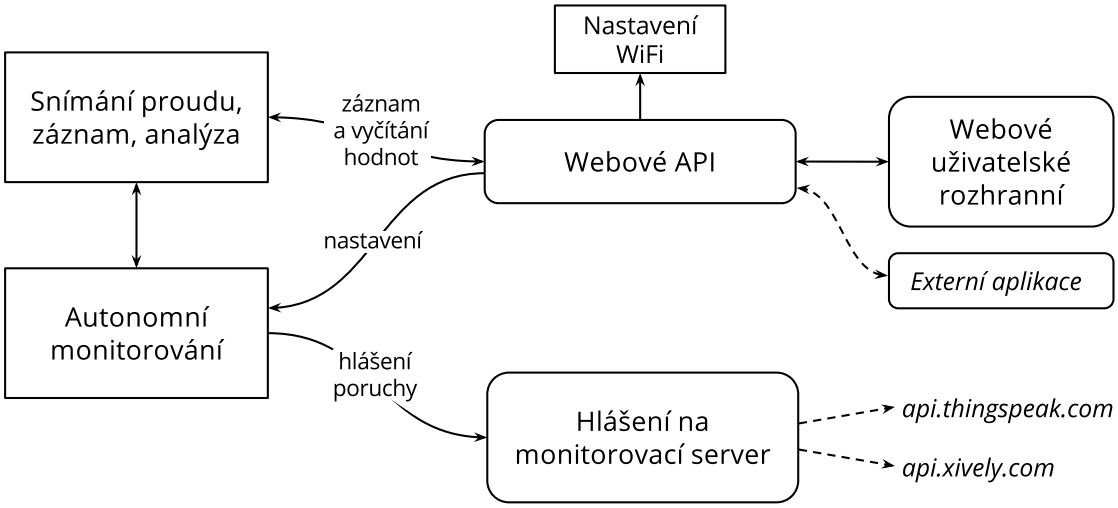
3.1 Struktura a funkce navrhovaného přístroje

3.1.1 Přehled hlavních funkcí

Jako první krok v návrhu přístroje je důležité si ujasnit, jak má fungovat a co od něj bude uživatel očekávat. Tyto požadavky z části vyplývají ze zadání práce, ale je nutné je více rozpracovat. Na obrázku 3.1 je znázorněno rámcové funkční dělení, které bylo zvoleno k implementaci.

Ovládání bude probíhat prostřednictvím webového rozhraní, a to buď ve formě interaktivních stránek v prohlížeči, nebo prostřednictvím API. Druhý způsob umožní snadnou integraci do případné externí aplikace.

Kromě přímého měření a zobrazení grafů by zařízení mělo dále mít autonomní režim, ve kterém bude aktuální naměřený stav pravidelně odesílán na monitorovací server. Tento režim umožní dlouhodobé monitorování připojeného spotřebiče.

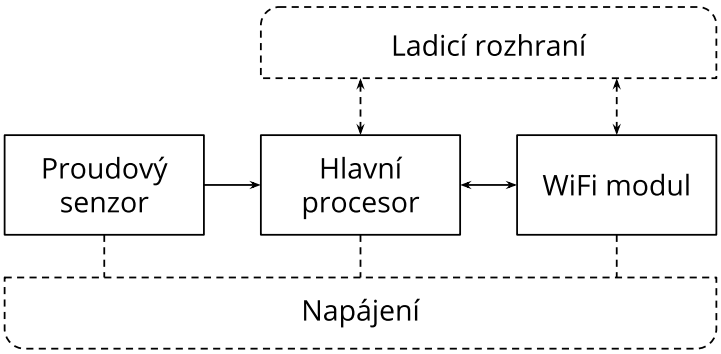


Obrázek 3.1: Blokové schéma funkcí přístroje

3.1.2 Členění hardwaru

Hardwarovou implementaci je možné rozdělit do tří hlavních bloků, jak je znázorněno na obrázku 3.2. Všechny tři bloky jsou víceméně samostatné a v principu by je bylo možné nahradit jinou implementací, například místo WiFi modulu připojit adaptér RS232 a měření provádět z PC.

Toto dělení má také význam z pohledu napájení, každý blok totiž bude nutné samostatně odrušit, aby nedocházelo ke kolísání napájecího napětí například v důsledku proudových špiček z WiFi modulu. U analogové části bude dále potřeba oddělit zem, aby proud tekoucí zemním vodičem nerušil měření.



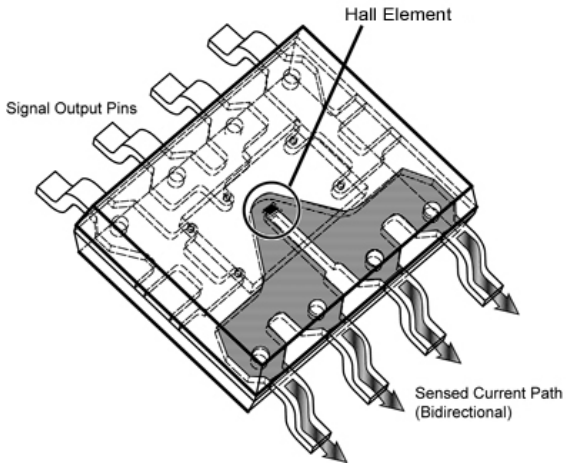
Obrázek 3.2: Blokové schéma zapojení

3.2 Způsob měření proudu

3.2.1 Volba snímacího prvku

V přístrojích tohoto typu jsou obvyklé tři základní způsoby měření proudu:

- **Modul s Hallovou sondou**, obvykle ve formě integrovaného obvodu, který se zapojí do série s měřeným spotřebičem. Tímto způsobem je možné měřit jak střídavý, tak stejnosměrný proud. Nevýhodou je vyšší cena součástky a nutnost zajistit dostatečnou izolaci. Příklad takového senzoru je na obr. 3.3.
- **Bočník vložený do měřeného obvodu** – měří se napětí na bočníku způsobené proudem, který jím protéká. Už z principu je jasné, že musíme zajistit izolaci, typicky tak, že měřicí část zapojení bude oddělena od zbytku aplikace pomocí DC/DC měniče a optočlenů. Druhou možností je použít specializovaný obvod, jako např. ADE7912, který obsahuje vše přímo v pouzdře; tím se ale podstatně zvyšuje výrobní cena přístroje. Výhodou kontaktního měření může být linearita převodního poměru a nízká cena snímacího prvku. Protože už máme kontakt se síťovou částí obvodu, je možné také měřit napětí, což umožní výpočet příkonu, účiníku a podobně.
- **Proudový transformátor**, kterým prochází síťový vodič. Měření je bezkontaktní, není zde nutné řešit izolaci; pořizovací cena není vysoká, nevýhodou jsou hlavně větší rozměry součástky.



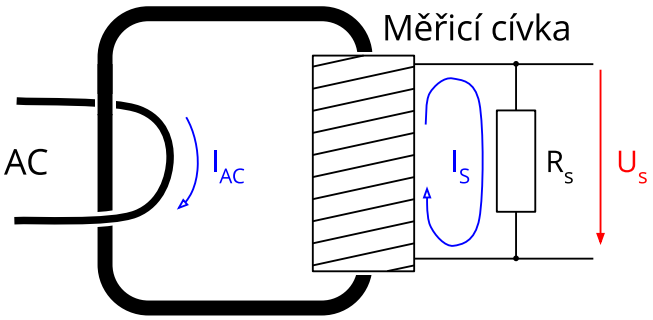
Obrázek 3.3: Proudový senzor s Hallovou sondou v pouzdru SOIC-8. *Zdroj: [4]*

Po zvážení kladů a záporů byl zvolen proudový transformátor, který je jednoduchý, snadno dostupný a nepotřebuje další izolaci, čímž zjednodušuje návrh plošného spoje.

3.2.2 Princip funkce proudového transformátoru

Proudový transformátor (obr. 3.4) sestává z uzavřeného jádra, na kterém je umístěno sekundární vinutí. Primární vinutí je tvořeno síťovým vodičem, který prochází skrz jádro a tvoří tak jediný závit.

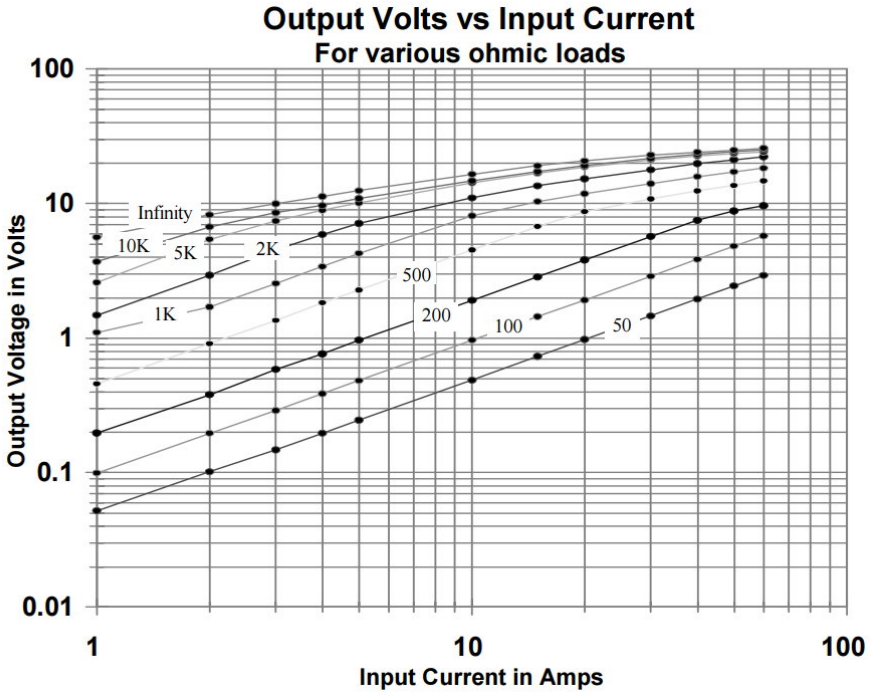
Abychom zjistili proud primárním vinutím, stačí změřit napětí na známém snímacím odporu, kterým protéká indukovaný proud. Neznámou hodnotu vypočteme z Ohmova zákona a převodního poměru transformátoru.



Obrázek 3.4: Schéma proudového transformátoru

3.2.3 Výpočet proudu z napětí na snímacím odporu

Při průtoku proudu I_{AC} primárním vinutím se na sekundárním vinutí indukuje proud I_{s}. Velikost proudu můžeme pak zjistit měřením napětí na snímacím odporu R_{s}. Napětí U_{s} je dle Ohmova zákona $U_{\mathrm{s}} = I_{\mathrm{s}} R_{\mathrm{s}}$.



Obrázek 3.5: Charakteristika převodního poměru AC-1010 v závislosti na zvoleném snímacím odporu; *Zdroj: Datasheet Talema AC-1010*

Doporučená velikost snímacího odporu je uvedena v datasheetu použité součástky, pro transformátor AC-1010, který byl zvolen, se doporučuje hodnota 100 Ω. Větší odpor by umožnil vyšší citlivost pro měření slabších proudů, ale zvýšila by se nelinearita, jak je zachyceno v grafu 3.5.

Pomocí transformačního poměru, který je uveden v datasheetu (pro AC-1010 to je 1:1000), můžeme vypočítat proud sekundárním vinutím I_s při daném primárním proudu I_p. Odtud pak vypočteme napětí na snímacím odporu U_s.

$$\frac{N_\mathrm{p}}{N_\mathrm{s}} = \frac{I_\mathrm{s}}{I_\mathrm{p}} \rightarrow I_\mathrm{s} = I_\mathrm{p}\frac{N_\mathrm{p}}{N_\mathrm{s}}$$

$$U_\mathrm{s} = I_\mathrm{p}\left(\frac{N_\mathrm{p}}{N_\mathrm{s}}R_\mathrm{s}\right) = I_\mathrm{p}k$$

Transformátor bohužel není ideální ani při použití doporučeného snímacího odporu, transformační poměr ještě závisí na proudu primárním vinutím. Dle datasheetu (při použití odporu $R_\mathrm{s} = 100\,\Omega$) je $k = 0.1\,\mathrm{V/A}$ při proudu 10 A, ale při 1 A už je $k = 0.097\,\mathrm{V/A}$. Protože cílem je měřit především slabší elektrické spotřebiče, jako například ventilátory, budu počítat s hodnotou pro nižší proud, $0.097\,\mathrm{V/A}$.

Je žádoucí, aby byl využit celý rozsah A/D převodníku, čímž se zajistí co nejmenší kvantizační chyba. Napětí je proto před vstupem do převodníku zesíleno operačním zesilovačem.

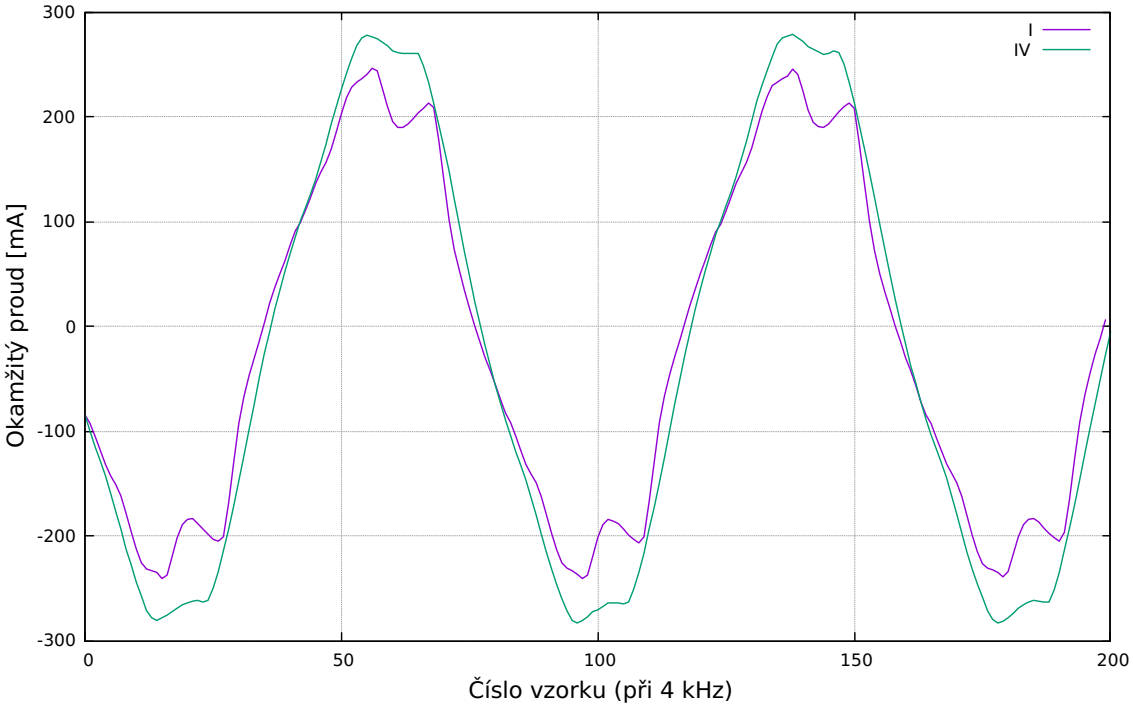
Vstupní rozsah převodníku je přibližně 3.3 V, přičemž umělá nula[1] (pro měření záporných hodnot) se nachází v polovině jeho rozsahu, tj. amplituda po zesílení může být nejvýše 1.65 V. Musíme zvolit koeficient takový, aby nedocházelo k ořezávání signálu. Ze spotřebičů použitých k experimentálnímu měření (různé ventilátory a domácí spotřebiče) měl největší špičkový odběr motor z mixéru, a to 1.37 A. Ventilátory mají typicky příkon do 50 W, což odpovídá mnohem menšímu proudu. V navrženém zapojení byl zvolen koeficient zesílení $\alpha = 10$, při kterém měřitelný rozsah dosahuje přibližně ± 1.7 A. Pro běžné spotřebiče by stačilo zesílení menší, jedná se o kompromis mezi univerzálností a přesností.

Z napětí na vstupu A/D převodníku (tj. po zesílení) je možné snadno vypočítat proud primárním vinutím. Tento vzorec je použit ve firmwaru pro zpracování naměřených hodnot (k – převodní poměr, α – koeficient zesílení):

$$U_{\mathrm{ADC}} = \alpha \cdot U_{\mathrm{s}} = \alpha \left(I_{\mathrm{p}} k\right) \rightarrow I_{\mathrm{p}} = \frac{U_{\mathrm{ADC}}}{\alpha \cdot k}$$

3.3 Analýza změřeného průběhu proudu

Úkolem zařízení je detekovat změny v průběhu proudového odběru. Pokud signál jednoduše zobrazíme v grafu, změny jde někdy pozorovat pouhým okem (obr. 3.6), jindy jsou nepatrné a těžko bychom je odhalili.



Obrázek 3.6: Porovnání odběru pokojového ventilátoru při různých rychlostech

[1] Umělá nula umožňuje měřit obě polarity napětí na snímacím odporu; detailně je to popsáno v kapitole 4.2.1

3.3.1 Frekvenční analýza

Pro získání spektra signálu byla zvolena metoda FFT, což je obvyklý nástroj používaný k diagnostice charakteristik střídavého signálu. Výsledkem je vektor komplexních čísel, ze kterého jsou pro zobrazení a další vyhodnocování vypočteny magnitudy jednotlivých složek. V grafu spektra jsou obvykle výrazné vyšší harmonické síťové frekvence 50 Hz.

Existují i jiné metody analýzy, například kosinová transformace, některé zdroje zmiňují možnost použít tzv. cepstrální analýzu. Pro jednoduchou detekci změn spektra FFT stačí, navíc je možné použít některou z dostupných matematických knihoven. Ostatní metody jsou méně používané, proto by byl problém je realizovat.

Kromě spektra je ještě užitečné zjistit skalární parametry signálu, jako je např. efektivní hodnota a maximální velikost proudu ve vzorku. Tyto hodnoty se dají snadno vynést do grafu a je možné je použít pro případnou další rozhodovací logiku. Efektivní hodnota (RMS) se vypočte podle následujícího vzorce (i_k je k-tý vzorek okamžitého proudu, n je počet vzorků):

$$I_{\mathrm{RMS}} = \sqrt{\frac{1}{n}\sum_{k=0}^{n} i_k^2}$$

3.3.2 Detekce změn charakteristiky odběru spotřebiče

V autonomním režimu je potřeba detekovat změny charakteristiky, které mohou indikovat poruchu. Jedná se o komplexní problém, který je možné řešit více způsoby.

Začneme změřením spektra odběru spotřebiče, které označíme jako referenci (A). Toto spektrum můžeme uložit do paměti a porovnávat s ním budoucí hodnoty. Jak ale vyčíslit odchylku nově změřeného spektra B od reference A? Je třeba zvážit, k jakým změnám může dojít:

- **Změna magnitudy složek** – nastává při změně zátěže, může být například projevem zadření ložisek ventilátoru. Při větší zátěži obvykle roste magnituda první harmonické (50 Hz) na úkor ostatních.
- **Frekvenční posun složek** – u motorů může naznačovat změnu rychlosti. Tento jev byl pozorován u sériového motoru při rozběhu a přibrzďování, jinak je poměrně vzácný.
- **Úplná změna charakteristiky** – tato změna nastane např. pokud se změní funkční režim zařízení (stand-by / zapnuto / vypnuto). Takovou změnu lze těžko vyjádřit číselně; projeví se jako kombinace předchozích dvou typů změn.

Pro číselné vyjádření změn jsem navrhl vzorec inspirovaný Hausdorffovou metrikou [5]. Tato metrika počítá vzdálenost dvou prostorových útvarů jako největší z nejmenších vzdáleností mezi body obou útvarů. Vzorec bylo nutné modifikovat, aby zahrnul změny ve všech částek útvaru (zde graf spektra) a ne jen v bodě, kde je vzdálenost nejmenší. Dále

jsem chtěl, aby se ve výsledku výrazněji projevily větší změny magnitudy, což zajišťuje kvadrát ve vzorci (respektive chybějící odmocnina ve výpočtu vzdálenosti bodů).

$$\mathrm{dist}(A,B) = \sum_{i=0}^{n} \min_{0 \le j < n} \left(w_{\mathrm{y}}(A_i - B_j)^2 + w_{\mathrm{x}}(i-j)^2 \right)$$

Při realizaci tohoto algoritmu stačí pracovat s $j \in \langle i - x_{\mathrm{d}}; i + x_{\mathrm{d}} \rangle$, čímž se zrychlí výpočet a hodnota se při vhodné volbě x_{d} prakticky nezmění. Parametry x_{d}, w_{x} a w_{y} výrazně ovlivňují chování metriky. Nastavíme je dle potřeby, například pokud chceme dát větší význam frekvenčnímu posunu složek, zvýšíme w_{x}. Musíme si však dát pozor, aby pak posun nebyl interpretován jako změna amplitudy sousedních složek.

3.4 Analýza možností WiFi čipu ESP8266

Zde bych rád představil WiFi čip ESP8266, který jsem v projektu použil. Jedná se o 32-bitový procesor čínského výrobce Espressif, který je možné použít jako WiFi adaptér nebo dokonce hlavní procesor pro jednoduché aplikace. ESP8266 je velmi populární v IoT komunitě a existuje pro něj řada knihoven a projektů.

Veřejně dostupná dokumentace tohoto obvodu není příliš kvalitní, návody a manuály jsou roztroušeny po diskuzních fórech [6], komunitní wiki [7] a dalších stránkách; proto považuji za užitečné zde shrnout základní informace.

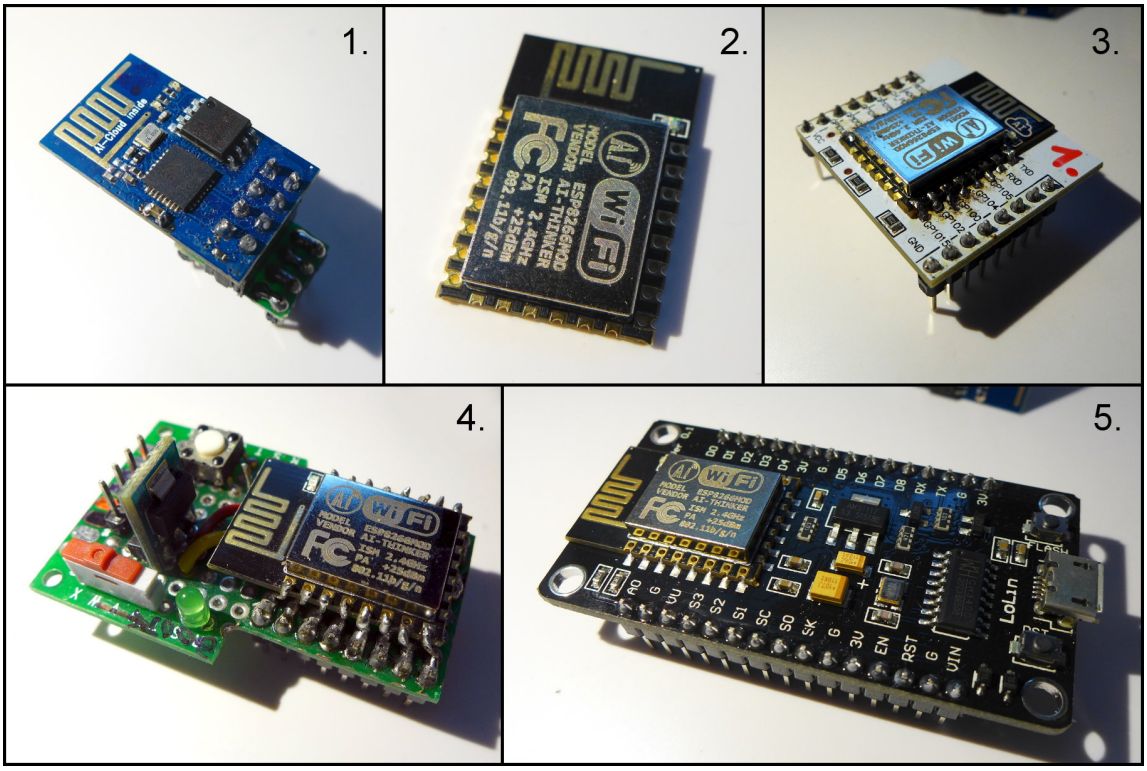
Jedná se o programovatelný 32-bitový procesor (resp. bootloader ve vnitřní ROM dokáže programovat externí programovou paměť Flash) s vestavěnou podporou WiFi. Je tedy možné napsat vlastní firmware a chování obvodu přizpůsobit potřebám projektu. Do Flash paměti lze kromě programu uložit HTML soubory pro webové rozhraní, javascriptové knihovny, obrázky a podobně.

Kromě vynikajících vlastností obvodu je výhodou také cena: Moduly ESP-01 a ESP-12 lze pořídit kolem 50 Kč na online tržištích jako *eBay.com* či *AliExpress.com*, za (mnohem) vyšší cenu pak i u českých překupníků (informace z r. 2016).

3.4.1 Architektura obvodu ESP8266

Obvod ESP8266 je postaven na jádře *Xtensa LX106*, což je architektura licencovaná firmou Tensilica různým čínským výrobcům, kteří si ji dále přizpůsobí pro své obvody. V principu je to podobný systém jako u firmy ARM, kde tato pouze prodává licence a nic nevyrábí.

Firmware má k dispozici 160 kB interní RAM, z toho 96 kB slouží jako datová paměť, zbytek je instrukční cache. Čip neobsahuje paměť programu, pouze bootloader v ROM, všechny moduly jsou proto vybaveny externí pamětí Flash, která je připojena rozhraním SPI. Velikost této paměti je typicky 1 MB, u některých modulů více.



Obrázek 3.7: Moduly s ESP8266: (1) ESP-01, (2) ESP-12, (3) Adaptér do nepájivého pole pro ESP-12, (4) Přípravek s ESP-12, (5) Vývojový kit pro připojení k PC přes USB

3.4.2 Dosah a funkce WiFi

WiFi driver v obvodu ESP8266 podporuje režim klienta i AP, dokonce umí obojí najednou. Režim AP je vhodný hlavně k prvotní konfiguraci a zapojení do sítě, pro běžný provoz je vhodnější režim klienta. V AP režimu totiž musí běžet DHCP server a další služby, které u klienta nejsou potřeba. Kvůli této režii je procesor v AP režimu pomalejší a někdy může být i nestabilní. AP režim je také náročnější na napájení.

Co se týče dosahu WiFi, dle experimentů [8] se moduly dokáží připojit k modemu na vzdálenost až 400 metrů, pochopitelně v ideálních podmínkách. Funkce AP stačí k pokrytí většího bytu, což pro běžné použití stačí.

3.4.3 Periferie obvodu

Kromě samotného WiFi driveru obvod dále obsahuje řadu užitečných periferií, v jednoduchých aplikacích tak ESP8266 dokáže zastat roli hlavního procesoru.

Periferie zahrnují:

- Řadu vstupně-výstupních pinů (GPIO)
- SPI rozhraní (k připojení programové paměti)
- Rozhraní I^2S

- Komunikační USART a jednosměrný UART pro ladící výpisy
- 10-bitový A/D převodník s rozsahem 1 V
- Řadič DMA, hardwarové časovače, watchdog

K některým periferiím bohužel chybí dokumentace, například wiki zmiňuje JTAG rozhraní, ale odkazovaná stránka neexistuje. Většinou však je možné na najít nějaký ukázkový program.

3.4.4 Možnost vytvoření vlastního firmwaru

Výrobce poskytuje SDK [9] pro vývoj vlastního firmwaru[2] spolu s řadou knihoven, které obsluhují hardware a starají se o nízkoúrovňovou režii, jako je ovládání WiFi driveru a implementace nižších vrstev TCP/IP. Je zde k dispozici jednoduchý DHCP server, funkce pro komunikaci s DNS, podpora pro protokol NTP a řada dalších užitečných funkcí. Většina SDK bohužel není open-source, jedná se o statické knihovny a hlavičkové soubory; z praktického hlediska to ale nevadí.

Moduly s obvodem ESP8266 jsou obvykle dodávány s nahraným firmwarem, který umožňuje čip ovládat pomocí AT příkazů. Jedná se v podstatě o rozhraní mezi UARTem a funkcemi SDK. Můj původní plán, když jsem poprvé obvod připojil k hlavnímu procesoru, byl ovládat SDK právě pomocí AT příkazů a serverovou logiku implementovat v hlavním procesoru. Brzy se však ukázalo, že tudy cesta nevede – program byl příliš složitý a nespolehlivý. AT příkazy stačí pro jednoduché projekty, ale pro složitější aplikace je výhodnější napsat si firmware vlastní.

K vývoji firmwaru je výhodné použít projekt *esp-open-sdk* [10], který automaticky stáhne a připraví vývojové prostředí včetně SDK. Stojí za zmínku, že na internetu je možné najít celou řadu neoficiálních firmwarů, které lze použít místo psaní firmwaru vlastního. Například existuje firmware pro Lua skripty („NodeMCU“), Arduino firmware, MicroPython a mnohé další.

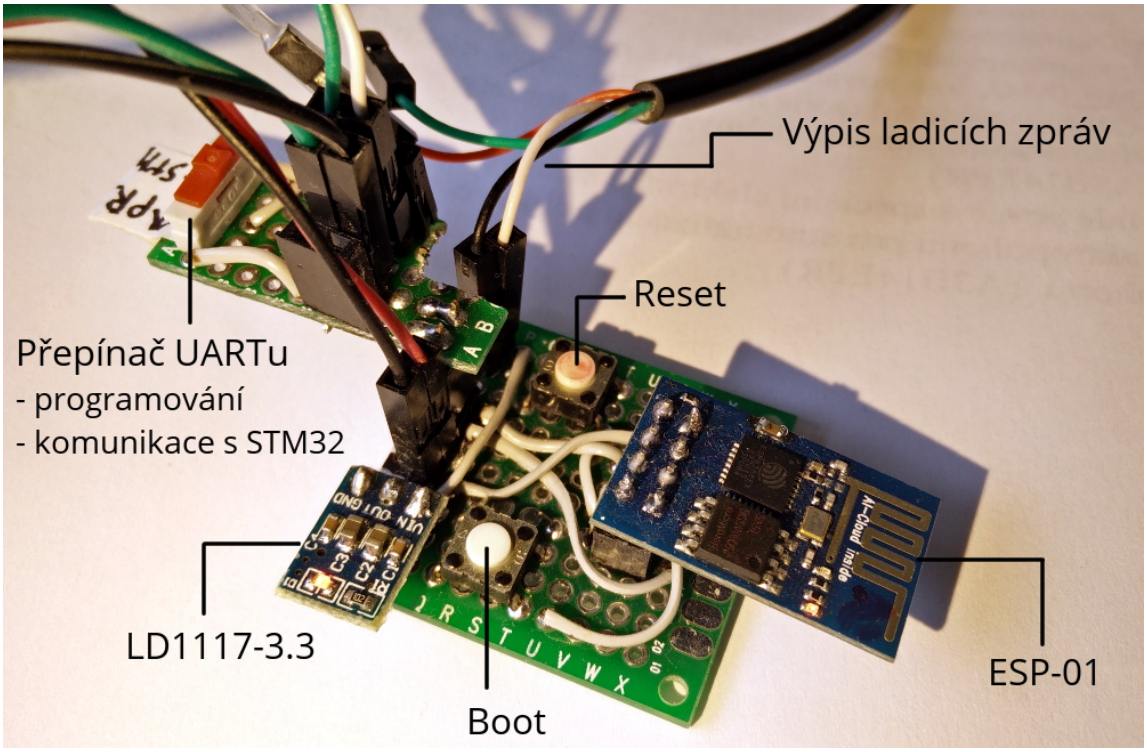
3.4.5 Nahrání firmwaru do Flash paměti

K nahrání zkompilovaného firmwaru do modulu slouží nástroj „ESP8266 Flash Download Tool“ [11]. Na Linuxu je k dispozici open-source program „esptool.py“ [12].

Pro upload firmwaru zapojíme ESP modul podle schématu na obr. 3.9 resp. 3.10. Na obr. 3.8 je ukázka jednoduchého přípravku pro programování modulů ESP-01.

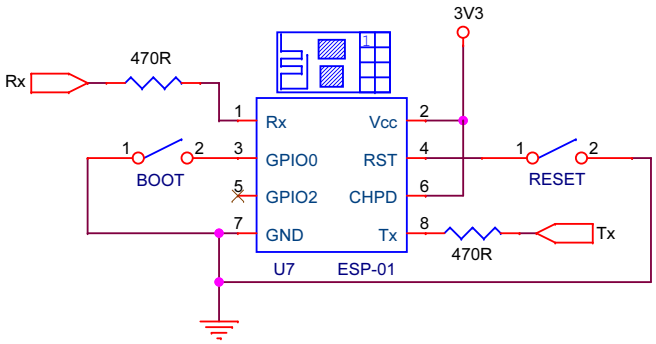
Upload probíhá prostřednictvím rozhraní UART, stačí tedy použít jakýkoliv USB-serial adaptér, např. s obvodem FTDI. Modul je napájen napětím 3.3 V; piny UARTu vydrží 5 V, ale doporučuje se použít ochranný odpor (na schématu je zakreslen odpor 470 R jako příklad).

[2] Na odkazované stránce je k dispozici více variant, nejjednodušší je verze „Non OS SDK“.

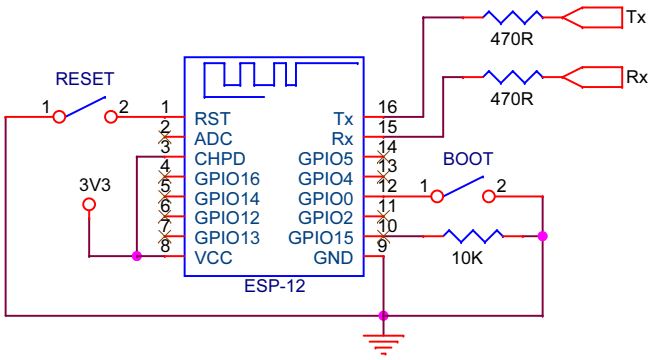


Obrázek 3.8: Jednoduchý přípravek pro programování ESP-01 a vývoj firmwaru.

Aby bylo možné nahrát firmware, musíme napřed aktivovat bootloader, což se provede připojením pinu GPIO0 na GND během resetu (tj. během resetu držíme stisknuté tlačítko BOOT).



Obrázek 3.9: Základní zapojení modulu ESP-01



Obrázek 3.10: Základní zapojení modulu ESP-12

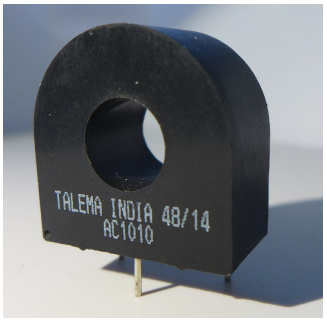
Kapitola 4

Hardwarová realizace

4.1 Volba komponent

4.1.1 Součástky analogové části obvodu

Pro měření proudu byl zvolen proudový transformátor Talema AC-1010, který je snadno dostupný a dokáže měřit proud až 10 A. Způsob měření proudu proudovým transformátorem vysvětluje podkapitola 3.2, ve zkratce se jedná o měření napětí na snímacím odporu, kterým protéká indukovaný proud.



Obrázek 4.1: Proudový transformátor AC-1010

Napětí je před digitalizací nejprve nutné zesílit, to zajišťuje operační zesilovač AD8544[1]. Zkušební prototyp přístroje byl osazen zesilovačem OP213, ten ale potřebuje napájení 5 V, takže v zapojení byl potřeba zvláštní regulátor. Obvod AD8544 je napájen napětím 3.3 V, stejně jako zbytek zapojení, stačí tedy regulátor pouze jeden.

4.1.2 Hlavní procesor

Důležitou součástí zapojení je hlavní procesor, který provádí vlastní měření a vyhodnocování. Pro tuto úlohu byl zvolen procesor STM32 F303 CBT.

Řada procesorů STM32 F3 je postavena na jádru ARM Cortex M4 a obsahuje FPU[2], což je výhodné pro výpočet spektra a další zpracování signálu. Absence FPU při použití nižšího modelu procesoru (například z řady STM32 F0) by znamenala, že floatová aritmetika bude realizována softwarově místo použití zvláštních instrukcí, což je pomalejší a velikost programu vzroste.

4.1.3 Volba WiFi modulu

Původně jsem zvažoval použít WiFi modul s čipem CC3100 od firmy Texas Instruments, naštěstí jsem ale včas objevil obvod ESP8266 (podrobně popsán v podkapitole 3.4), který

[1] Obvod AD8544 obsahuje 4 operační zesilovače, z toho byly použity pouze dva. Stačilo by samozřejmě varianta obvodu se dvěma zesilovači.

[2] *Floating Point Unit* - hardwarová jednotku pro práci s desetinnými čísly

je jednodušší a více používaný v IoT[3] projektech. Díky tomu je o něm k dispozici řada materiálů a projektů, které mi pomohly s vývojem firmwaru.

Na trhu je více modulů s čipem ESP8266, nejběžnější jsou ESP-01 a ESP-12. Oba mají své výhody i nevýhody:

- Modul ESP-12 má vyvedenou většinu pinů, které lze využít jako GPIO, má FCC certifikaci a je určen pro SMD montáž, proto je vhodnější do komerčních výrobků.
- Modul ESP-01 je o něco levnější a díky konektoru (8-pinový header) jej lze s jednoduchým adaptérem připojit i do nepájivého pole; Chybí mu certifikace, je tedy výhodný spíše pro vývoj a hobby projekty.

Z praktických důvodů (lepší ladění, snadná výměna, menší footprint) byl zvolen modul ESP-01. Díky tomu, že je připojen pomocí konektoru, je možné mít více modulů s různými verzemi firmwaru (například různé lokalizace uživatelského rozhraní) a vyměňovat je dle potřeby.

4.1.4 Napájení

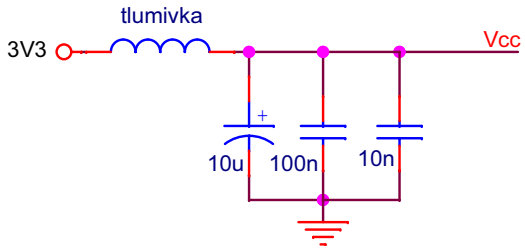
Měřicí přístroj má být vestavěn do zásuvkového adaptéru, je tedy třeba zajistit napájení přímo ze sítě – k tomu byl zvolen modul síťového spínaného zdroje IRM-10-5 (výstup 5 V) značky Mean Well.

Všechny použité čipy pracují s napájením 3.3 V, což zajišťuje lineární regulátor LD1117-3.3. Odběr obvodu se pohybuje kolem 100 mA, to odpovídá ztrátovému výkonu 170 mW při napěťovém úbytku 1.7 V. Pro chlazení tedy bohatě stačí měděná ploška, což bylo ověřeno na zkušebním prototypu.

Dále je nutné zajistit, aby nedocházelo k vzájemnému rušení jednotlivých částí obvodu, například proudové špičky z WiFi modulu by mohly způsobit pokles napájecího napětí a ovlivnit přesnost měření (to se stávalo u prototypu, kde bylo napájení zapojeno nevhodně). K odrušení slouží tlumivky a blokovací kondenzátory; induktor působí při proudových impulzech jako impedance a potřebná energie je dodána kondenzátory, které připojíme co nejblíž k integrovanému obvodu. Obvykle se kombinuje několik různých kapacit, například 10 µF, 10 nF a 100 nF, aby se pokryly různě dlouhé a silné pulzy. V případě WiFi modulu je třeba dát si pozor na stejnosměrný odpor tlumivky, který se často pohybuje v řádu ohmů a při větším odběru dochází k velkému úbytku napětí.

Během oživování a ladění firmwaru z bezpečnostních důvodů není vhodné, aby na desce bylo síťové napětí. To je zajištěno pomocí konektoru pro připojení externího zdroje a přepínače pro výběr napájení; je tak možné použít laboratorní zdroj nebo i baterii.

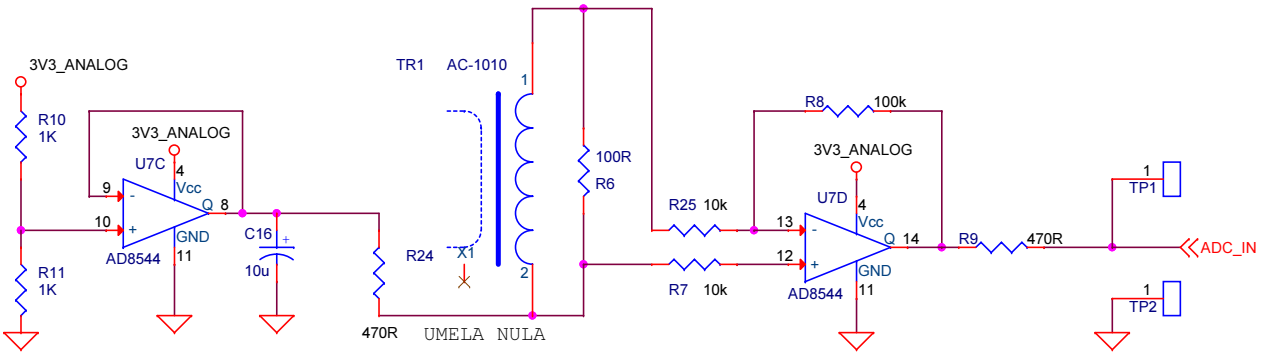
[3] *Internet of Things* – „chytrá“ do sítě připojená zařízení, často s cloudovou integrací



Obrázek 4.2: Příklad odrušení napájení integrovaného obvodu

4.2 Zapojení analogové části obvodu

Princip měření je popsán v podkapitole 3.2. Ve zkratce: proud primárním vinutím lze měřit jako napětí na zatěžovacím odporu, které je dále zesíleno operačním zesilovačem. Kompletní zapojení je znázorněno na obrázku 4.3.



Obrázek 4.3: Zapojení analogové části přístroje

4.2.1 Měření záporného napětí na snímacím odporu

Protože proud tekoucí primárním vinutím je střídavý, bude napětí na snímacím odporu nabývat kladných i záporných hodnot. Abychom mohli měřit obě půlperiody, můžeme jeden konec cívky připojit na potenciál uprostřed měřitelného rozsahu, který označíme *umělá nula*. Snímané napětí pak bude rozdílem umělé nuly a napětí na odporu. K vytvoření umělé nuly je použit odporový dělič, dále posílený sledovačem napětí.

Se změnou nulového potenciálu však vzniká problém – jaká hodnota na výstupu A/D odpovídá nule? Je jasné, že to nebude přesně 1.65 V; rezistory v děliči nejsou přesně stejné a nemůžeme také zanedbat napěťový offset zesilovače. Ideální by bylo do obvodu zařadit multiplexer a během pravidelné kalibrace vstup připojit na umělou nulu, pak by její hodnotu šlo přímo změřit. Tím se však komplikuje zapojení.

Existuje druhá metoda, která se obejde beze změn zapojení, ale je méně přesná. Využijeme zde faktu, že střední hodnota střídavého proudu je nulová; za nulový potenciál tedy budeme považovat střední hodnotu naměřeného průběhu. Tento způsob vypadá na první pohled elegantně, ale pokud doba měření neodpovídá násobku periody vstupního signálu (20 ms

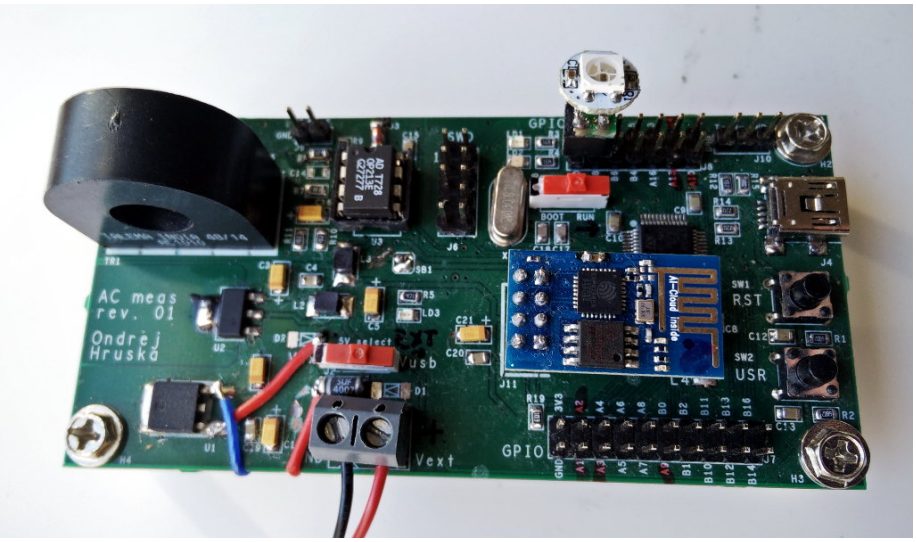
při 50 Hz), vznikne nepřesnost. Pro dostatečně dlouhé průběhy (několik period) tuto chybu můžeme zanedbat, protože se dostává pod rozlišovací schopnost převodníku.

4.3 Realizace prvního prototypu

Za účelem ověření funkce měření a validace zapojení jsem sestrojil první prototyp přístroje, který je v konečné podobě vyobrazen na obrázku 4.4. Prototyp umožnil vývoj firmwaru ještě před vyrobením finální desky.

Jedná se o můj vůbec první návrh plošného spoje, v zapojení tak došlo k několika chybám způsobených nezkušeností, přesto ale po drobných úpravách deska funguje jak má a svůj účel splnila. Hlavním problémem bylo nestabilní napájení způsobené špatně odrušeným WiFi modulem, který byl do zapojení přidán dodatečně jako experiment. Napájení dále degradovalo chybné zapojení regulátorů, které byly spojeny do série (v prototypu je potřeba napájení 5 V pro operační zesilovač OP213, proto byl nutný druhý regulátor). Tyto chyby byly samozřejmě opraveny ve finálním zapojení.

Kromě změny typu operačního zesilovače a úprav napájení se zapojení prototypu převážně shoduje s finální deskou. Schéma a výrobní podklady obou desek jsem přiložil na CD.



Obrázek 4.4: Fotografie prvního prototypu měřicího přístroje

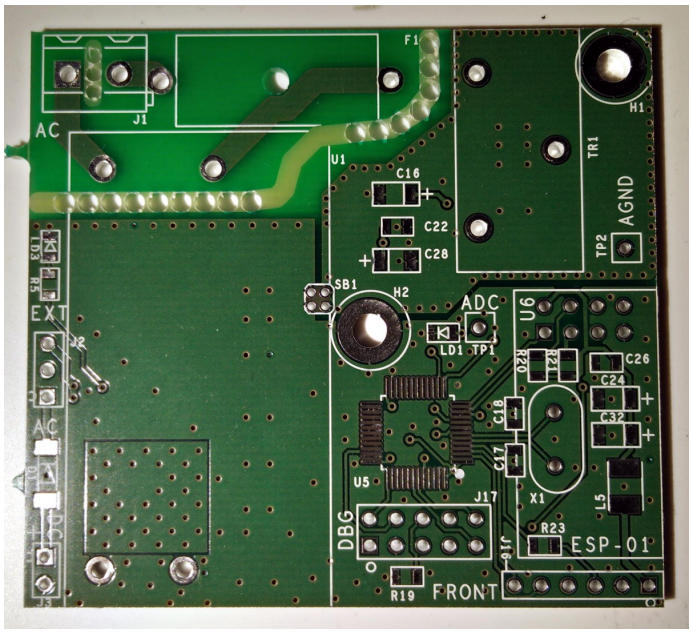
4.4 Návrh finální desky plošných spojů

Pro přístroj byla zvolena standardní plastová krabička se zástrčkou na jedné a zásuvkou na druhé straně. Téměř polovinu prostoru zabral napájecí modul, nezbylo tak než přistoupit k několika kompromisům, například umístit součástky na obě strany desky a vypustit resetovací tlačítko, které je nakonec pouze vyvedeno na přední panel.

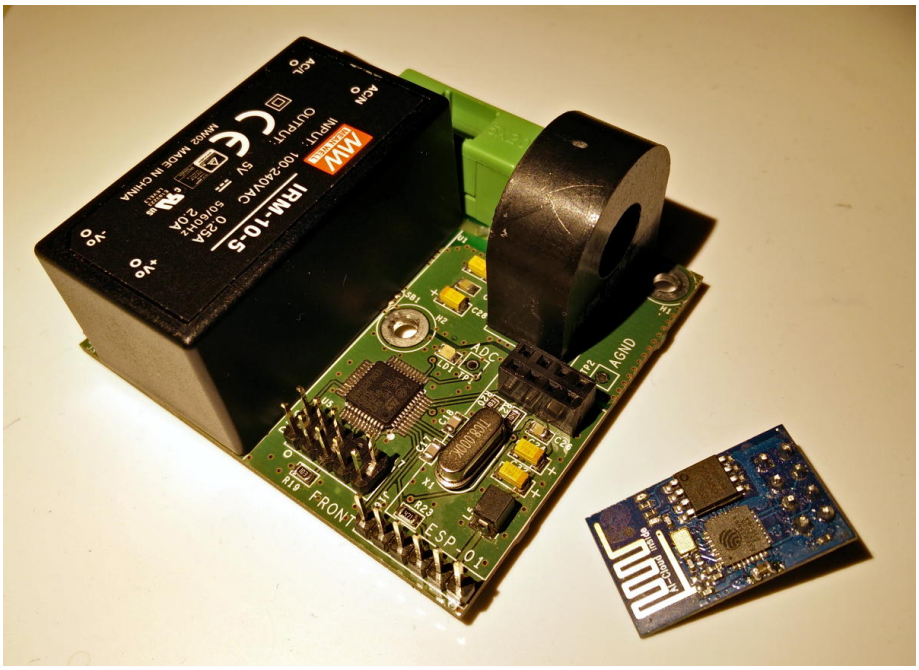
Deska je vybavena konektorem „DBG“, který slouží k programování procesoru, čtení ladicích výpisů a k aktivaci bootloaderu pomocí pinu BOOT0. Dále je tu konektor „FRONT“

pro připojení předního panelu, na který jsou vyvedeny všechny piny pro hardwarové uživatelské rozhraní (o něm více v podkapitole 6.4).

Síťová část desky je oddělena od měřicí logiky velkou izolační vzdáleností, navíc je tu odkryta pájecí maska a v místech, kde je izolační vzdálenost nejmenší, je část substrátu odvrtána. Tím se omezí případné plazivé proudy a další nežádoucí jevy. Bezpečnost dále zajišťuje pojistka a ochranné prvky vestavěné v napájecím modulu.



Obrázek 4.5: Vyhotovená finální verze desky plošných spojů před osazením



Obrázek 4.6: Deska plošných spojů po osazení; vpravo: odpojený WiFi modul ESP-01

Obrázek 4.7: Fyzické uložení desky plošných spojů v adaptérové krabičce; vlevo: odpojený přední panel

Kapitola 5

Návrh firmwaru

V této kapitole bude představena struktura a funkce firmwaru hlavního procesoru a WiFi modulu. Realizace uživatelského rozhraní bude popsána v kapitole následující.

5.1 Záznam a zpracování signálu

5.1.1 Záznam signálu

Pro digitalizaci signálu byl zvolen 12-bitový A/D převodník, který je součástí hlavního procesoru. Převodník dokáže vzorkovat s frekvencí až 5 MHz, což by ale bylo zbytečné pro pomalé signály, které budeme měřit, protože všechny významné složky jsou v řádu několika kilohertz. Nižší vzorkovací frekvence navíc pomůže potlačit šum.

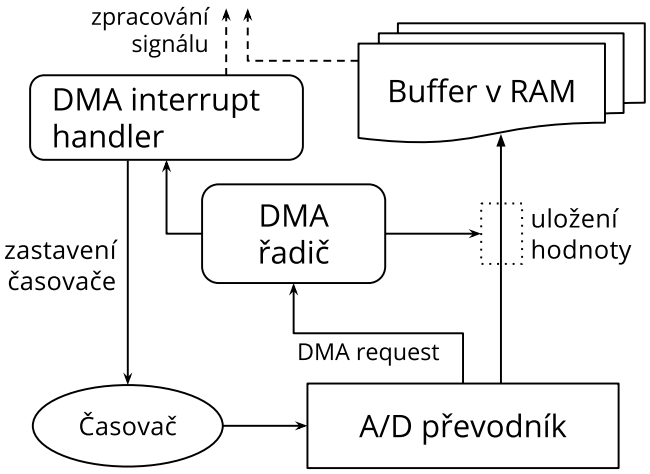
Pokud se například omezíme na šířku pásma 2 kHz, dle Nyquistova teorému musí být vzorkovací frekvence alespoň dvojnásobná, tedy 4 kHz. Pro výpočet FFT (a zobrazení grafu) je vhodné, aby frekvence byla násobkem počtu bodů (mocnina dvou), tj. například 4096 Hz nebo 5120 Hz. Skutečnou vzorkovací frekvenci bude samozřejmě možné zvolit v uživatelském rozhraní.

Záznam a zpracování signálu ve firmwaru realizuje funkční blok, který je aktivován buď externě z uživatelského rozhraní, nebo subsystémem autonomního monitorování. V autonomním režimu se pravidelně kontroluje stav spotřebiče a hodnoty jsou odesílány na monitorovací server. Tato funkce bude blíže vysvětlena v podkapitole 6.3.

K záznamu průběhu proudu je použit řadič DMA[1], který se stará o kopírování výstupu A/D převodníku do záznamového bufferu a počítání vzorků. Mechanismus záznamu pomocí DMA (obr. 5.1) je poměrně komplikovaný a vyžaduje správné nastavení všech tří periferií (časovač, DMA kanál, A/D převodník). Výhodou je, že během odměru se program může věnovat jiným činnostem, jako je například komunikace s WiFi modulem nebo obsluha uživatelského rozhraní. To má význam především u dlouhých záznamů.

Vzorkovací frekvenci udává hardwarový časovač, jehož prescaller[2] a čítací rozsah jsou předem zvoleny tak, aby se frekvence co nejvíce blížila požadovanému nastavení. Zvolenou frekvenci nelze vždy dosáhnout přesně, odchylka je však většinou zanedbatelná. Skutečná vzorkovací frekvence se pro kontrolu zobrazí v uživatelském rozhraní.

[1] Direct Memory Access – paměťová operace prováděná „na pozadí“, bez aktivní účasti procesoru
[2] prescaller je hardwarový blok, který dělí hodinový signál zvoleným koeficientem



Obrázek 5.1: Princip záznamu pomocí DMA

Po dokončení odměru je vyvoláno přerušení, při kterém se zastaví časovač a deaktivuje DMA kanál. Na tuto událost čeká zbytek měřící logiky, jak je znázorněno na obrázku 5.2. Hodnoty jsou převedeny na datový typ float, přepočteny na miliampéry a je provedena kompenzace nulové hodnoty, jak bylo popsáno v podkapitole 4.2.1. Po dalším zpracování (výpočet parametrů signálu, příp. FFT) je výsledek předán WiFi modulu pro odeslání do prohlížeče nebo na monitorovací server.

5.1.2 Vzorkovací buffer

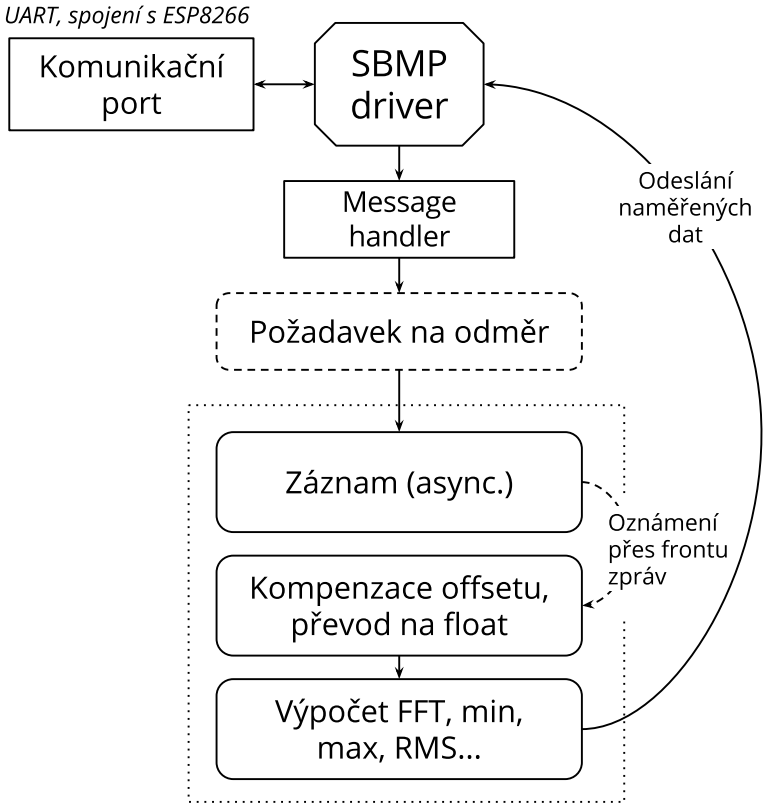
Délka bufferu, kam řadič DMA ukládá vzorky, je prakticky omezena velikostí dostupné paměti; použitý procesor má pouze 32 kB. Aby nedocházelo k plýtvání pamětí, jsou vzorky ukládány do pole typu `uint32_t[]`, které sdílí paměťovou oblast s polem typu `float[]` pro výsledky výpočtu (viz obrázek 5.3). Zvolil jsem délku 4096, což odpovídá 16 384 B. Pro ukládání 12-bitových vzorků by stačilo pole typu `uint16_t[]`, stejně pak ale budeme potřebovat 32 bitů pro float a konverze z 32-bitového integeru na float je jednodušší než kdyby byla použita 16-bitová čísla.

Tímto bufferem je zabrána polovina paměti, zbytek slouží k uložení globálních datových struktur a jako stack. Může se zdát, že zbylo hodně místa, ale velkou část paměti zabírají struktury knihovny *newlib*[3], dále je potřeba nějaké místo pro komunikační buffery a podobně.

5.1.3 Výpočet FFT

Jako hlavní nástroj pro analýzu signálu byla zvolena Fourierova transformace, vypočtená algoritmem FFT. To je realizováno funkcemi z *ARM DSP Library*, dodávané spolu s knihovnou *CMSIS*.

[3] *newlib* je přenositelná implementace *libc*



Obrázek 5.2: Blokový diagram záznamu a zpracování signálu v STM32

```
#define SAMP_BUF_LEN 4096

union samp_buf_union {
        uint32_t uints[SAMP_BUF_LEN];
        float    floats[SAMP_BUF_LEN];
        uint8_t  bytes[SAMP_BUF_LEN * 4];
};

extern union samp_buf_union samp_buf;
```

Obrázek 5.3: Deklarace vzorkovacího bufferu, tři způsoby přístupu ke stejné paměťové oblasti

Nároky FFT na buffer

Pro výpočet FFT slouží stejný buffer, do kterého jsou ukládány vzorky (obr. 5.3), avšak funkce pracuje s komplexními čísly, kapacita je tedy poloviční[4]. Z toho vyplývá omezení na 1024 bodů FFT pro buffer délky 4096: je potřeba 2048 vzorků, z toho každý jako komplexní číslo zabere dva indexy.

Kromě funkcí pro komplexní FFT jsou v knihovně též funkce optimalizované pro reálná čísla, nepodařilo se mi je ale použít k dosažení stabilních výsledků – magnituda náhodně kolísala i pro přesný signál z generátoru. Funkce pro komplexní čísla (`cfft`) naproti tomu fungují spolehlivě, nevýhodou je plýtvání místem v bufferu.

Po výpočtu komplexní FFT program spočítá magnitudu všech bodů a tyto magnitudy jsou použity k dalšímu vyhodnocení a zobrazení.

Normování FFT

Při výpočtu různě dlouhých FFT vektorů se magnitudy bodů značně liší (to je způsobeno různým počtem vzorků, což ovlivňuje celkovou energii jednotlivých složek). Aby bylo možné výsledek použít pro zobrazení spektrogramu nebo grafu, je nutné složky normovat.

Normování je realizováno jednoduchým vzorcem: každá složka je vydělena celkovou délkou vektoru. Výsledné hodnoty jsou pak téměř stejné pro různé délky FFT, z čehož lze usuzovat, že výpočet je správný.

Okénková funkce

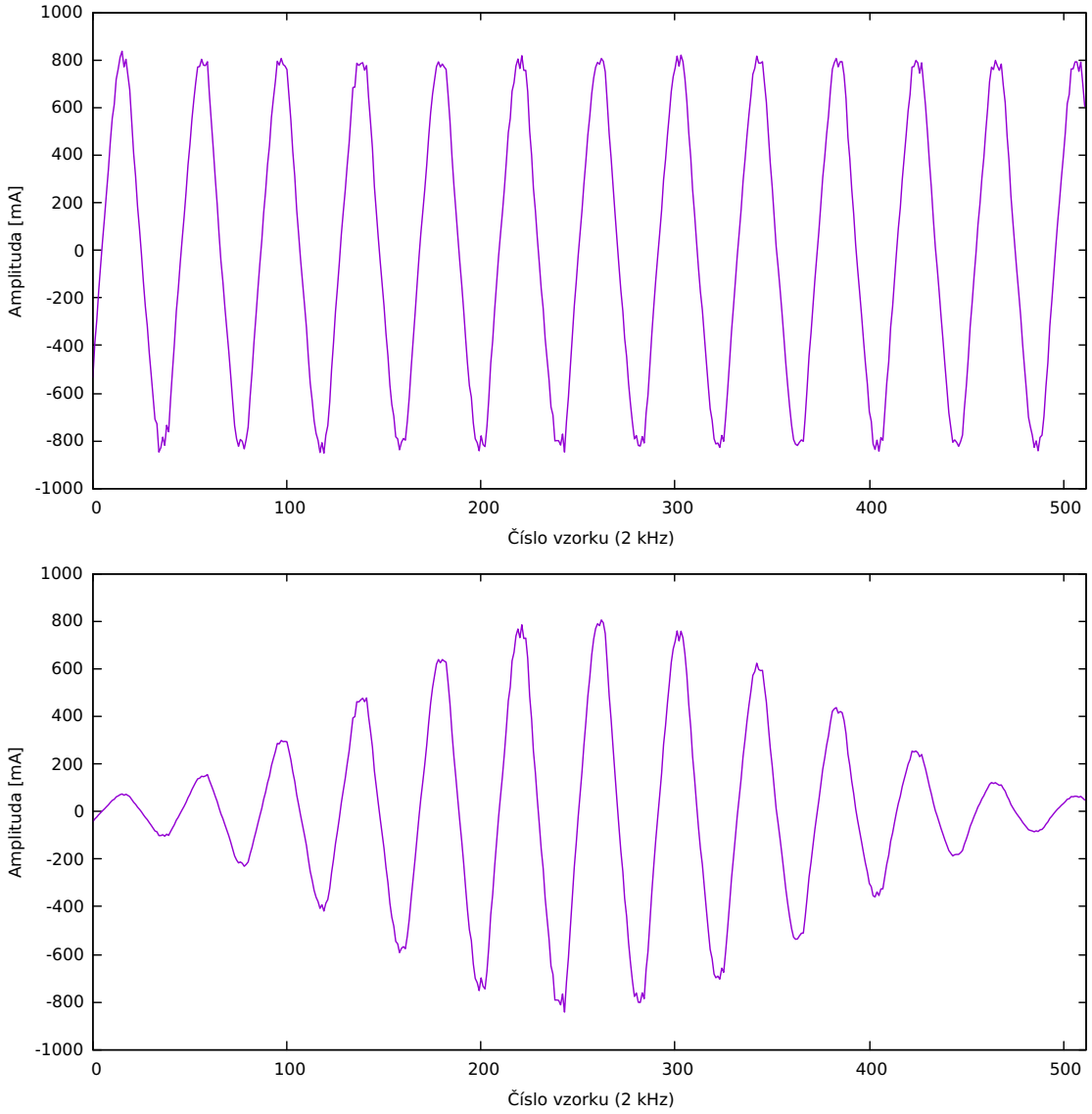
Chceme-li výpočtem FFT získat čisté spektrum, musí být signál periodický; pokud tomu tak není, bude výsledek vypadat rozostřeně. V ideálním případě, kdy je délka záznamu přesně násobkem periody signálu (20 ms pro síťovou frekvenci 50 Hz), bude výsledek perfektní. Ve skutečnosti však frekvence v síti není přesně 50 Hz a uživatel navíc může volit libovolnou délku odměru, v záznamu tak vznikne nespojitost.

Aby se potlačily negativní vlivy nespojitosti na začátku a konci záznamu, je nutné použít okénkovou funkci, která amplitudu signálu na okrajích sníží; tento princip je použit ve většině osciloskopů s funkcí FFT. Zkusil jsem několik variant a jako nejlepší se ukázalo Hammingovo okno. Vektor okénkové funkce byl vygenerován v Matlabu (obr. 5.4). Na obrázku 5.5 je zachycen vliv okénkové funkce na signál.

```
w=hamming(512).'
dlmwrite('ham512.csv', w, 'delimiter', ',', 'precision', '%0.8f')
```

Obrázek 5.4: Kód pro vygenerování vektoru okénkové funkce a uložení výsledku do CSV souboru

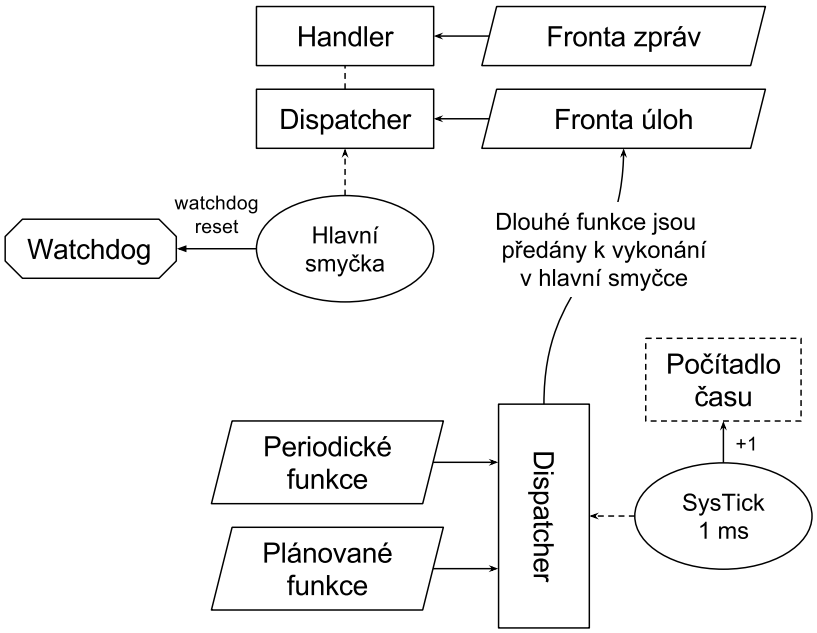
[4] Komplexní čísla jsou reprezentována jako `{real1, imag1, real2, imag2, ...}`



Obrázek 5.5: Ukázka aplikace Hammingova okna na signál

5.2 Podpůrná logika firmwaru STM32

Součástí firmwaru je několik podpůrných subsystémů, které fungují jako jakási kostra, která vše drží pohromadě. Blokový diagram této části firmwaru je znázorněn na obrázku 5.6.



Obrázek 5.6: Blokový diagram podpůrné logiky

5.2.1 Časová základna, periodické a odložené úlohy

K časování celého systému je použit hardwarový časovač SysTick, který spouští přerušení s periodou 1 ms. Kromě měření času pro čekací funkce SysTick také obsluhuje frontu plánovaných úloh a tabulku periodických funkcí. Tyto slouží k zpožděnému volání, resp. k provádění nějaké činnosti v pravidelných intervalech. To je využito například pro odrušení zákmitů tlačítek a obsluhu indikačních LED.

5.2.2 Fronta zpráv a úloh

Další součástí systému je fronta zpráv a úloh. Zprávy jsou datové objekty určené k předávání informací z obsluhy přerušení do hlavní smyčky, například pro oznámení konce měření. Fronta úloh slouží především k vykonání dlouhých funkcí poté, co nastane jejich čas ve frontě plánovaných úloh. Kdyby se totiž tyto úlohy vykonávaly přímo v SysTick handleru, nebylo by v nich možné použít čekací funkce, které závisí na pravidelné inkrementaci časové proměnné, a vznikly by tak nekonečné čekací smyčky. Tyto dvě fronty jsou pravidelně obsluhovány v hlavní smyčce programu.

5.2.3 Watchdog

Správnou funkci programu zajišťuje hardwarový watchdog, což je čítač, který po dosažení nuly resetuje procesor. Aby k tomu nedošlo, je třeba čítač pravidelně resetovat, čímž se nastaví na výchozí hodnotu.

Watchdog je resetován v hlavní smyčce vždy po uplynutí určité doby, která se měří pomocí SysTicku. Díky tomu watchdog nepřímo odhalí i „zastavení času“ – situaci kdy dojde k zablokování SysTicku, který zajišťuje inkrementaci časové proměnné.

5.3 Firmware WiFi modulu

5.3.1 Webserver pro uživatelské rozhraní

Firmware WiFi modulu je založen na open source projektu *esp-httpd* [13], který implementuje jednoduchý webserver. Úpravy a vylepšení, které jsem do serveru přidal v průběhu práce, se chystám nabídnout autorovi k začlenění do jeho verze.

Použitá serverová knihovna se stará o dekódování HTTP hlaviček, spojování dělených requestů a sestavování odpovědí. Firmware dále obsahuje jednoduchý souborový systém (pouze pro čtení), ve kterém jsou uloženy HTML stránky a další soubory potřebné pro uživatelské rozhraní.

Některé soubory jsou při požadavku z prohlížeče přímo načteny z paměti a odeslány zpět, jiné se generují dynamicky v callback funkci (např. výsledky měření). Kromě toho server umožňuje kombinaci obou metod – použít soubor jako šablonu a při zobrazení části textu doplnit programově. Příklad takovéto šablony je uveden na obrázku 5.7.

```
<table>
  <tr>
    <th>Firmware</th>
    <td>v%vers_fw%, build <i>%date%</i> at <i>%time%</i></td>
  </tr>
  <tr>
    <th>HTTPD</th>
    <td>v%vers_httpd%</td>
  </tr>
</table>
```

Obrázek 5.7: Ukázka HTML šablony pro server *esp-httpd*

5.3.2 Konfigurace WiFi

Protože přístroj nemá žádný displej, kde by bylo možné zvolit síť WiFi a zadat heslo, musí toto být řešeno prostřednictvím webového rozhraní. Při prvním spuštění modul přejde do AP režimu, kde je možné tuto konfiguraci provést, případně zůstat v AP režimu a provádět měření např. z laptopu. Po připojení k externí síti modul přejde do režimu klienta.

Konfiguraci připojení do sítě je také možné provést pomocí funkce WPS, jak bude vysvětleno v podkapitole 6.4.

5.4 Propojení procesoru s WiFi modulem

WiFi modul je k hlavnímu procesoru připojen jediným UARTem. Obvod ESP8266 podporuje i synchronní režim (USART), ale jen některé moduly mají tyto piny vyvedené na konektor; modul ESP-01, který byl v projektu použit, mezi ně nepatří.

Prostřednictvím tohoto portu probíhá spouštění měření, vyčítání naměřených hodnot, indikace stavu WiFi a další komunikace mezi oběma procesory. Pro spolehlivou funkci spojení je nezbytné navrhnout vhodný komunikační protokol.

5.4.1 Požadavky na komunikační protokol

Pro jednoduchý přenos dat by nebylo třeba vymýšlet nic složitého. Komunikace je ale asynchronní, obousměrná a navíc je použita přenosová rychlost zdaleka přesahující běžné hodnoty. Požadavky na protokol je možné shrnout do následujících bodů:

- **Detekce chyb,** případně jejich oprava. Při vysokých rychlostech mohou vznikat chyby působením externího elektromagnetického rušení, což se často stávalo při vývoji, když byl modul spojen s procesorem dlouhými vodiči. K chybě také může dojít, pokud např. zakolísá napájení některého z procesorů.
- **Řízení toku**, důležité kvůli omezené velikosti přenosových bufferů. Toto by z části řešily signály RTS a CTS, ale je lepší nespoléhat se na možnosti hardwaru, protokol tak bude univerzálnější.
- **Jednoznačné spojení požadavku a odpovědi**, aby nedocházelo ke zmatení komunikace při proplétání více souběžných přenosů. Tím se řeší jeden z hlavních nedostatků originálního firmwaru ESP8266 s AT příkazy.

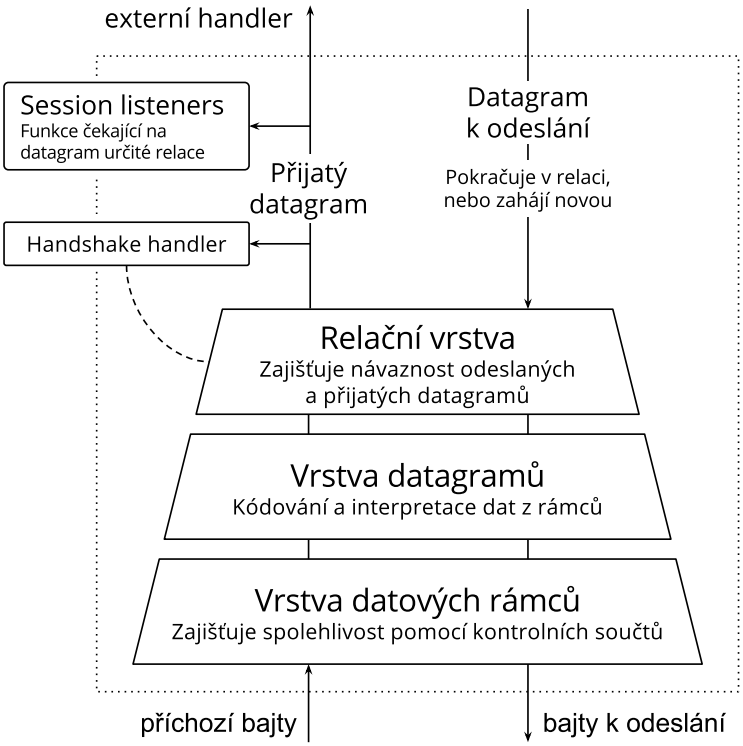
5.4.2 Protokol SBMP

Komunikační protokol jsem pojal jako samostatný projekt, který bude možné použít i v dalších projektech. Je jasně specifikovaný, open-source a přenositelný – referenční implementace funguje beze změn na Atmel AVR, procesorech STM32 s drobnými úpravami i na ESP8266.

Protokol nese jméno SBMP – *Simple Binary Messaging Protocol*. Jeho specifikace je k dispozici na serveru GitHub [14], kopie je také na přiloženém CD. Zde se pokusím krátce shrnout funkce protokolu a jakým způsobem jsou v něm přenášeny bloky dat (výsledky měření).

5.4.3 Vrstvy protokolu SBMP

Protokol je možné rozdělit do tří vrstev, jak je znázorněno na obrázku 5.8. Pro začlenění SBMP knihovny do aplikace stačí implementovat přijímací a vysílací funkce.



Obrázek 5.8: Zjednodušené schéma SBMP driveru

Vrstva rámců se stará o spolehlivost; obsahuje buffer pro příchozí data, ze kterých počítá kontrolní součet (KS). Po přijmutí celého rámce je vypočtený KS porovnán s přijatou hodnotou a pokud se shoduje, jsou data předána vyšší vrstvě. V případě chyby je rámec ignorován. Opakování chybného přenosu je možné realizovat v aplikační vrstvě tak, že vysílající strana čeká na potvrzení a požadavek po určité době bez potvrzení pošle znovu.

Struktura datového rámce je znázorněna v tabulce 5.1. Hlavička (první 4 bajty) má vlastní KS, čímž se do jisté míry eliminuje situace, kdy je přijata chybná délka a driver pak přijde o data z dalších rámců. Na konci rámce je KS bloku dat, dlouhý 4 bajty (pro CRC32), 1 bajt (XOR) nebo 0 bajtů, pokud KS nebyl použit.

Start	Typ KS	Délka dat	XOR hlavičky	Data	KS dat
0x01	1 bajt	2 bajty	1 bajt	...	0–4 bajty

Tabulka 5.1: Struktura datového rámce protokolu SBMP

Datagram, tedy payload rámce, (obr. 5.2) obsahuje číslo relace, typ datagramu a vlastní data. Číslo relace slouží k řetězení odeslaných a přijatých zpráv, specifikuje kontext. S každou

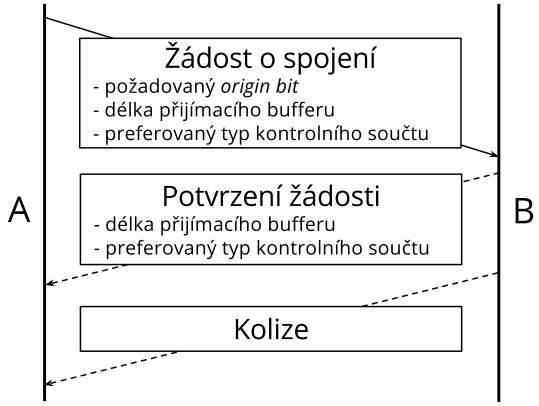
novou relací (sekvence navazujících zpráv) je číslo zvýšeno o jedna. Typ datagramu pak udává, jakou informaci datagram přenáším a jak se s daty má zacházet.

Číslo relace	Typ datagramu	Data
2 bajty	1 bajt	...

Tabulka 5.2: Struktura SBMP datagramu

5.4.4 Navázání komunikace (handshake)

Aby mohla začít komunikace, obě strany (označíme A a B) se musí napřed seznámit se svým protějškem. Průběh této operace je znázorněn na diagramu 5.9. Obě strany si zde vymění informace o svých schopnostech, které budou potřeba pro další komunikaci – velikost bufferu a preferovaný typ kontrolního součtu. Zahajující strana (např. A) si navíc v žádosti zvolí svůj *origin bit* (nejvyšší bit čísla relace).



Obrázek 5.9: SBMP handshake

Pokud je strana B zastižena v klidovém stavu, přijme opačný *origin bit* a žádost potvrdí. Tím je handshake dokončen a může začít normální komunikace, přičemž origin bit zůstává stejný, inkrementace čísla relace je omezena na nižší bity.

Dojde-li ke konfliktu, tj. B čeká na odpověď a A místo toho zahájí vlastní handshake, B pošle A chybový datagram a obě strany operaci přeruší. Po určité době (nejlépe náhodné) je možné pokus o spojení opakovat.

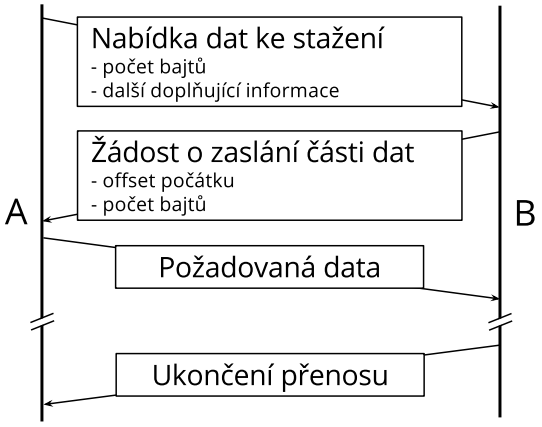
5.4.5 Přenos bloku dat

Protokol SBMP není přímo navržen k přenosu velkých bloků dat, jako je např. pole naměřených hodnot či vypočtený FFT vektor; pro kontrolu chyb je nutné každý rámec napřed uložit do bufferu, a paměť je omezená. Tento problém je možné obejít posíláním dat po částech.

SBMP disponuje metodou pro dělený přenos dat, která je znázorněna na obrázku 5.10. Strana A nabídne straně B blok dat, který si B může číst vlastní rychlostí, dokonce na

přeskáčku. To je výhoda, dojde-li k chybě a rámec se ztratí: pokud B nedostane včas odpověď, může o ztracenou část požádat znovu. Kdyby se blok posílal najednou, musel by se zopakovat celý přenos.

Transakce je ukončena, jakmile A přijme zvláštní ukončovací datagram od B, případně automaticky po uplynutí nějakého časového intervalu. Díky relační vrstvě SBMP může probíhat i více přenosů najednou, případně lze dle potřeby přenos prokládat jinými příkazy.



Obrázek 5.10: Dělený přenos bloku dat

Kapitola 6

Ovládání a funkční režimy přístroje

Jak bylo naznačeno v úvodu a předchozích kapitolách, ovládání přístroje probíhá prostřednictvím webového rozhraní. V této kapitole bude toto rozhraní popsáno a vysvětleno jako reference pro uživatele. Také zde bude krátce představeno webové API, které lze použít k integraci přístroje do externích aplikací.

6.1 Webové uživatelské rozhraní

Rozhraní jsem navrhl tak, aby bylo kompatibilní se všemi moderními prohlížeči a fungovalo i v mobilech a tabletech. To je důležité především pro prvotní konfiguraci, kdy se takto dá nastavit připojení k síti, další měření a obsluha pak může probíhat na PC.

Poznámka: *Snímky obrazovky uvedené v této kapitole mají pozměněné barvy, aby se neplýtvalo inkoustem při tisku. Stránky mají ve skutečnosti tmavé pozadí.*

Uživatelské prostředí je ve formě jednoduché webové aplikace s několika stránkami a postranním menu. Pokud se uživatel připojí k AP pomocí smartphonu, měla by se mu webová aplikace automaticky otevřít. Pokud se tak nestane, je možné v prohlížeči IP adresu (192.168.4.1) zadat ručně.

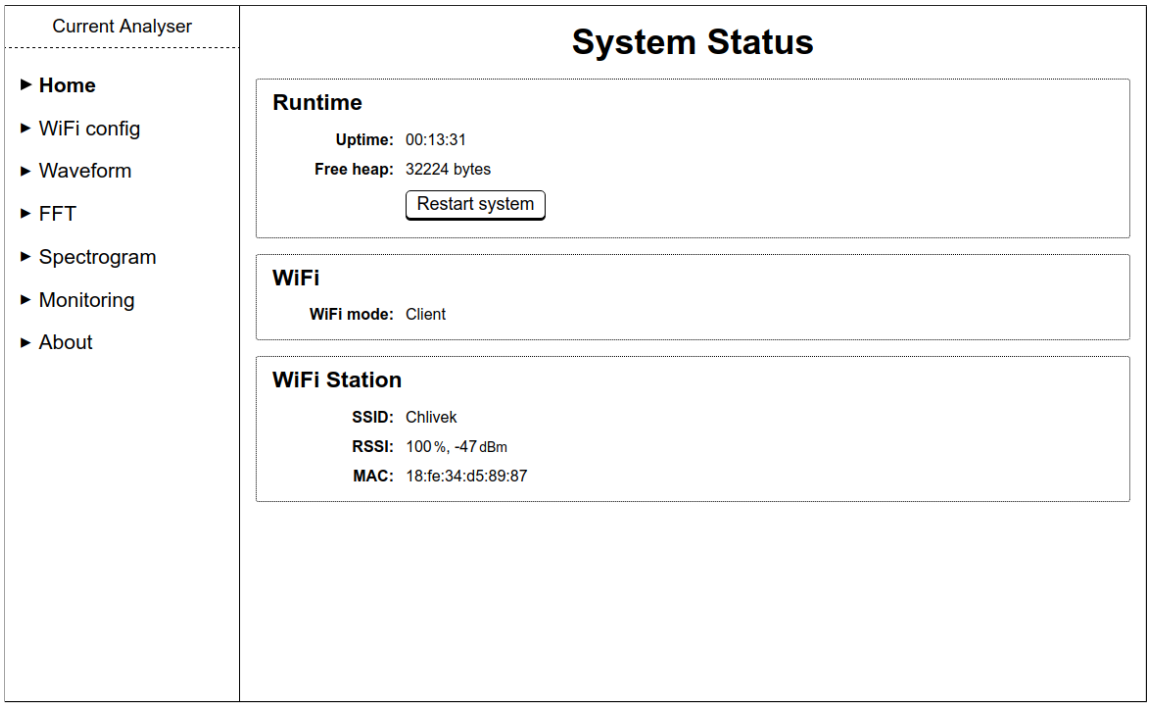
6.1.1 Přehled stavu systému

První stránka uživatelského rozhraní obsahuje přehled informací o stavu systému (obr. 6.1). Informace na stránce se automaticky obnovují každých 10 sekund.

V horní části je uvedena aktuální velikost volné paměti, což má význam především při vývoji firmwaru, ale může sloužit i jako indikátor správné funkce; při běžném provozu by se toto číslo mělo pohybovat kolem 30 kB. Případné problémy, jako je pomalé načítání stránek a chybové hlášky při měření, může napravit reset přístroje, který můžeme provést stiskem příslušného tlačítka na této stránce. Pokud softwarový restart nepomůže, nezbývá než použít hardwarové tlačítko nebo na několik sekund odpojit napájení.

6.1.2 Konfigurace WiFi

Po kliknutí na tlačítko „WiFi config“ v levém menu se zobrazí stránka pro nastavení WiFi. Po několika sekundách by se měly ve spodním boxu objevit dostupné sítě, ke kterým se lze připojit, spolu s informacemi o kvalitě signálu a typu zabezpečení.



Obrázek 6.1: Úvodní stránka uživatelského rozhraní – přehled stavu systému.

Pro připojení do zvolené sítě na ni stačí kliknout. Je-li síť zabezpečena heslem, objeví po kliknutí dialog pro jeho zadání. Prohlížeč je následně přesměrován na stránku s informacemi o průběhu připojování, kde by se po úspěšném připojení měla objevit získaná IP adresa (pokud spojení s prohlížečem stále funguje).

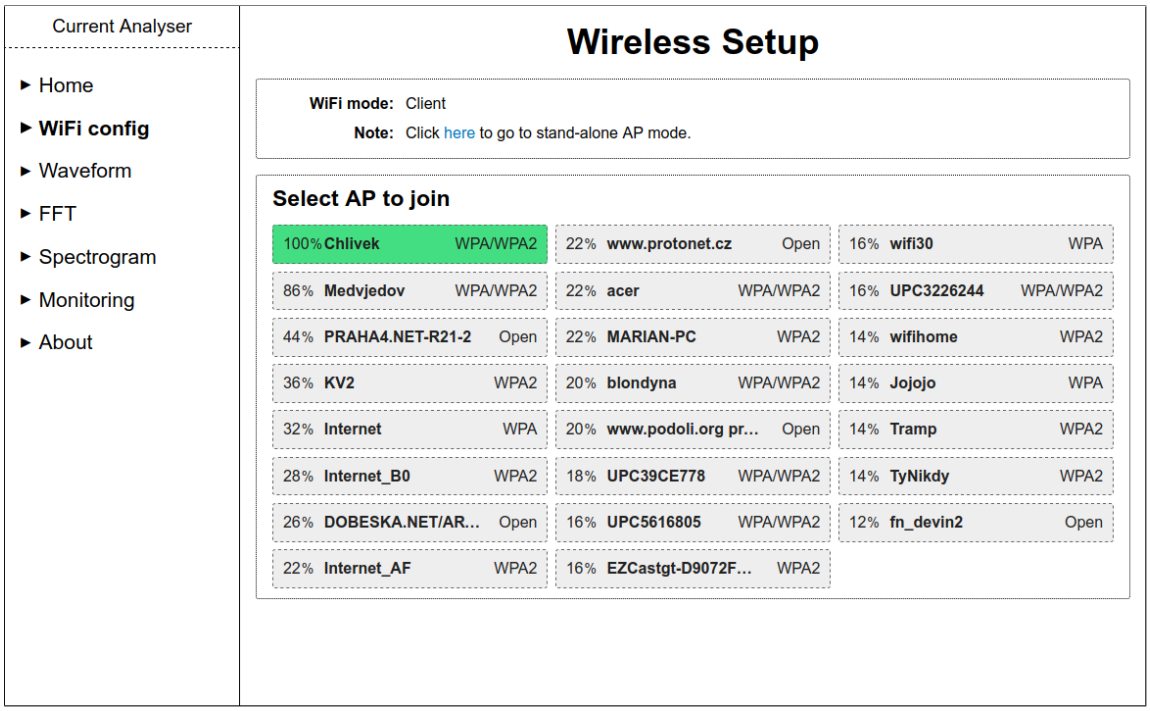
Může se stát, že WiFi modul zůstane nastaven na jednu síť a po přenesení na jiné místo je potřeba nastavení změnit – ovšem webové rozhraní není dostupné. V takovém případě můžeme modul přepnout do AP režimu stiskem modrého tlačítka.

6.1.3 Záznam a zobrazení průběhu proudu

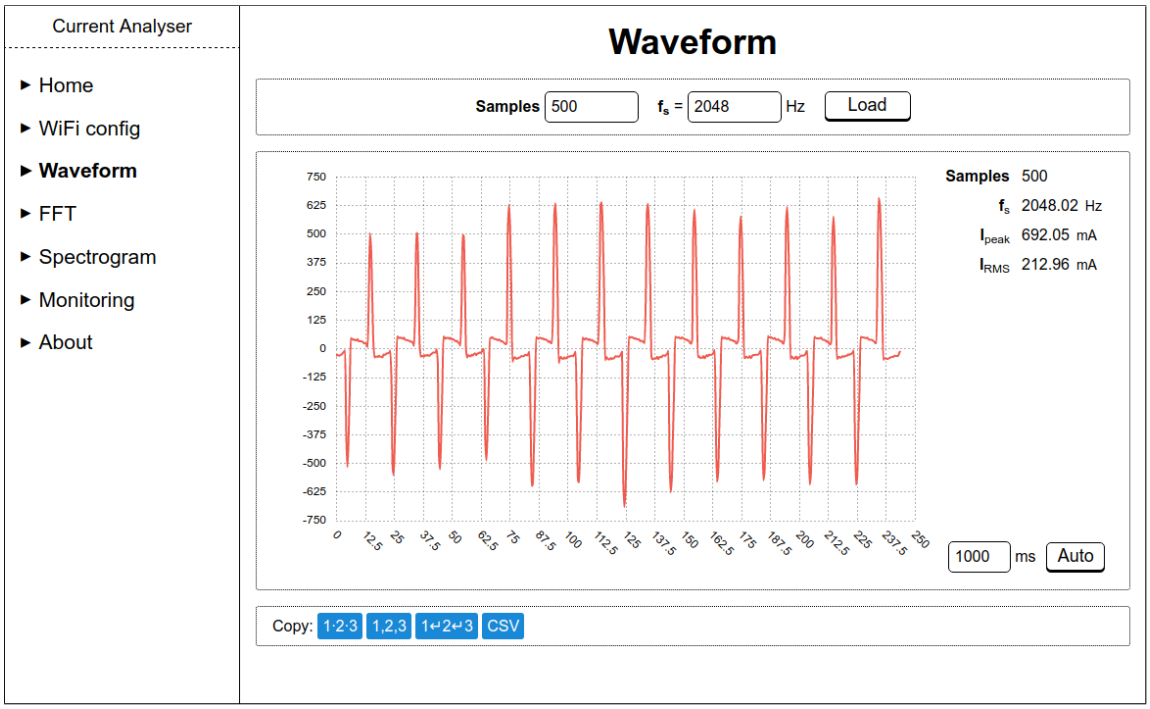
První z měřicích funkcí přístroje je prostý záznam průběhu (obr. 6.3). Zde se můžeme podívat na surový signál, který přístroj měří a ze kterého dále počítá spektrum. Jedná se o vzorky z A/D převodníku, převedené na miliampéry a zarovnané na nulu pomocí střední hodnoty. Při malém počtu vzorků zde můžeme pozorovat chybu této kompenzace, jak je popsáno v podkapitole 4.2.1.

V horní části stránky nastavíme počet vzorků (maximum je 4096) a vzorkovací frekvenci, odměr následně spustíme kliknutím na tlačítko „Load“. Naměřená data lze pomocí tlačítek pod grafem zkopírovat do schránky, odkud je můžeme vložit například do Matlabu nebo Excelu.

Tažením kurzoru myši po grafu při současném stisku levého tlačítka je možné část grafu vybrat ke zvětšení, které zůstane aktivní i po načtení nových dat. Reset do původního zobrazení provedeme stiskem pravého tlačítka myši. (Tuto funkci bohužel nejde použít na mobilu.)



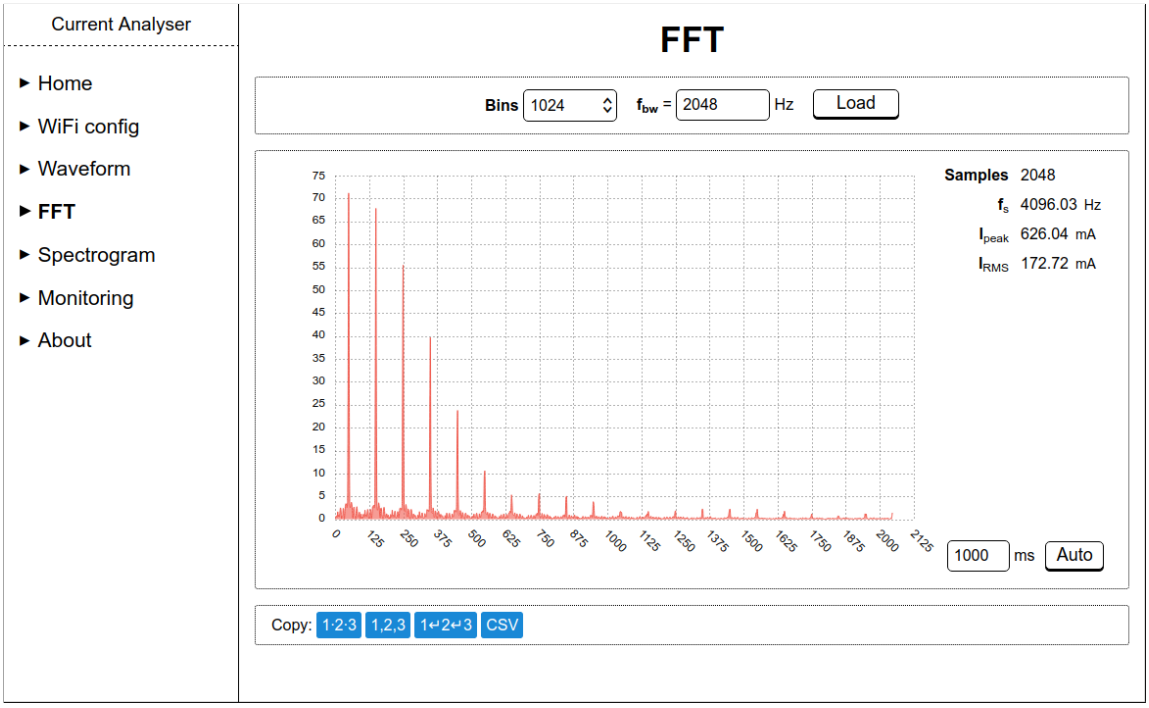
Obrázek 6.2: Volba WiFi režimu a výběr sítě.



Obrázek 6.3: Uživatelské rozhraní pro záznam a zobrazení průběhu proudu

6.1.4 Výpočet a zobrazení spektra proudu

Strínka „FFT“ má podobné ovládání jako funkce zobrazení průběhu, ovšem v tomto případě volíme počet bodů (bins) a šířku pásma (f_{bw}); vzorkovací frekvence bude dvojnásobná, jak vyžaduje Nyquistův teorém. Vypočtené spektrum opět můžeme zkopírovat v různých formátech pomocí tlačítek pod grafem.



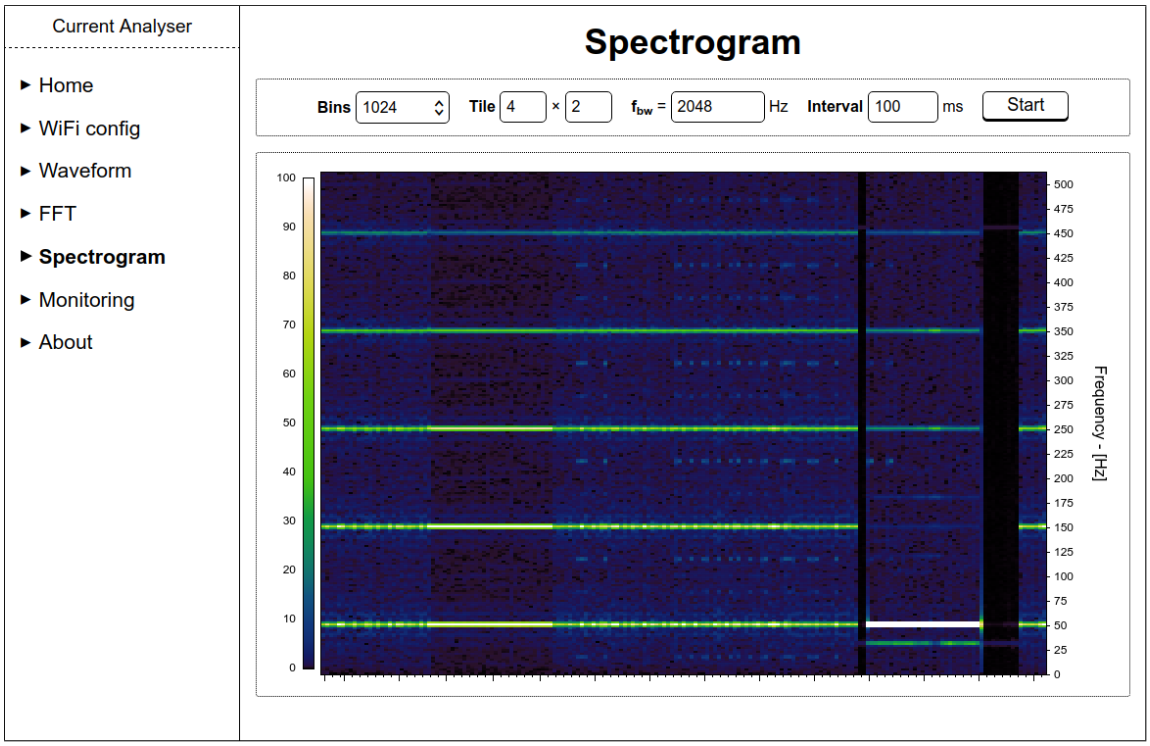
Obrázek 6.4: Zobrazení spektra proudu

Jiný způsob zobrazení spektra nabízí spektrogram, pro který je vyhrazena samostatná stránka (obr. 6.5). Nastavení je obdobné jako u čárového grafu, ovšem přibylo políčko pro zadání obnovovacího intervalu (jak často bude spuštěn odměr a výpočet spektra) a nastavení velikosti „pixelů“ grafu. Jedná se o rozměry obdélníku, který znázorňuje jeden bod spektra. Jejich změnou je možné spektrogram vizuálně zrychlit či zvětšit.

Na rozdíl od jednoduchých grafů zde není možnost zkopírovat data, jedná se především o pomůcku k pozorování dlouhodobějších změn charakteristiky. V případě potřeby ovšem lze celý spektrogram jako obrázek uložit prostřednictvím kontextového menu (pravé tlačítko myši).

6.2 Webové API pro integraci s externími aplikacemi

Komunikace mezi prohlížečem a měřicím přístrojem je realizována pomocí HTTP requestů, podobně jako u běžných webových stránek. Chceme-li například zobrazit graf průběhu, JavaScriptový program v prohlížeči odešle požadavek na server. Obvod ESP8266, ve kterém je server implementován, požádá hlavní procesor o provedení odměru a čeká na odpověď.



Obrázek 6.5: Zobrazení proudového spektrogramu

Když je měření dokončeno, hodnoty jsou v odeslány zpět do prohlížeče a JavaScript je může libovolně zpracovat a zobrazit.

Server nekontroluje odkud request přišel, stačí že je ve správném formátu. Toho můžeme využít ke konstrukci externích aplikací: stačí znát IP adresu přístroje a vědět, jak o měření požádat.

6.2.1 Specifikace API

Prostý časový průběh proudu změříme odesláním HTTP GET requestu na adresu `/measure/raw?n=COUNT&fs=FREQ`, kde `COUNT` a `FREQ` nahradíme počtem vzorků a frekvencí v hertzech (obr. 6.6). Odpovědí je JSON objekt obsahující pole vzorků, vypočtené parametry signálu a `success`. Pokud je hodnota pole `success` rovna `false`, zbytek objektu je neplatný a měření musíme opakovat.

Výpočet spektra probíhá obdobně, v adrese pouze změníme `raw` na `fft`. V tomto případě musí být počet vzorků $n \in \{32, 64, ..., 2048\}$, stejně jako ve webovém rozhraní. Pole `samples` v odpovědi pak bude obsahovat jednotlivé body vypočteného spektra.

Kromě měření proudu a spektra je možné z externí aplikace provádět vše, co je k dispozici ve webovém rozhraní – skenovat WiFi sítě (`/wifi/scan`), resetovat systém (`/system/reset`), testovat zda server běží (`/system/ping`) a podobně. Všechny adresy a formát odpovědí je možné zjistit ze zdrojového kódu firmwaru, nebo ve webovém prohlížeči pomocí DevTools (nástroj pro vývojáře webu, aktivovaný obvykle klávesou F12).

```
$ curl "http://192.168.1.13/measure/raw?n=10&fs=1024"
{
    "samples": [0.665, 1.495, -5.150, 3.157, 20.604, 32.234, ...],
    "stats": {
        "count": 10, "freq": 1024.007,
        "min": -27.581, "max": 32.234,
        "rms": 17.422, "format": "RAW"
    },
    "success": true
}
```

Obrázek 6.6: Ukázka měření proudu z příkazové řádky příkazem `curl`.

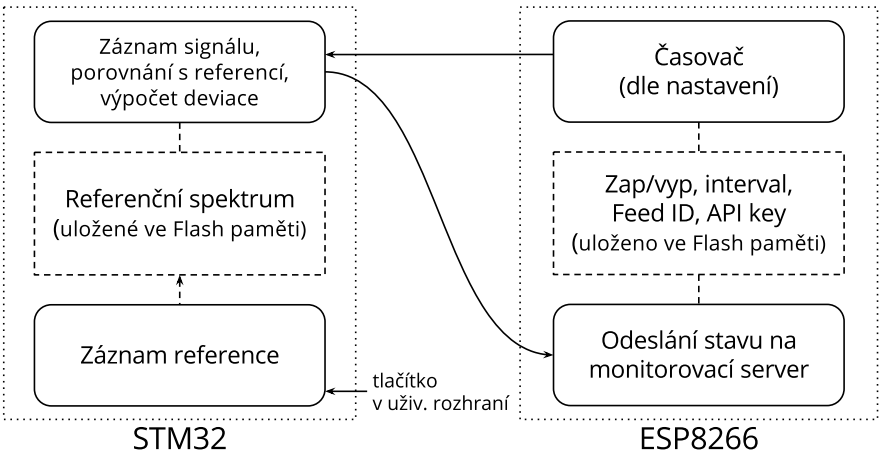
6.3 Režim autonomního monitorování

Přímé měření a zobrazení grafů je užitečné k získání představy o tom, jak charakteristika odběru daného spotřebiče vypadá; pro detekci poruch je ale nezbytné zaznamenávat stav dlouhodobě a mít možnost vyhodnotit změny například za dobu několika dní.

Nabízelo by se ukládat data na paměťové médium, jako je SD karta. Protože ale máme k dispozici připojení k internetu, je výhodné záznamy odesílat na externí server, který se postará o jejich uložení do databáze a další vyhodnocení.

V autonomním režimu tedy přístroj dle nastavení pravidelně měří charakteristiku proudu, porovnává průběh s uloženou referencí a odchylku spolu s dalšími parametry signálu odesílá na monitorovací server. Postup výpočtu odchylky od reference je popsán v podkapitole 3.3.

Na autonomní funkci se podílejí oba procesory (STM32 a ESP8266), jak je patrné z diagramu 6.7. Nastavení a referenční spektrum si firmware uloží do paměti Flash, v případě výpadku napájení o ně tedy nepřijdeme a monitorování může nerušeně pokračovat.



Obrázek 6.7: Blokové schéma funkce autonomního režimu

6.3.1 Komunikace s monitorovacími servery Xively a ThingSpeak

Ve firmwaru obvodu ESP8266 je implementována komunikaci se dvěma populárními IoT platformami: *Xively* [15] a *ThingSpeak* [16]. Jsou to cloudové služby, které ukládají změřené hodnoty do databáze a v nejjednodušším případě z nich vykreslují graf (obr. 6.8). ThingSpeak je navíc integrován s Matlabem, což se může hodit pro další analýzu.

Obě zmíněné platformy fungují podobně a služba je v obou případech zdarma (s určitými omezeními). Ve webovém rozhraní zvolené platformy si uživatel vygeneruje autentizační klíč a založí datový kanál s unikátním identifikačním číslem. Ukládání naměřených hodnot pak probíhá jednoduchým HTTP requestem. V případě platformy Xively se jedná o PUT request s daty ve formátu CSV, XML nebo JSON.

```
PUT /v2/feeds/FEED_ID.csv HTTP/1.1
Host: api.xively.com
X-ApiKey: API_KEY
Content-Type: text/csv

Name1,Value1
Name2,Value2
...
```

K odeslání hodnot na server ThingSpeak je možné použít POST request, který není o moc složitější. V tomto případě se jako ID použije klíč *Write API Key*, který si zkopírujeme z webového rozhraní.

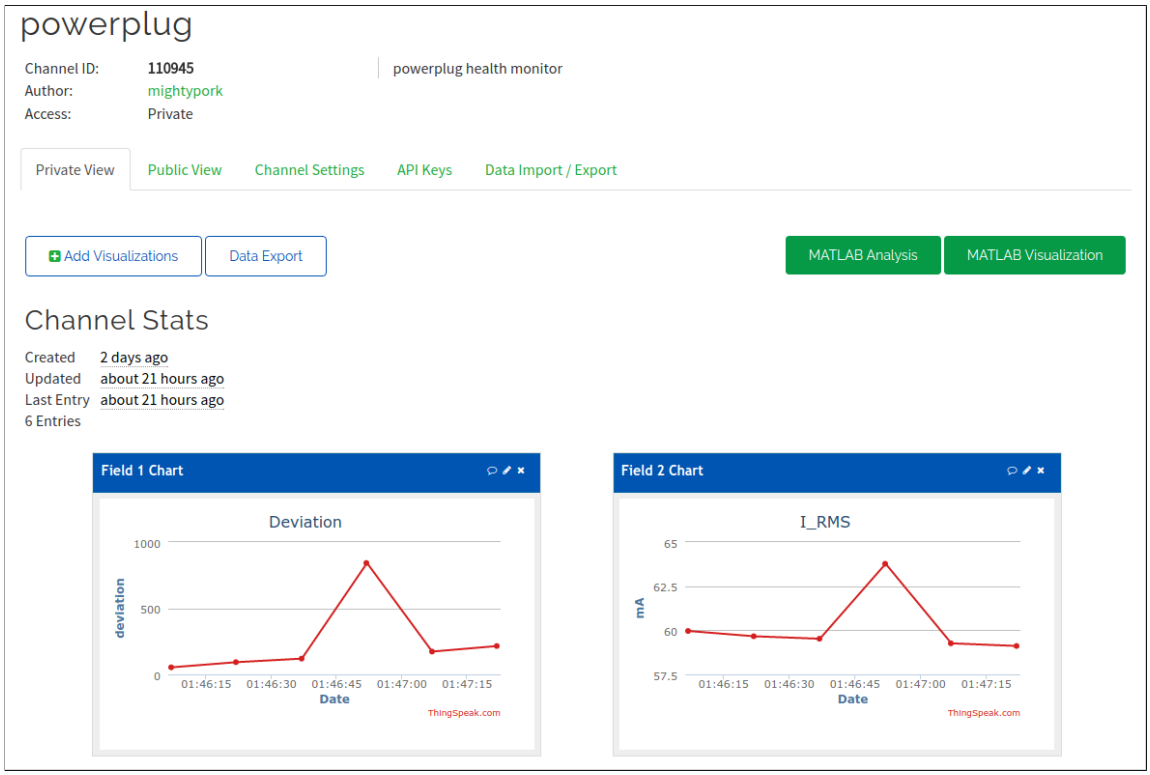
```
POST /update HTTP/1.1
Host: api.thingspeak.com
Content-Type: application/x-www-form-urlencoded

key=WRITE_API_KEY&field1=value1&field2=value2...
```

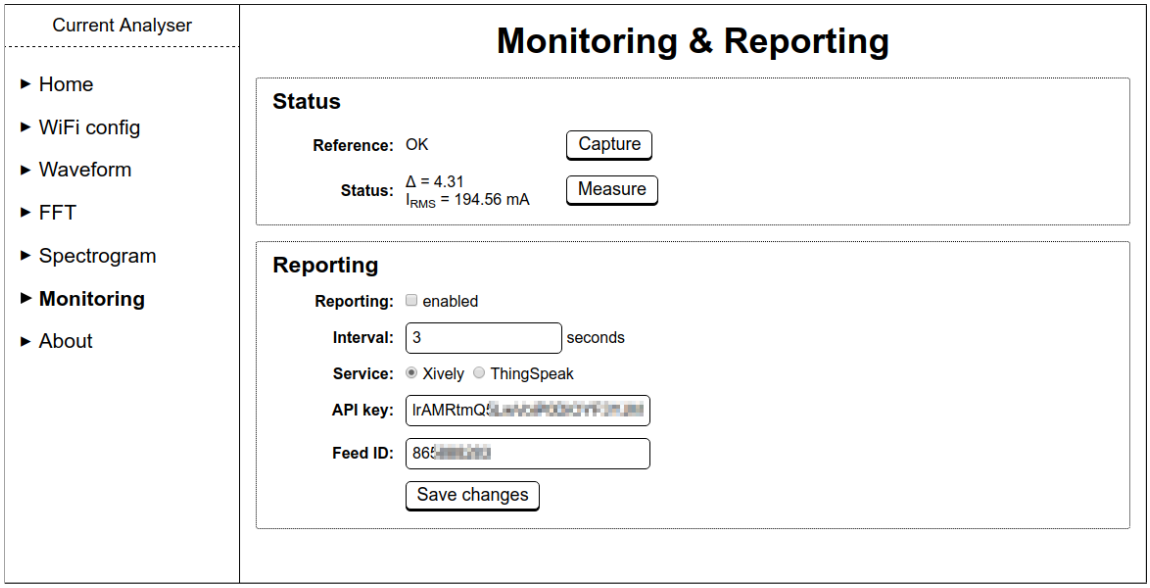
6.3.2 Nastavení autonomního režimu

Konfigurace autonomního režimu probíhá na stránce „Monitoring“ uživatelského rozhraní měřicího přístroje (obr. 6.9).

Záznam a uložení reference provedeme stiskem tlačítka „Capture“, aktuální charakteristiku odběru můžeme následně porovnat s referencí pomocí tlačítka „Measure“. Ve spodní části stránky se nachází formulář pro nastavení monitorovacích serverů a intervalu měření.



Obrázek 6.8: Webové rozhraní služby ThingSpeak se zobrazením grafů.



Obrázek 6.9: Stránka s nastavením autonomního režimu.

6.4 Hardwarové uživatelské rozhraní

Přední panel přístroje je vybaven třemi tlačítky – „Reset“, „WPS“ a „AP“. Ve spodní části se pak nacházejí dvě barevné indikační LED.

Obrázek 6.10: Vnější podoba měřicího přístroje; v dolní části panelu dvě barevné indikační LED

6.4.1 Ovládací tlačítka

Funkce červeného tlačítka „Reset“ je zjevná, restartuje se jím hlavní procesor spolu s WiFi modulem. Nastane-li situace, kdy je potřeba zařízení restartovat, je tak k dispozici kromě tlačítka ve webovém rozhraní ještě hardwarový reset. Taková situace by samozřejmě neměla nastat, případné selhání programu by měl odhalit watchdog a provést reset automaticky.

Modré tlačítko „AP“ zapne AP režim, pokud je modul v režimu klienta. Toto tlačítko slouží k obnovení přístupu do nastavení, pokud například přístroj přeneseme na jiné místo a potřebujeme přenastavit připojení do sítě (modul se jinak trvale snaží připojit k uloženému SSID).

Žluté tlačítko s nápisem „WPS“ slouží ke spuštění funkce *Wifi Protected Setup*. Stiskneme-li toto tlačítko, WiFi modul přejde do režimu WPS a čeká na stisknutí potvrzovacího tlačítka na WiFi modemu. Inicializace režimu WPS je indikována oranžovou barvou indikační LED, která se změní v jasně žlutou, jakmile je inicializace hotová. Poté, co uživatel stiskl potvrzovací tlačítko na WiFi modemu přístroj obdrží přístupové heslo a může navázat spojení. Režim WPS je také možné opustit manuálně opakovaným stiskem žlutého tlačítka.

6.4.2 Barevná indikace stavu

Přístroj není vybaven displejem, místo toho jsou na předním panelu dvě indikační LED – pravá LED indikuje stav WiFi, levá pak průběh měření. Sledováním barevných signálů

můžeme snadno zjistit, zda je aktivní režim AP, je-li přístroj připojen do sítě a zda probíhá měření.

Určitou nevýhodou oproti displeji je, že pomocí barev nejde jednoduše vypsat IP adresa (získaná v režimu klienta). Musíme ji tedy zjistit v uživatelském rozhraní WiFi modemu, nebo použít vyhledávací program (na Linuxu příkaz `nmap`, na telefonech se systémem Android např. aplikace „Fing“ [17]).

Barevné signály jsou následující:

- Po spuštění přístroje LED třikrát bíle blikne (indikace startu), následuje barevný kód upřesňující důvod resetu. Zelená barva značí normální spuštění nebo reset hardwarovým tlačítkem, červená indikuje chybu (např. watchdog reset).
- V případě výpadku komunikace s WiFi modulem LED dlouze červeně bliká. Tento signál je také možné pozorovat krátce po spuštění, než je s modulem navázáno spojení.
- Stav WiFi indikují krátké barevné pulzy: modrá označuje AP, červená/zelená značí režim klienta (zelená = spojení navázáno). V režimu klient+AP je stav indikován dvojicí barevných pulzů, každý pro jeden režim.
- Funkce WPS je také indikována barevně: Po stisku tlačítka se rozsvítí oranžová barva značící přípravu na WPS. Jakmile je WPS aktivní, barva se změní na jasně žlutou a zůstane svítit, dokud není spojení navázáno nebo nedojde k chybě. Poté, co WiFi modul získal heslo do sítě, musí se ještě připojit, proto může několik sekund po skončení WPS blikat červený signál, který se změní na zelený po úspěšném připojení.
- Měření je signalizováno fialovou barvou levé LED, která má v klidu barvu modrou (indikace napájení). Svítí-li fialové světlo, probíhá záznam.

Kapitola 7

Ověření funkce měřením reálných spotřebičů

V této kapitole budou uvedeny ukázky měření reálných síťových spotřebičů. Cílem práce byla především konstrukce realizace přístroje, proto mají tato měření spíše ilustrační charakter.

Poruchy, které má přístroj detekovat, budou simulovány přibrzděním motoru, což představuje například zadření ložisek. Budou zde také demonstrovány změny v proudové charakteristice při různých pracovních režimech (volba rychlostního stupně).

7.1 Síťový ventilátor SUNON

Prvním z měřených přístrojů byl malý síťový ventilátor značky SUNON, typ DP203A. Rotor lze snadno přibrzdit i zastavit rukou a sledovat tak změny průběhu a spektra. Ventilátor bohužel nejde rozebrat, ale pravděpodobně se jedná o jednoduchý indukční motor.

Na obrázcích 7.1 a 7.2 je zachycen klidový průběh proudu a jeho spektrum. Přibrzdění nebo i úplné zastavení motoru překvapivě nemá na průběh velký vliv, pouze nepatrně stoupne magnituda harmonických složek (50 Hz, 100 Hz, ...). Tato změna se ovšem výrazně projeví v monitorovacím režimu, detekce poruchy by tedy tímto způsobem měla být možná.

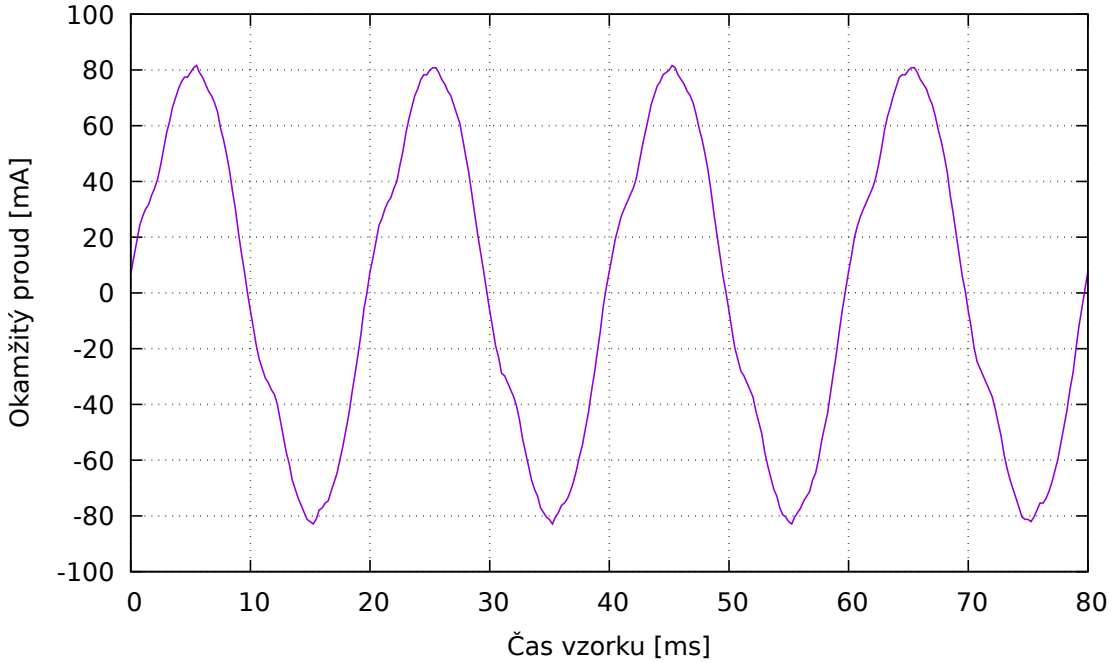
7.2 Pokojový ventilátor StarFan

Další ze zkušebních spotřebičů byl velký pokojový ventilátor značky StarFan. U tohoto ventilátoru se podařilo zachytit přechodový děj při spuštění (obr. 7.3). Proudová špička je typickým jevem při spouštění motorů a některých napájecích zdrojů.

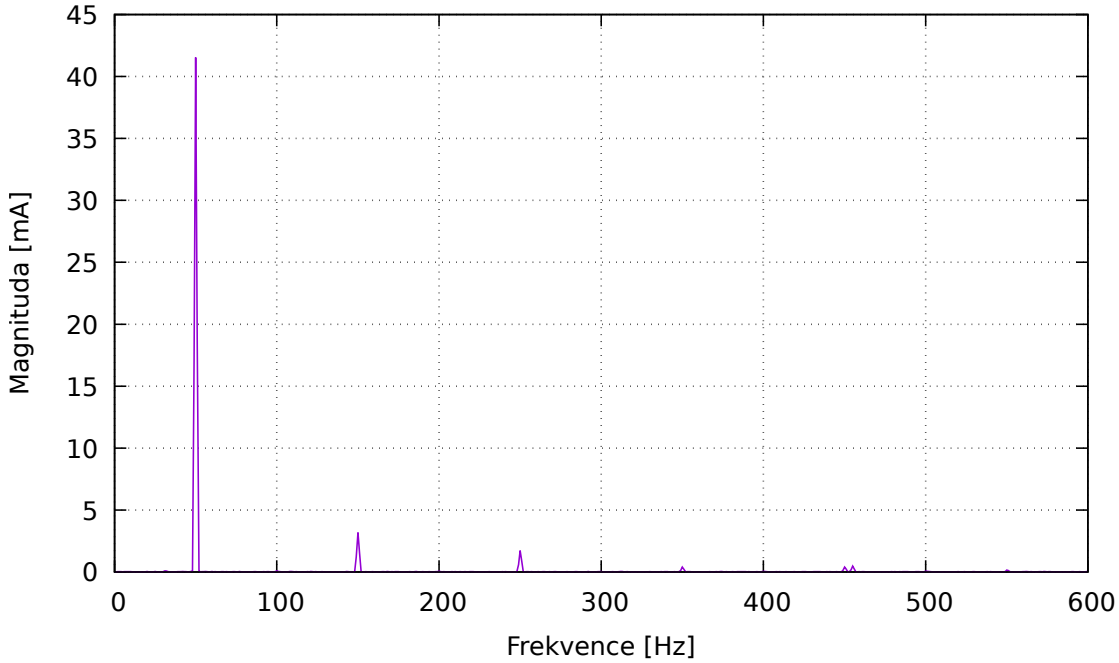
Ventilátor kvůli ochranné kleci nejde přibrzdit, ale má 3 volitelné rychlosti, které můžeme porovnat. Podíváme-li se na spektrum, pozorujeme, že při vyšších rychlostech roste složka 50 Hz na úkor ostatních. Ukázka změny časového průběhu proudu je zachycena na obrázku 3.6. Spektrum při nejnižší rychlosti znázorňuje graf 7.4.

7.3 Sériový motor z mixéru

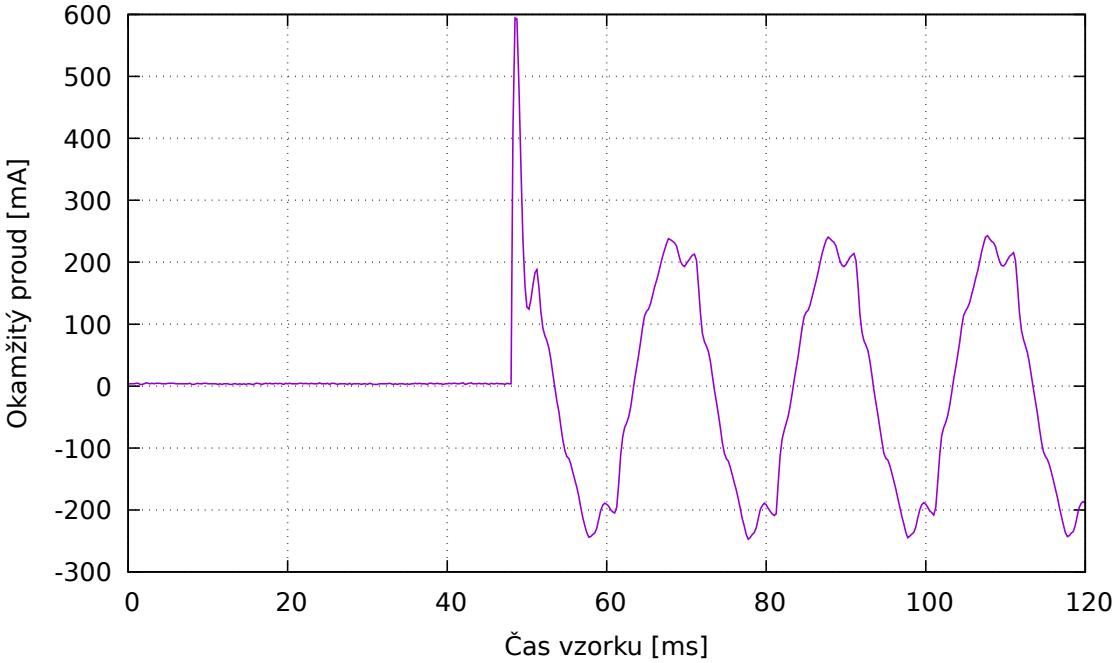
Předchozí dva ventilátory při zatížení nevykazují výrazné změny spektra, pouze se mění magnituda složek. To ovšem neplatí pro všechny typy motorů; ukázkou budiž univerzální sériový motor zapůjčený z katedry pohonů a trakce, který je dalším testovacím subjektem.



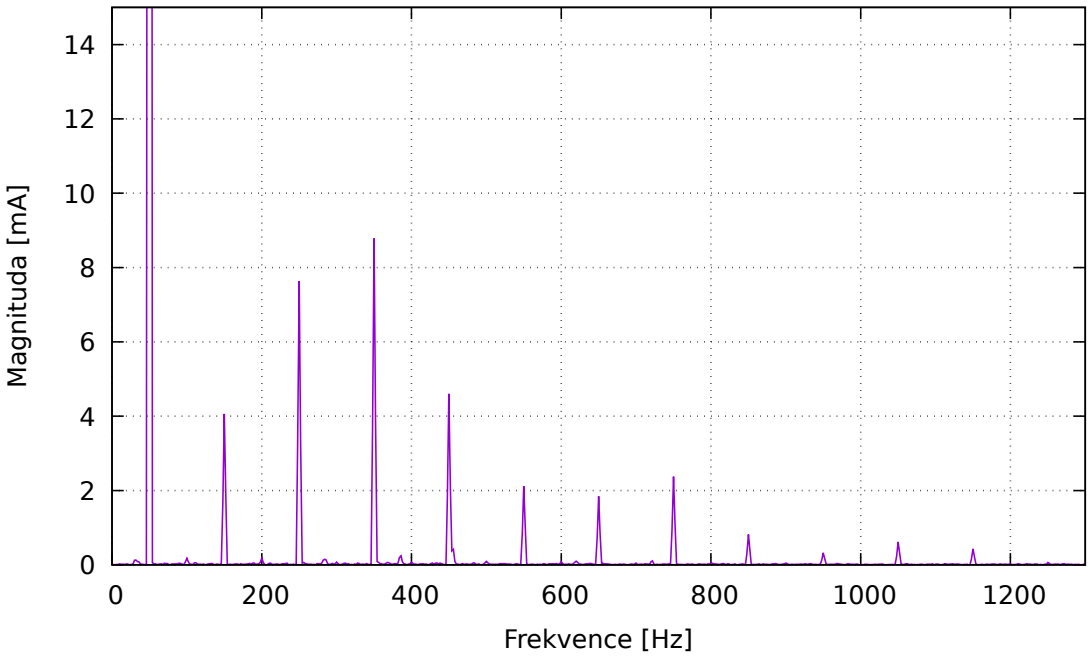
Obrázek 7.1: Časový průběh proudového odběru ventilátoru SUNON; $f_s = 4\,\text{kHz}$, $I_{\text{RMS}} = 55.43\,\text{mA}$.



Obrázek 7.2: Spektrum proudového odběru ventilátoru SUNON; $f_s = 2\,\text{kHz}$.



Obrázek 7.3: Přechodný děj po spuštění ventilátoru StarFan; $f_s = 4\,\mathrm{kHz}$



Obrázek 7.4: Spektrum odběru ventilátoru StarFan. $f_s = 4\,\mathrm{kHz}$ (přiblíženo, složka 50 Hz dosahuje magnitudy 125 mA)

7.3.1 Vliv diodového omezení rychlosti na průběh proudu

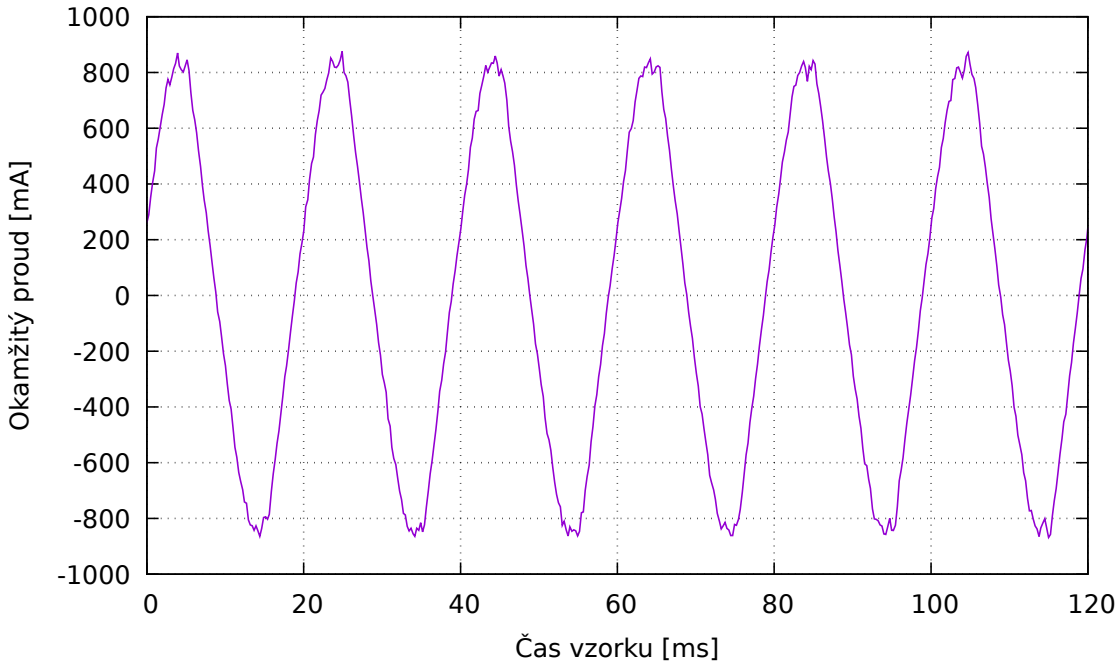
Motor je vybaven dvoustupňovým voličem rychlosti. Snížení rychlosti je dosaženo zapojením diody 1N5408 do série s motorem. Vliv diody můžeme pozorovat v grafu časového průběhu jako potlačení jedné půlperiody (obr. 7.5, resp. 7.6). Zde se nepříjemně projevuje chyba způsobená kalibrací nuly pomocí střední hodnoty (viz podkapitola 4.2.1), v tomto případě by byla lepší kalibrace ruční, nebo pomocí multiplexeru.

Zapojení omezovací diody se samozřejmě projeví také ve spektru, kde přibudou sudé harmonické (100 Hz, 200 Hz atd.) a magnituda lichých harmonických složek bude přibližně poloviční. Změny spektra zachycuje obrázek 7.7.

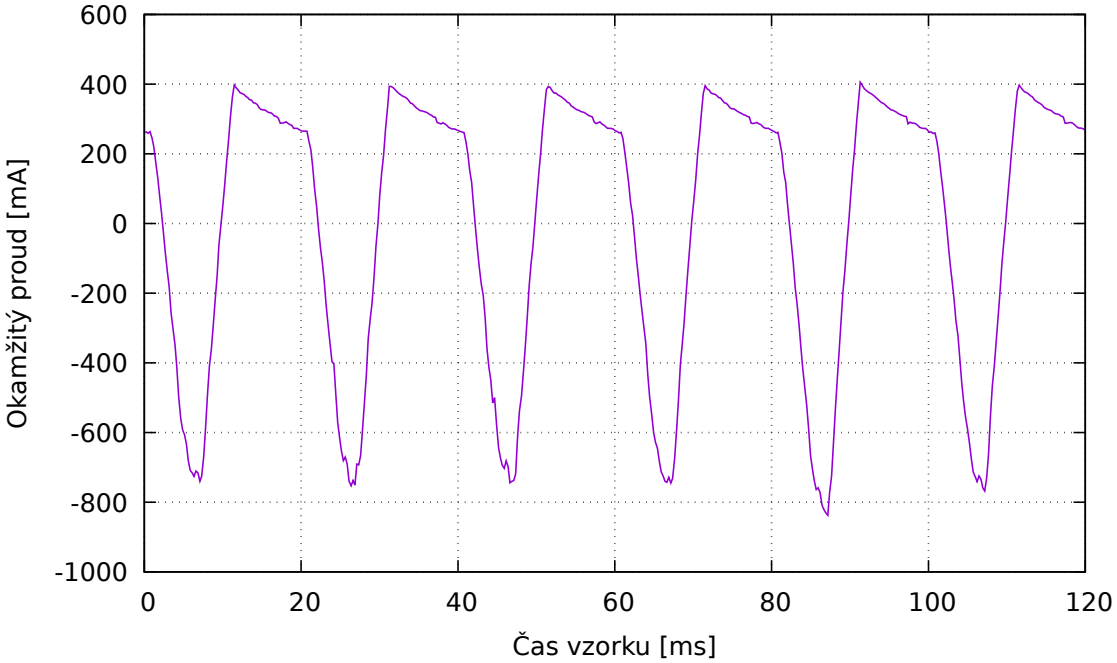
7.3.2 Složky spektra sériového motoru ve vyšších frekvencích

Spektrum tohoto motoru je ještě zajímavější, podíváme-li se do vyšších frekvencí (obr. 7.8). Na rozdíl od předchozích měření ventilátorů, jejichž spektrum prakticky končí kolem 1 kHz, zde můžeme sledovat shluky nenulových složek o frekvenci kolem 5 kHz a 10 kHz (v případě poloviční rychlosti pak 4 kHz a 8 kHz).

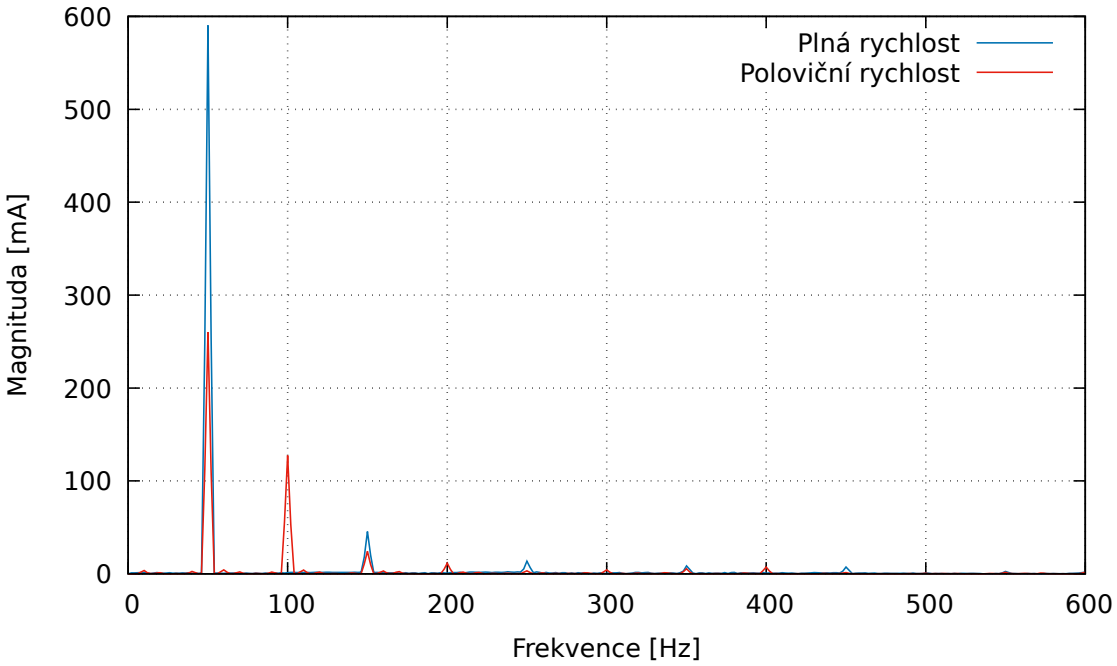
Tyto shluky se při rozběhu posouvají do vyšších frekvencí, naopak přibrzdění způsobí pohyb opačný. Je tedy zřejmé, že se jejich frekvence odvíjí od rychlosti motoru, což by mohlo být užitečné pro nějaký druh diagnostiky. Můžeme se domnívat, že tyto složky spektra jsou způsobeny třením mezi kartáči a rotorem; předchozí zkoumané motory byly totiž bezkartáčové a tento jev se u nich neprojevil.



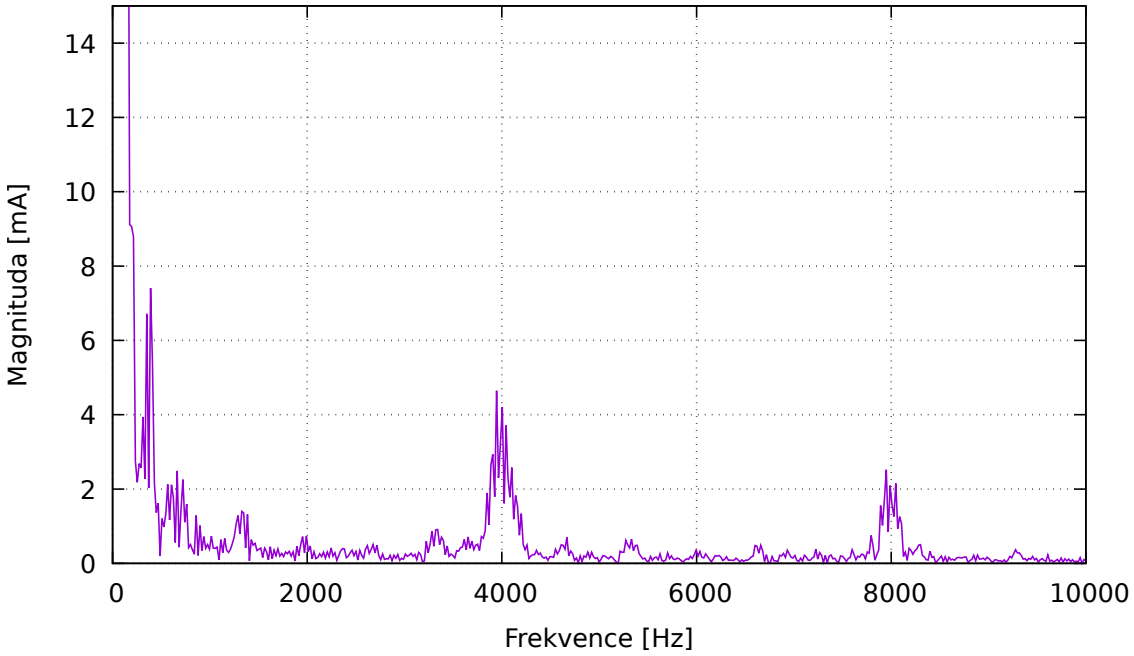
Obrázek 7.5: Odběr sériového motoru, plná rychlost



Obrázek 7.6: Odběr sériového motoru s omezením rychlosti



Obrázek 7.7: Změna spektra proudového odběru sériového motoru při přepnutí z poloviční na plnou rychlost



Obrázek 7.8: Shluky nenulových složek spektra odběru sériového motoru ve vyšších frekvencích (přiblíženo)

Kapitola 8

Závěr

V rámci této bakalářské práce byl úspěšně realizován měřicí přístroj pro diagnostiku síťových spotřebičů pomocí analýzy charakteristik odebíraného proudu. Systém byl vestavěn do zásuvkového adaptéru a disponuje intuitivním ovládáním pomocí tlačítek, indikačních LED a webového rozhraní, které je přístupné prostřednictvím sítě WiFi.

Uživatelské rozhraní umožňuje zobrazit časový průběh proudu spotřebičem, vypočítat spektrum pomocí FFT a zobrazit spektrogram. Přístroj je dále vybaven autonomním režimem, ve kterém je stav spotřebiče pravidelně porovnáván s referencí a výsledek je odesílán na monitorovací servery Xively a ThingSpeak.

Přístroj je schopný měřit proud do velikosti 1.7 A, a tedy monitorovat spotřebiče o maximálním příkonu přibližně 390 W. Pro měření byl použit 12-bitový A/D převodník, rozlišovací schopnost je tedy 0.83 mA. Drobnou úpravou zapojení by bylo možné zvýšit citlivost, ovšem na úkor vstupního rozsahu.

Funkce přístroje byla ověřena měřením reálných spotřebičů, a experimenty ukázaly, že některé (simulované) poruchy je skutečně možné takto detekovat. Kromě detekce poruch je též možné využít autonomního režimu k monitorování domácí automatizace, časových spínačů a podobně.

Přístroj dokáže měřit pouze proud, užitečným rozšířením by bylo přidat možnost měřit napětí, což by umožnilo výpočet účiníku a příkonu spotřebičů a otevřelo tak cestu k dalším diagnostickým metodám. Přístroj by dále bylo možné rozšířit o schopnost ukládat záznamy na SD kartu pro dlouhodobé monitorování bez nutnosti připojení k internetu.

Příloha A

Bibliografie

[1] H. D. Haynes. „Electrical Signature Analysis (ESA) Developments at the Oak Ridge Diagnostics“. Angl. In: *Proceedings of the 8th International Congress on Condition Monitoring and Diagnostic Engineering Management*. Sv. 2. 1995, s. 511–518.

[2] Sukhjeet Singh, Amit Kumar a Navin Kumar. „Motor Current Signature Analysis for Bearing Fault Detection in Mechanical Systems“. Angl. In: *3rd International Conference on Materials Processing and Characterisation (ICMPC 2014)*. Ed. Swadesh Kumar Singh. Sv. 6. 2014, s. 171–177.

[3] Mohamed Rgeai et al. „Gearbox Fault Detection Using Spectrum Analysis of the Drive Motor Current Signal“. Angl. In: *Proceedings of the 4th World Congress on Engineering Asset Management*. (Athens, Greece). Ed. Dimitris Kiritsis et al. Springer-Verlag London Ltd, zář. 2009, s. 758–768. ISBN: 9781849960021.

[4] Michael Doogue a Monica Thomas. „Integrating Hall-Effect Magnetic Sensing Technology into Modern Household Appliances“. In: (2013). URL: http://www.allegromicro.com/en/Design-Center/Technical-Documents/Hall-Effect-Sensor-IC-Publications/Integrating-Hall-Effect-Magnetic-Sensing-Technology-Into-Modern-Household-Appliances.aspx (cit. 29. 04. 2016).

[5] *Hausdorff distance, Wikipedia*. Angl. URL: https://en.wikipedia.org/wiki/Hausdorff_distance (cit. 25. 04. 2016).

[6] Espressif. *ESP8266 Community Forum*. Angl. URL: http://www.esp8266.com/ (cit. 15. 04. 2016).

[7] Espressif. *ESP8266 Community Wiki*. Angl. URL: http://www.esp8266.com/wiki/doku.php (cit. 15. 04. 2016).

[8] Charles Lohr a Brian Benchoff. *ESP8266 Distance Testing*. Angl. Zář. 2014. URL: http://hackaday.com/2014/09/26/esp8266-distance-testing/ (cit. 09. 04. 2016).

[9] *SDKs – ESP8266 Developer Zone*. Angl. URL: http://bbs.espressif.com/viewforum.php?f=46 (cit. 02. 05. 2016).

[10] Paul Sokolovsky. *esp-open-sdk*. Angl. URL: https://github.com/pfalcon/esp-open-sdk (cit. 15. 04. 2016).

[11] Espressif. *ESP8266 Flash Download Tool*. Angl. URL: http://bbs.espressif.com/viewtopic.php?t=25 (cit. 15. 04. 2016).

[12] Fredrik Ahlberg. *esptool.py*. Angl. URL: https://github.com/themadinventor/esptool (cit. 15. 04. 2016).

[13] Jeroen Domburg. *esp-httpd*. Angl. URL: https://github.com/Spritetm/esphttpd (cit. 26. 04. 2016).

[14] Ondřej Hruška. *Simple Binary Messaging Protocol*. Angl. URL: https://github.com/MightyPork/sbmp (cit. 19. 04. 2016).

[15] *Xively.com*. Angl. URL: https://xively.com/ (cit. 01. 05. 2016).

[16] *ThingSpeak.com*. Angl. URL: https://thingspeak.com/ (cit. 01. 05. 2016).

[17] *Mobilní aplikace Fing*. Angl. URL: https://play.google.com/store/apps/details?id=com.overlook.android.fing (cit. 08. 05. 2016).

Příloha B

Obsah přiloženého CD

`device-proto`	**Firmware a hardware zkušebního prototypu**
`fw-esp`	Firmware ESP8266
`fw-stm32`	Firmware STM32
`pcb`	Schéma a výrobní podklady desky plošných spojů
`device-final`	**Firmware a hardware finální verze přístroje**
`fw-esp`	Firmware ESP8266
`fw-stm32`	Firmware STM32
`pcb`	Schéma a výrobní podklady desky plošných spojů
`rev23b`	- Revize 23b - vyrobená verze desky
`rev24`	- Revize 24 - verze s opravami chyb
`bibliography`	**Použitá literatura**
`datasheets`	Datasheety použitých součástek
`papers`	Odkazované vědecké články (volně šiřitelné)
`figures`	**Ilustrace použité v práci**
`diagrams`	Blokové diagramy
`photo`	Fotografie
`plots`	Grafy
`schema`	Schémata zapojení
`screnshots`	Snímky obrazovky (uživatelské rozhraní)
`README.txt`	Doprovodná informace k CD
`BP_Hruska_2016.pdf`	Úplný text bakalářské práce