



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ

Katedra Elektroenergetiky

Synchrofázory v elektrizačních soustavách

Synchrophasors in power networks

diplomová práce

Studijní program: Elektrotechnika, energetika a management
Studijní obor: Elektroenergetika

Vedoucí práce: doc.Dr. Jan Kyncl

Jiří Humlíček

Praha 2014

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra elektroenergetiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Jiří Humlíček**

Studijní program: Elektrotechnika, energetika a management
Obor: Elektroenergetika

Název tématu: **Synchrofázory v elektrizačních soustavách**

Pokyny pro vypracování:

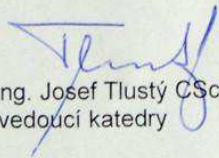
1. Seznamte se s problematikou synchrofázorů v elektrizačních soustavách
2. Vytvořte programové vybavení implementující zvolenou metodu pro vhodný mikroprocesor
3. Pomocí simulovaných signálů ověřte funkci zařízení

Seznam odborné literatury:

- [1] C37.118.1-2011 - IEEE Standard for Synchrophasor Measurements for Power Systems
- [2] C37.118.2-2011 - IEEE Standard for Synchrophasor Data Transfer for Power Systems
- [3] Kyncl, Jan, Adithya Hariram, and Martin Novotny. "On measurement of synchronous phasors in electrical grids." Circuits and Systems (ISCAS), 2013 IEEE International Symposium on. IEEE, 2013.

Vedoucí: doc.Dr. Jan Kyncl

Platnost zadání: do konce letního semestru 2014/2015


prof. Ing. Josef Tlustý CSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 6. 2. 2014

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, atd.) uvedené v použitých zdrojích. Nemám závažný důvod proti použití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb. , o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne _____

podpis

Poděkování

Rád bych poděkoval vedoucímu mé diplomové práce doc.Dr. Janu Kynclovi za jeho motivující přístup k řešení této práce, cenné rady a čas který mi věnoval. Velký dík patří také mé rodině za podporu během celého dosavadního studia.

Abstrakt

Tato práce se zabývá využitím synchronního měření v elektrické síti. Je zde popsána metoda získání fázoru z měřeného signálu s využitím statistických metod. Algoritmus implementující tuto metodu je vytvořen v jazyce „C“. Program je upraven pro jednoduchý mikrokontrolér řady ATMega s možností měřit dva kanály (např. napětí a proud). Funkčnost algoritmu je otestována simulačním signálem. Výsledky měření jsou posílány po sériové lince do zobrazovací jednotky, kde jsou interpretovány na LCD displeji.

Abstract

This thesis deals with the use of synchronous measurements in the power network. There is described method of getting phasor the measured signal using statistical methods. The algorithm implementing this method is created in language "C". The program is modified to simple microcontroller ATMega series with the possibility to measure two channels (eg. voltage and current). The functionality of the algorithm is tested by simulation signal. The measurement results are sent over the serial line to the display unit, which are interpreted on the LCD.

Klíčová slova

fázor, měření frekvence, synchronní měření, PMU jednotka, využití synchronního měření, systém WAMS, střední hodnota, rozptyl, mikrokontrolér, AVR Studio 5, sériová linka, LCD displej

Keywords

phasor, frequency measurement, synchronous measurement, PMU unit, synchronous measurement use, WAMS system, mean value, variance, microcontroller, AVR Studio 5, serial interface, LCD display

Obsah

Obsah	- 8 -
1 Úvod.....	- 10 -
2 Synchronní měření	- 11 -
2.1 Definice fázoru.....	- 11 -
2.2 Definice synchronního fázoru.....	- 11 -
2.3 Jednotky pro měření synchronních fázorů.....	- 14 -
2.4 Využití synchronního měření v síti.....	- 14 -
2.4.1 Estimace ustáleného stavu elektrických sítí.....	- 15 -
2.4.2 Monitorování stability sítě	- 17 -
2.4.3 Měření parametrů vedení za chodu.....	- 18 -
2.4.4 Podpora bezpečného spínání v distribučních sítích	- 18 -
2.4.5 Detekce ostrovního provozu	- 19 -
2.4.6 Systémy WAMS	- 19 -
3 Zvolená metoda algoritmu	- 22 -
3.1 Vyhodnocení frekvence	- 22 -
3.2 Získání fázoru	- 29 -
4 Vlastní realizace algoritmu	- 32 -
4.1 Popis vývojového prostředí.....	- 32 -
4.2 Popis zvolených mikrokontrolérů	- 33 -
4.2.1 Mikrokontrolér ATMEL ATmega162.....	- 34 -
4.2.2 Mikrokontrolér ATMEL ATmega644.....	- 34 -
4.3 Implementace algoritmu	- 35 -
4.3.1 Načtení dat	- 36 -
4.3.2 Výpočet frekvence	- 38 -
4.3.3 Výpočet fázoru.....	- 40 -
4.3.4 Reprezentace a kódování výsledků	- 41 -
4.3.5 Dekódování a výpis na displej	- 44 -
4.4 Hardwarová realizace.....	- 45 -
5 Ověření funkčnosti.....	- 48 -
5.1 Softwarový test	- 48 -
5.2 Hardwarový test	- 50 -

6	Závěr	- 53 -
7	Seznam použité literatury	- 55 -
8	Seznam použitých zkratek	- 57 -
9	Seznam obrázků	- 58 -
10	Seznam tabulek	- 59 -
11	Seznam příloh	- 60 -
12	Obsah přiloženého CD	- 60 -
13	Tištěné přílohy	- 61 -

1 Úvod

V současné době jsou kladeny vysoké nároky na kvalitu a spolehlivost přenosu elektrické energie. Vzhledem k úspornému provozu sítí je vhodné řídit toky energie v elektrické soustavě tak, aby byla síť optimálně využita. Z tohoto důvodu musí být v síti rozmístěno zařízení, umožňující monitoring aktuálních provozních parametrů sítě a nadřazený systém, jenž tyto data zpracovává a vyhodnocuje. Aby mohla být data vzájemně porovnatelná, je využíváno synchronního měření v definovaných časových okamžicích. Na základě těchto dat a znalosti modelu sítě je možné provést vhodnou konfiguraci sítě, aby nedocházelo k provozním stavům na hranici bezpečnosti přenosu. Dojde-li k těmto nenadálým stavům, lze problém řešit rychleji a efektivněji a tím docílit vyššího standardu přenosu elektrické energie.

V teoretické části této práce bude zahrnuta problematika synchronního měření, jejímž výstupem je synchronní fázor elektrického napětí či proudu (případně dva napěťové fázory). Dále funkce synchronního fázoru v měřícím řetězci pro monitorování stavu elektrické sítě. Následuje kapitola věnována možnostem využití synchronního měření na úrovni přenosové soustavy.

Cílem druhé části práce bude praktická realizace zařízení pro měření fázorů pomocí mikrokontroléru. Tomu předchází představení možného způsobu získání fázoru za pomoci statistických metod. V závěru budou předvedeny výsledky získané při simulaci funkce zařízení.

2 Synchronní měření

V této kapitole bude uvedena základní charakteristika fázoru a synchronního fázoru. Dále je zde pojednáno o oblastech využití synchronního měření v elektrických sítích, zejména přenosové soustavy.

2.1 Definice fázoru

Fázor je vektor reprezentovaný amplitudou a fází natočení. Je to způsob reprezentace harmonického signálu, který je využíván pro analýzu střídavě napájených sítí v energetice. Harmonický signál v sinusovém tvaru lze vyjádřit ve tvaru:

$$x(t) = X_m * \cos(\omega t + \varphi) \quad (2.1)$$

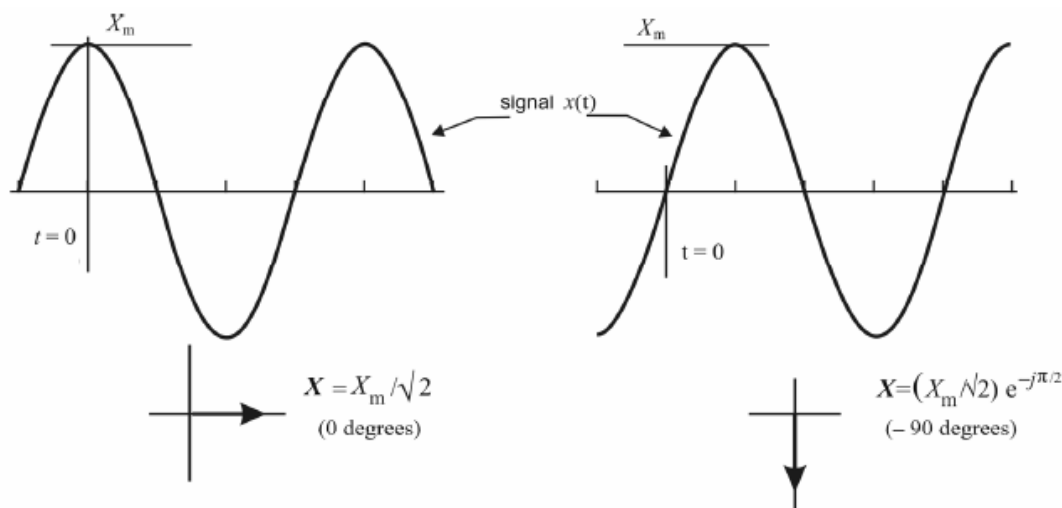
Tuto rovnici můžeme napsat jako fázor ve tvarech

$$\begin{aligned} \mathbf{X} &= \frac{X_m}{\sqrt{2}} * e^{j\varphi} \\ &= \frac{X_m}{\sqrt{2}} * (\cos(\varphi) + j \sin(\varphi)) \\ &= X_r + jX_i \end{aligned} \quad (2.2)$$

kde X_m je amplituda signálu, φ je úhel posunutí/natočení vzhledem k času $t=0$, $\frac{X_m}{\sqrt{2}}$ je efektivní hodnota průběhu a členy s indexem r a i jsou reálné a imaginární části komplexního čísla v gaussově rovině. Hlavní předností fázoru je, že neobsahuje parametr času, charakterizuje tedy periodickou funkci jako celek.

2.2 Definice synchronního fázoru

Synchronní fázor (synchrofázor) $\mathbf{X}$ reprezentuje signál $x(t)$ z rov. 2.1, kde φ je okamžitý fázový úhel při jmenovité frekvenci sítě synchronizované UTC (Coordinated Universal Time). Fázový úhel 0° tedy odpovídá maximální hodnotě signálu viz obr. 2.1. Kladný úhel φ je brán proti směru hodinových ručiček a pohybuje se v rozsahu $\pm 180^\circ$.



Obr. 2.1 Značení synchronních fázorů; zdroj [1]

Harmonickou funkci zapsanou v následující rovnici 2.3

$$x(t) = X_m * \cos(\omega_0 t + \varphi) = X_m * \cos(2\pi f_0 t + \varphi) \quad (2.3)$$

kde f_0 je jmenovitá síťová frekvence (50/60Hz) obsažená ve fázoru (rov. 2.2). Obecně však může být amplituda i frekvence funkcí času $X_m(t)$ a $f(t)$. Proto je definována funkce $g=f-f_0$, kde f_0 je jmenovitá frekvence a g je rozdílová frekvence mezi aktuální a jmenovitou frekvencí. Pokud je funkce g také funkcí času¹, má harmonická funkce tvar rov. 2.4:

$$\begin{aligned} x(t) &= X_m(t) * \cos(2\pi \int f dt + \varphi) \\ &= X_m(t) * \cos(2\pi \int (f_0 + g) dt + \varphi) \\ &= X_m(t) * \cos(2\pi f_0 t + (2\pi \int g dt + \varphi)) \end{aligned} \quad (2.4)$$

Synchrofázor je pak v exponenciálním tvaru vyjádřen rov. 2.5:

$$\mathbf{X}(t) = \frac{X_m(t)}{\sqrt{2}} * e^{j(2\pi \int g dt + \varphi)} \quad (2.5)$$

V případech, kdy $X_m(t)=X_m$ konstantní a $g=\Delta f$ je konstantní odchylka od jmenovité frekvence ($\int g(t) dt = \int \Delta f dt = \Delta f t$), synchrofázor je zjednodušen na rov. 2.6:

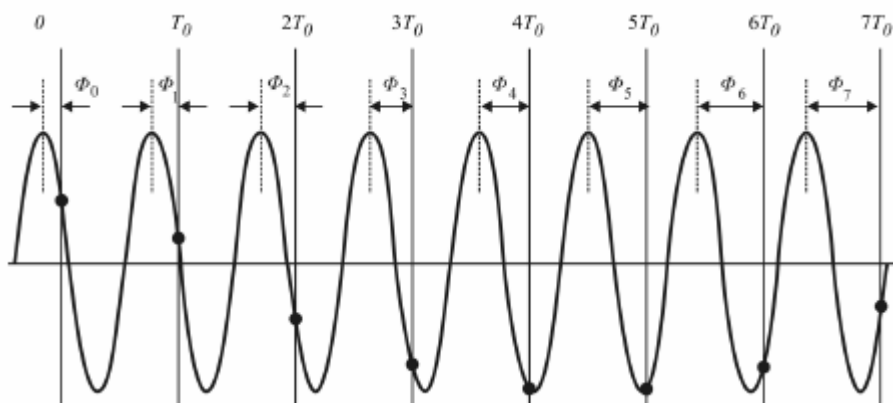
$$\mathbf{X}(t) = \frac{X_m}{\sqrt{2}} * e^{j(2\pi \Delta f t + \varphi)}, \quad (2.6)$$

fázor se otáčí konstantní frekvencí Δf , což je rozdíl mezi skutečnou a nejmenovitou frekvencí.

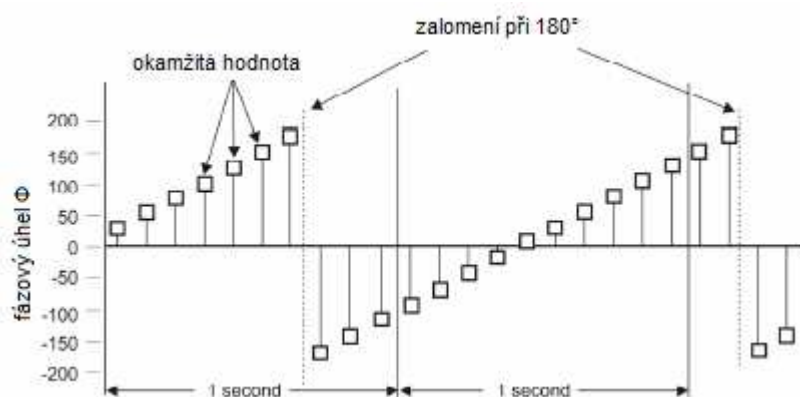
Na následujícím obr. 2.2 je zobrazeno několik period signálu jehož frekvence se liší od jmenovité ($f > f_0$) a časové úseky ($T_0, 2T_0, \dots$), které odpovídají jmenovité frekvenci. Tyto časové

¹ $g(t)=f(t)-f_0$

značky jsou dále reprezentovány fázorem ($X_0, X_1, \dots$). V případě, že bude rozdílná frekvence $f < 2f_0$, bude mít fázor konstantní amplitudu ale fázový úhel ($\Phi_0, \Phi_1, \dots$) se bude měnit, jak zobrazuje následující obrázek. Fázový úhel by v tomto případě stále narůstal konstantním přírůstkem až by překročil 180° . Synchronfázory se však používají v rozsahu -180° až $+180^\circ$, fázový úhel by tedy pokračoval opět od záporných hodnot jak ukazuje obr. 2.3.



Obr. 2.2 Průnik jmenovité frekvence a signálu s $f > f_0$; zdroj [1]



Obr. 2.3 Vzorky odebrané při odlišné frekvenci; zdroj [1]

Všechny měřené veličiny jsou vztaženy ke společné časové základně a frekvenci, to znamená, že jsou mezi sebou přímo porovnatelné. Rozdíly aktuální a jmenovité frekvence jsou odhadnuty z fázového úhlu. Odhad synchronfázoru v sobě zahrnuje i přechodné jevy, jako např. kolísání a místní frekvenční výkyvy. Synchronfázory jsou časově závislé a jejich aktuální hodnota je závislá na frekvenci a čistotě signálu.

2.3 Jednotky pro měření synchronních fázorů

Jednotky pro měření synchronních fázorů (PMU) jsou zařízení připojená k elektrické síti přes měřicí transformátory proudu (MTP) a napětí (MTN). Tyto přístroje vyhodnocují amplitudu napětí a proudu, frekvenci sítě a fázový úhel v daném místě v konkrétních časových okamžicích. Zároveň obsahují komunikační jednotku a pomocí protokolu předávají data nadřazenému systému. Protokol využívaný k přenosu je definovaný normou IEEE C37.118, která také zahrnuje vlastnosti a kvalitu měření požadovanou od přístrojů PMU. Jako komunikační cesta je dnes nejčastěji využit Ethernet. Perioda odesílání dat závisí na důležitosti daného měřicího bodu, často bývá nastavitelná a pohybuje se až do periody síťového napětí. Jednotka PMU měřené veličiny vyjadřuje buď jako jednotlivé fázory, nebo jako části symetrických složek. Zařízení také může obsahovat doplňkové funkce:

- digitální vstupy pro snímání stavů technologických zařízení
- výpočet činného a jalového výkonu
- vícenásobné komunikační porty
- korekce pro chyby amplitudy a fáze měřících transformátorů

2.4 Využití synchronního měření v síti

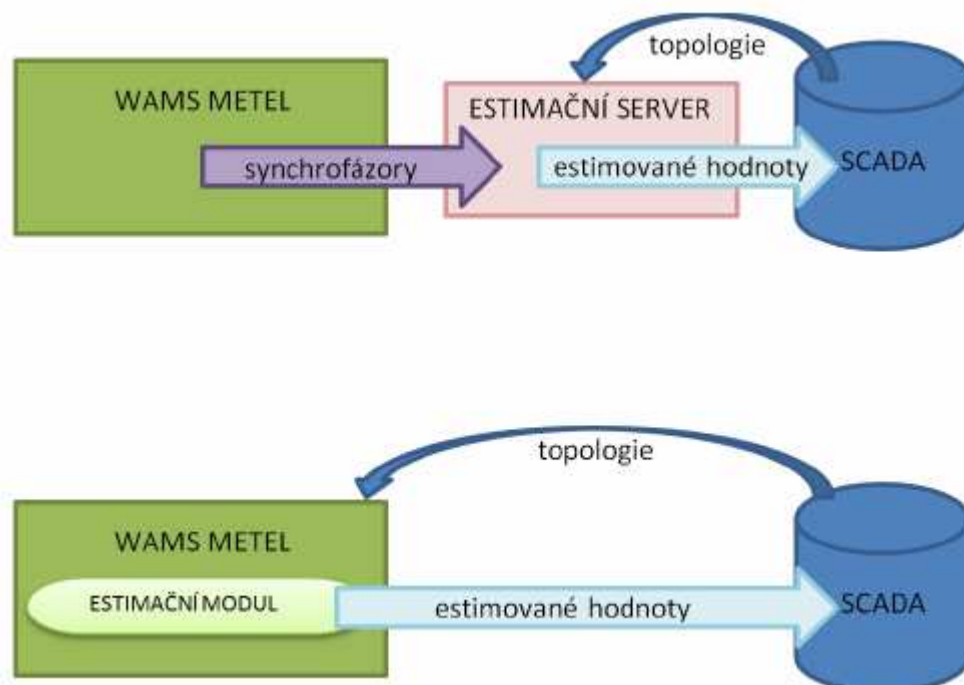
Synchronní měření v síti má velmi široký rozsah využití, zejména pro moderní systémy řízení provozu sítí (bezpečnost, stabilita a finanční efektivita). Podmínkou jeho použití je časová synchronizace z různých měřících bodů v síti. Řízení využívá stavový vektor sítě. Dále je zde využito dalších měření, pro estimaci² tohoto vektoru (napětí, proudů, činných a jalových výkonů). Přesnost je ovlivněna celou řadou zařízení v měřícím řetězci (měřicí transformátory, digitalizace dat) a časovou synchronizací jednotlivých měření.

Pro optimální využití silně zatížených sítí je výhodná znalost aktuálního ustáleného stavu a dynamických změn. V tomto případě je nesynchronizované měření zdrojem velkých chyb. Pro estimaci sítě je synchronní měření výhodné z důvodu možnosti využití porovnávacího výpočtu. Další aplikací je získání dat pro rozsáhlé systémy měření (WAMS), které přináší nový náhled na řešení problémů s přenosem a distribucí elektrické energie.

² Kvalifikovaný odhad na základě zkoumání, šetření a strukturování

2.4.1 Estimace ustáleného stavu elektrických sítí

Estimace slouží k získávání spolehlivých dat pro dispečerské řízení, analýzu a hodnocení provozu. Tento proces upravuje informace plynoucí ze systému WAMS (Wide Area Monitoring System) popsaného níže a systému SCADA (Supervisory control and data acquisition) pro sběr, zpracování a vyhodnocování dat. Estimační proces zpřesňuje získané informace a výsledky vstupují opět do SCADA systému. Jako informační parametr pro předávání dat ze systému WAMS slouží synchronní fázory napětí a proudu. Na obr. 2.4 jsou zobrazeny dvě možnosti toku dat pro estimaci stavu sítě.

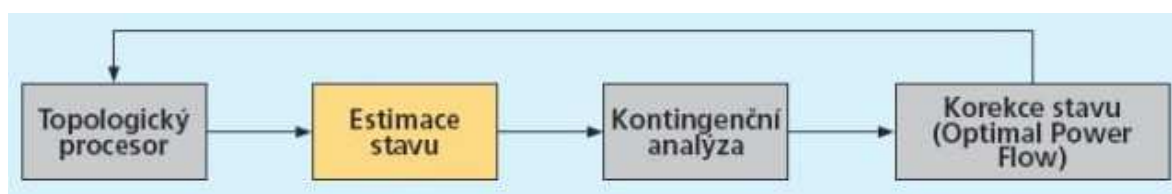


Obr. 2.4 Tok dat při estimaci stavu sítě; zdroj [2]

Pro správnou funkci estimačního programu je nutná znalost topologie sítě. To lze v podmínkách ČR splnit v soustavách 400 a 220 kV. Ale v distribučních sítích je situace jiná. Tyto sítě obsahují velké množství uzlů, což způsobuje několikanásobné zvětšení dimenze výpočetního modelu. Potřebu kontinuálního monitorování stavu řízené soustavy lze realizovat rozšířením metody estimace i do sítí vn, kde je nedostatek měření reálného času. Tento proces slouží jako náhrada klasického výpočtu chodu sítě, ale vyrovnává se lépe s neurčitostí vstupních dat.

S nástupem metod pro flexibilní řízení výkonových toků - FACTS, je nutné uvažovat tyto zásahy v estimačním modelu. Změny v konfiguraci sítě, které tato metoda využívá, vnáší do

estimačního modelu nepřesné informace. Vzhledem ke složitosti velkých sítí není topologie sítě přesně dána a pro model se používá převažující zapojení. To však může při atypickém zapojení způsobovat problém s estimovatelností. S ohledem na zejména ekonomické aspekty je v dnešní době tendence provozovat elektrické soustavy blíže hranicím bezpečného provozu. S tím však souvisí vyhodnocování bezpečnosti ES v reálném čase. V následujícím obr. 2.5 je zobrazeno, jakou funkci má estimace ve vyhodnocovacím procesu pro výpočet ustáleného stavu. V současnosti již nestačí pouze tento pohled, ale přínosné je také dynamické vyhodnocování soustavy a to nejen na úrovni státu. Do vyhodnocení je nutné zahrnout i aktuální modely okolních soustav doplněné o výsledky estimace stavu. Robustnost celého systému se dá zvýšit sběrem dat z dálkových měření a údajů z měření synchronních fázorů.



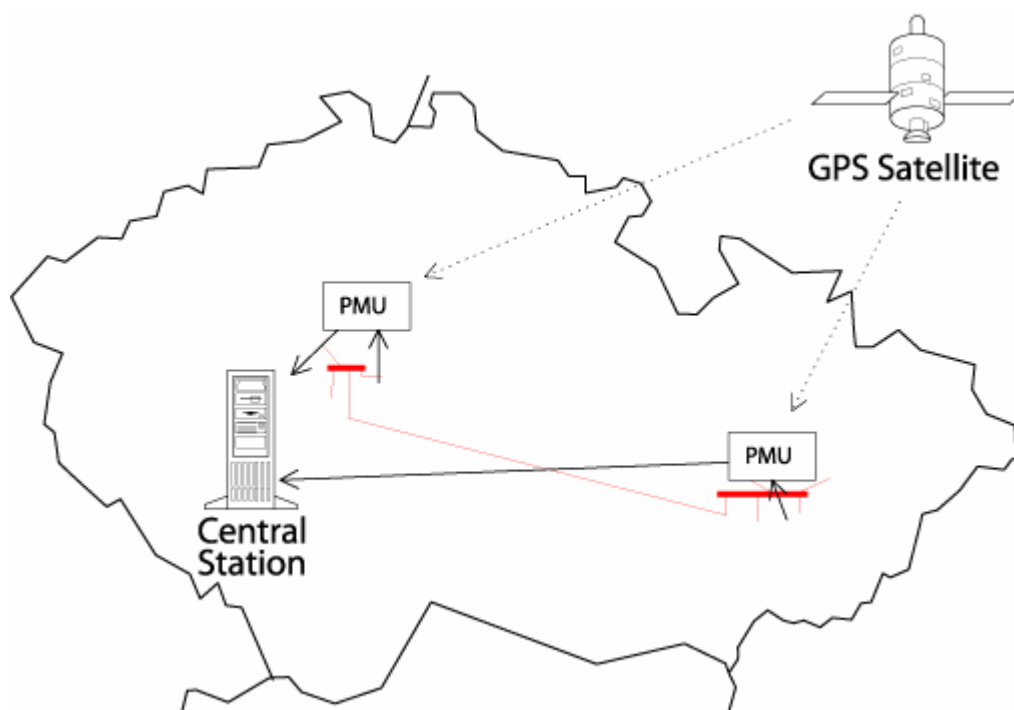
Obr. 2.5 Schéma pro vyhodnocování provozu ES; zdroj [3]

Pro další zvýšení bezpečnosti provozu je nutno monitorovat a kontrolovat stav celého energetického celku. Proto by dispečinky soustav měly obsahovat i koordinační centra, která si spolu předávají data z reálných měření z dané oblasti. Tento systém je však náchylný na ztrátu dat z určité oblasti. V tomto případě musí estimátor vyhledat jiná vhodná měření. Proto se v této situaci využívají dvouúrovňové hierarchické estimace. Tímto krokem se sníží datová zátíženost centra, které dostává již zpracovaná data, ale současně se tento systém stává náchylnější na ztrátu konzistence dat komunikačního kanálu.

Estimace lze využít i v sítích vn, tento směr využití zatím není příliš rozšířen. Tato síť není dostatečně zmapována měřicími systémy pro získání postačujících dat. Dále je zde nutné zaměřit se na jinou metodiku výpočtu, než která je používána v přenosové soustavě, a to vzhledem k odlišným vlastnostem soustavy vn (malý počet smyček, velký počet větví a uzlů, napětíová závislost odběrů, nesymetrické zatížení, atd.). Podrobněji je tato metoda popsána ve zdroji [3].

2.4.2 Monitorování stability sítě

Monitorování stability sítě je využíváno na dlouhých vedeních přenosové soustavy. Vstupními daty jsou hodnoty fázorů napětí, proudů a frekvence měřené na obou koncích vedení. Měření těchto veličin provádí jednotka měření fázorů (PMU). Tato jednotka je synchronizována pomocí signálu GPS, případně pomocí lokálního vysílače přesného času, což je v případě střední Evropy vysílač DCF77³. Tyto jednotky odesílají naměřená data do sběrné stanice, kde se z těchto dat vypočtou charakteristické veličiny, ze kterých se určuje amplituda a frekvence oscilací na tomto vedení. Systém měření znázorňuje obr. 2.6.



Obr. 2.6 Princip monitorování stability sítě; zdroj [2]

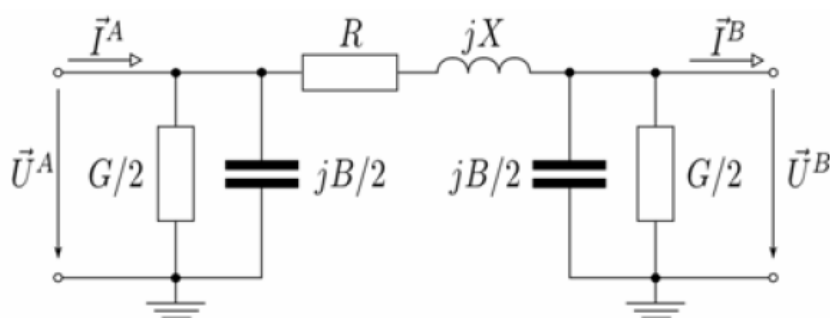
Vzhledem k tomu, že jednotky PMU umožňují kontinuální měření, je možné sledovat dynamické vlastnosti systému a vyslat včasné varování před nestandardními stavy. Lze tak předejít případným oscilacím systému a nabízí se možnost vytvořit si komplexní přehled o stavu systému. Zároveň mohou tyto data sloužit jako vstupní informace pro dynamické modely.

³ Vysílač přesného času UTC umístěný poblíž Frankfurtu vysílajícího na frekvenci 77,5 kHz

2.4.3 Měření parametrů vedení za chodu

Toto měření má velmi blízko k monitorování stability sítě popsané v podkapitole 2.4.2. Zde probíhá měření synchrofázorů proudu a napětí v reálném čase na obou koncích zkoumaného vedení. Z těchto dat se vyhodnocují charakteristické parametry náhradního schématu vedení (viz obr. 2.7) jako činný odpor R , reaktance X , susceptance B a svod G .

Tyto parametry neznamenají pouze konstanty daného vedení, ale jsou zdrojem pro získání dalších informací. Z činného odporu (R) vodiče lze vycházet při výpočtu teploty vodiče, jako měřítko průvěsu vedení, a také činné ztráty na vedení. Z reaktance (X) jsou určeny jalové ztráty, úbytek napětí na vedení i stabilita dané sítě. Ze susceptance (B) vychází jalové ztráty a napětí naprázdno. Svod (G) charakterizuje činné ztráty a kvalitu vedení.

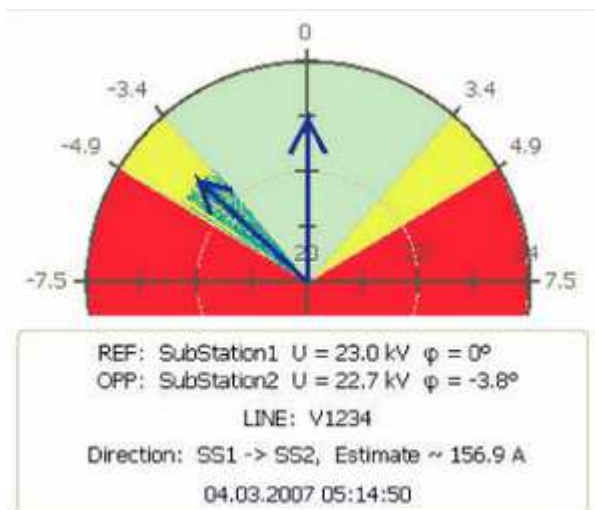


Obr. 2.7 Náhradní schéma vedení pomocí π -článku; zdroj [2]

2.4.4 Podpora bezpečného spínání v distribučních sítích

Pokud jsou jednotky PMU instalovány v rozvodně na přívodní i vývodní straně má dispečer k dispozici informaci, za předpokladu rozepnutého vývodu, o úhlových rozdílech mezi danými vedeními. Této informace může využít k rozhodování při požadavku sepnutí dané sítě. Na následujícím obr. 2.8 je zobrazen možný grafický výstup z měřicího systému.

Systém zobrazuje základní informace o napětí na obou stranách spínače, jejich vzájemný posun, směr toku výkonu a předpokládaný proud po sepnutí. Dále je zde pro větší názornost několik barevných polí definující optimální oblast spínání.



Obr. 2.8 Grafický výstup fázorů napětí; zdroj [2]

2.4.5 Detekce ostrovního provozu

Kontinuálním snímáním, analýzou průběhu kmitočtu, jeho derivací a úhlu fázorů napětí v celém systému WAMS, i systémů okolních (evropských) soustav jsou zjišťovány rozdíly a výsledky jsou odesílány systému SCADA. V uživatelském prostředí jsou následně tyto hodnoty zobrazovány. Synchronní měření také umožňuje za vhodných podmínek ostrovy spojovat a udržovat tak stabilní provoz sítě.

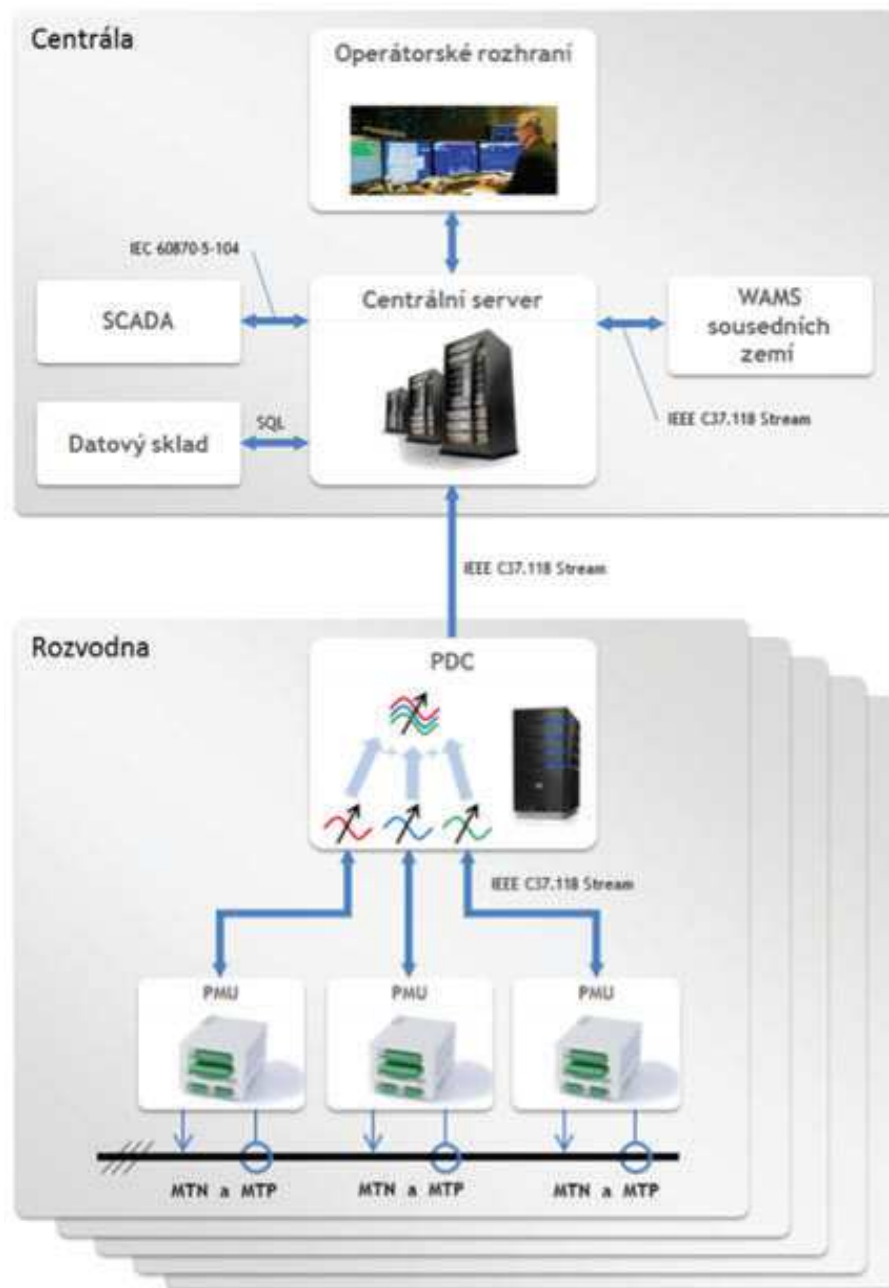
2.4.6 Systémy WAMS

Systémy WAMS (Wide Area Monitoring System) slouží pro vyhodnocování systémových parametrů rozlehlé sítě, např. na národní úrovni. Tento systém, jak zobrazuje obr. 2.9, se skládá ze zařízení pro měření synchronních fázorů napětí a proudů (PMU jednotky), datových koncentrátorů (PDC), centrálního serveru a aplikačního softwaru.

Jednotky pro měření fázorů jsou umístěny ve strategických bodech přenosové soustavy. Tyto jednotky komunikují přes protokol definovaný normou IEEE c37.118.

Informace předávají přes energetické informační kanály do koncentrátoru fázorových dat (PDC). Tento prvek slučuje přijaté fázory a seřazuje je do celků se stejnými časovými značkami, které odesílá centrálnímu serveru ke zpracování. Tato jednotka také shromažďuje informace o provozních stavech elektrické sítě a rovněž je předává nadřazenému systému, ke

zpracování. V případě, kdy dojde k přerušení přenosu dat mezi koncentrátorem a centrálním serverem, umožňuje zařízení dočasně ukládat data pro jejich pozdější doplnění ať už automatické, nebo ruční.



Obr. 2.9 Systém WAMS; zdroj [4]

Funkce centrálního serveru spočívá ve shromažďování dat z PMU, datových koncentrátorů, informací o stavu sítě a jejich vyhodnocování. Zpracované výsledky předává dalším systémům, např. SCADA. Mezi typické vyhodnocovací funkce centrálního serveru patří:

- monitoring fázových úhlů
- detekce a sledování ostrovních provozů
- detekce a analýza oscilací
- detekce snížení rezervy statické stability
- monitorování stability napětí

Operátorské rozhraní slouží k interpretaci analytických funkcí systému WAMS, které jsou využívány převážně v řídicích centrech přenosových sítí.

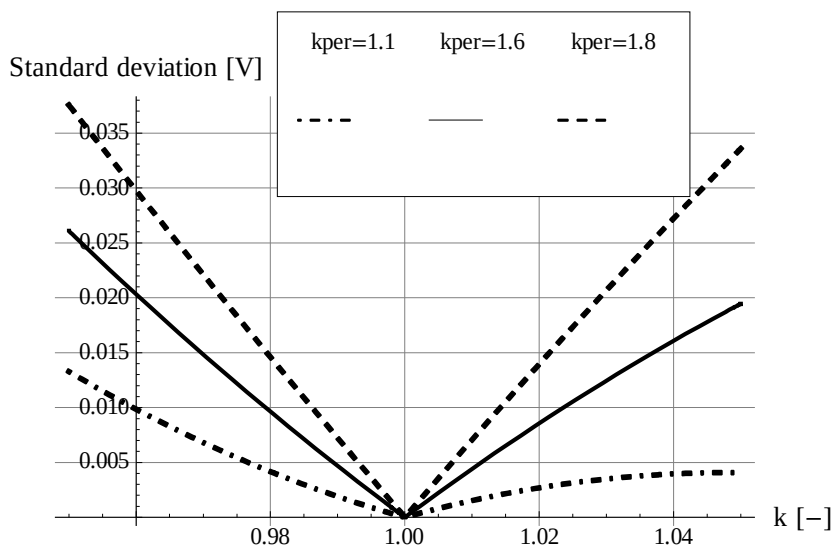
3 Zvolená metoda algoritmu

Tato kapitola je věnována popisu zvolené metody algoritmu, ze které jsou získány hodnoty fázoru. Metoda je založena na statistickém zpracování naměřených dat, o čemž pojednává článek [5]. Hlavní přednost tohoto algoritmu je ve využití jednoduchých matematických operací, které jsou aplikovatelné i v nejnižších výkonových řadách vyráběných mikrokontrolérů. S těmito předpoklady lze realizovat jednoduché, levné a relativně přesné zařízení pro měření v elektrických sítích nejen pro získávání synchronních dat.

Algoritmus je členěn na několik částí. Jedná se o část měření frekvence signálu jejíž součástí je i informace o věrohodnosti výsledné hodnoty ve formě střední kvadratické odchylky. Frekvence je dále využita pro získání výsledného fázoru. Získaný fázor (např. napětí) má primárně algebraický tvar $a+jb$, kde a je reálná část a b je imaginární část komplexního čísla. Alternativně je možné fázor vyjádřit v exponenciálním tvaru $U \cdot e^{j\varphi}$, kde U je modul a φ je fáze komplexního čísla.

3.1 Vyhodnocení frekvence

Jak již bylo výše naznačeno, k vyhodnocení frekvence je využito statistických metod. Nyní předpokládejme spojitý signál viz obr. 3.2 (např. funkce sinus) jehož délka je větší, než je jeho jmenovitá perioda. Tato hodnota je značena k_{per} a pohybuje se v rozsahu jedné až dvou period. Obě tyto hranice není možné zahrnout do výpočtu. Pro velmi malé k_{per} je sice výpočet časově méně náročný, avšak narůstá nepřesnost výsledku, což je signalizováno zvyšujícím se rozptylem σ^2 . Právě rozptyl je již zmiňovaný kvalitativní parametr výsledku. Závislost velikosti zvoleného k_{per} na kvalitě získaného výsledku zobrazuje následující obr. 3.1.



Obr. 3.1 Vliv volby kper na velikosti střední kvadratické odchylky; zdroj [5]

Pro základní představení metody jsou využity následující rovnice pro spojitou funkci. Rovnice 3.1 k získání střední hodnoty funkce sinus:

$$x = \frac{1}{kT} \int_{t=t_0}^{t_0+kT} \sin\left(2\pi \frac{t}{T}\right) dt, \quad (3.1)$$

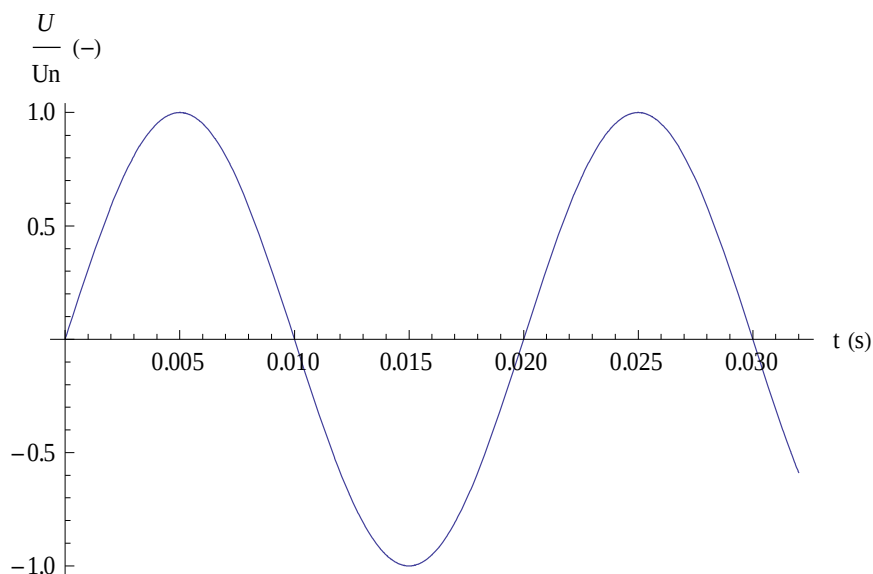
kde kT je zvolená velikost periody signálu a x je střední hodnota zkoumané funkce při zvolené periodě. Následuje vztah 3.2 pro výpočet střední hodnoty výsledků z předchozí rovnice:

$$mx = \frac{1}{(kper - k)T} \int_{t_0=0}^{(kper-k)T} x \cdot dt_0. \quad (3.2)$$

Rovnice střední kvadratické odchylky má tvar:

$$\sigma^2(k, kper) = \frac{1}{(kper - k)T} \int_{t=0}^{(kper-k)T} (x - mx)^2 \cdot dt_0, \quad (3.3)$$

kde $(kper-k)T$ je úsek periody, o kterou je prodloužen signál oproti jmenovité periodě. Integrandem je druhá mocnina rozdílu střední hodnoty funkce pro danou periodu a střední hodnota přes všechny možné periody.



Obr. 3.2 Spojitá funkce sinus ($f=50\text{Hz}$; $k_{per}=1,6$)

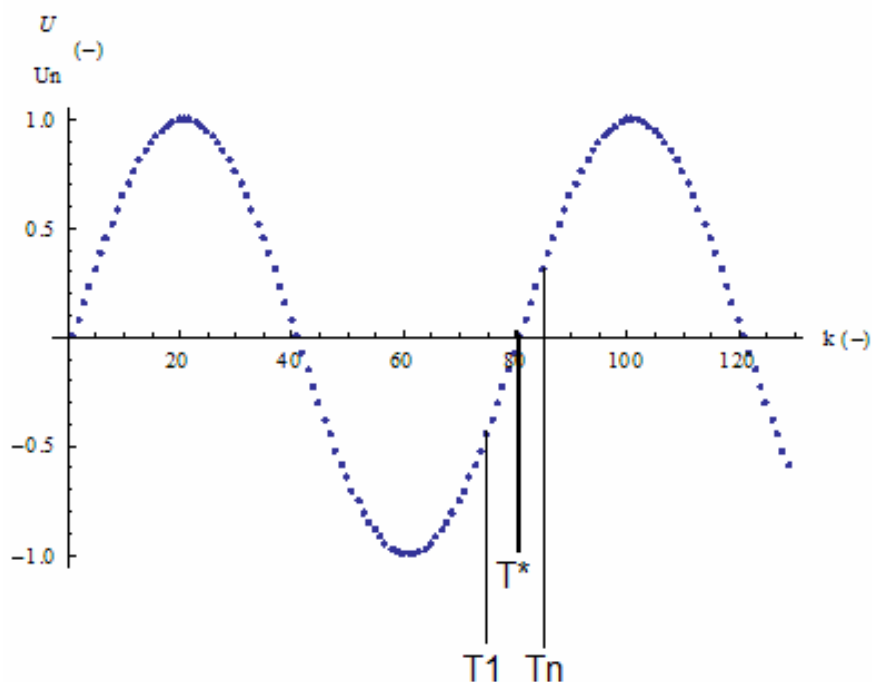
Vzhledem k využití mikrokontroléru v této aplikaci, je nutné diskretizovat spojitý signál (obr. 3.3) a předchozí vztahy vhodně upravit. Počet vzorků signálu závisí na výkonnosti použitého mikrokontroléru. Zde má k význam zvoleného počtu vzorků periody, T je velikost jmenovité periody.

$$x = \frac{1}{kT} \sum_{t=t_0}^{t_0+kT} \sin(2\pi \frac{t}{T}) \quad (3.4)$$

$$mx = \frac{1}{(kper - k)T} \sum_{t_0=0}^{(kper-k)T} x \quad (3.5)$$

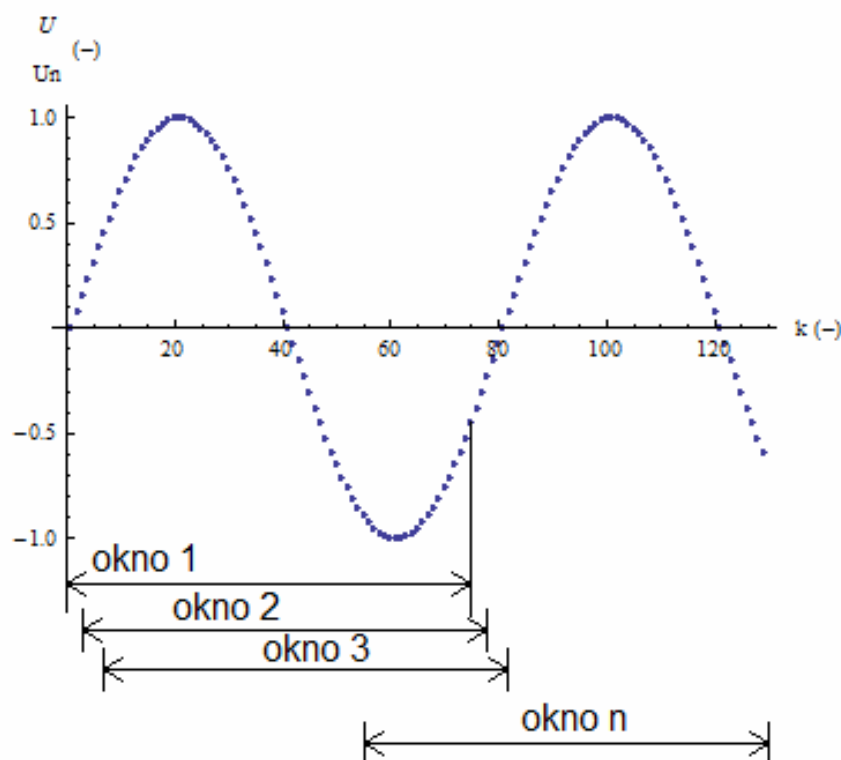
$$\sigma^2(k, kper) = \frac{1}{(kper - k)T} \sum_{t=0}^{(kper-k)T} (x - mx)^2 \quad (3.6)$$

Výpočet probíhá následujícím způsobem. Předpokládejme, že hledaná perioda signálu T^* (napětí či proud) se pohybuje v rozmezí vzorků T_1 až T_n , jak naznačuje obr. 3.3. Je zvolena výpočetní perioda $kT=T_1$ a následuje výpočet rovnice 3.4 s těmito parametry. Získaný výsledek je dále využit při výpočtu střední hodnoty a rozptylu. Nyní následuje posun výpočetního okna o jeden vzorek dopředu, což znázorňuje obr. 3.4, tedy t_0 v rovnici 3.4 je inkrementováno z nuly na vzorek č.1. Opět proběhne výpočet střední hodnoty x a hodnota se uloží. Tento proces se opakuje tolikrát, kolik vzorků je mezi periodou T_n (dle zvoleného měřicího rozsahu) a koncem načteného souboru dat. Po projetí všech oken dané periody je možno provést výpočet střední hodnoty z uložených výsledků dle rovnice 3.5 a následně vyhodnotit rozptyl, rov. 3.6. Tím získáme dvojici hodnot (mx_{T1} , σ^2_{T1}), která bude následně dále vyhodnocována.



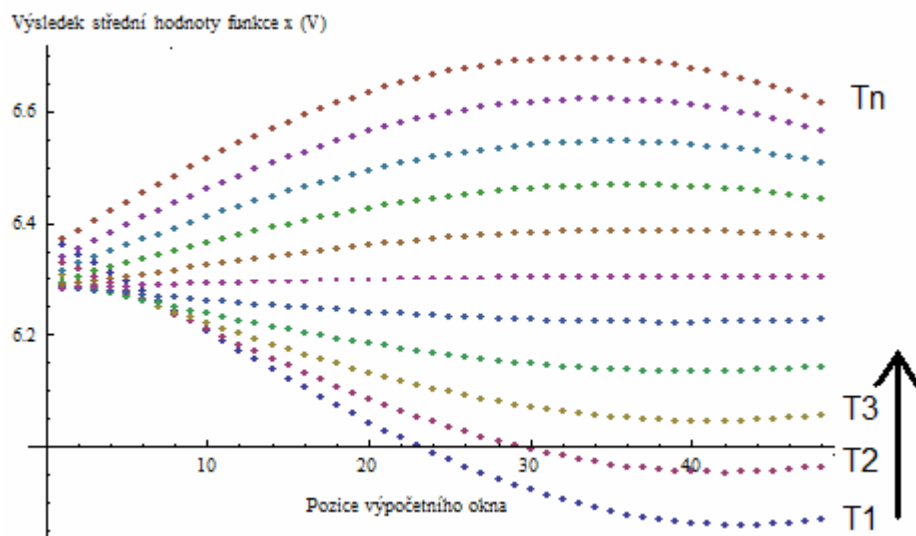
Obr. 3.3 Diskretizovaná funkce sinus ($f=50\text{Hz}$; $k_{\text{per}}=1,6$; $N=128$ vzorků)

V této chvíli je velikost zvolené periody zvýšena o jeden vzorek na hodnotu T_2 . Celý proces výše uvedeného vyhodnocení dat se opakuje. Tedy následuje výpočet střední hodnoty signálu přes periodu $kT=T_2$, uložení výsledku a posun výpočetního okna přes stejný počet vzorků, jako v předchozím případě. Výsledkem je taktéž dvojice hodnot ($m_{x_{T_2}}, \sigma^2_{T_2}$) uložena pro další zpracování. Konkrétní číselné hodnoty počtu vzorků, počtu volených period a počtu výpočetních oken budou uvedeny v kapitole 4 Vlastní realizace algoritmu.

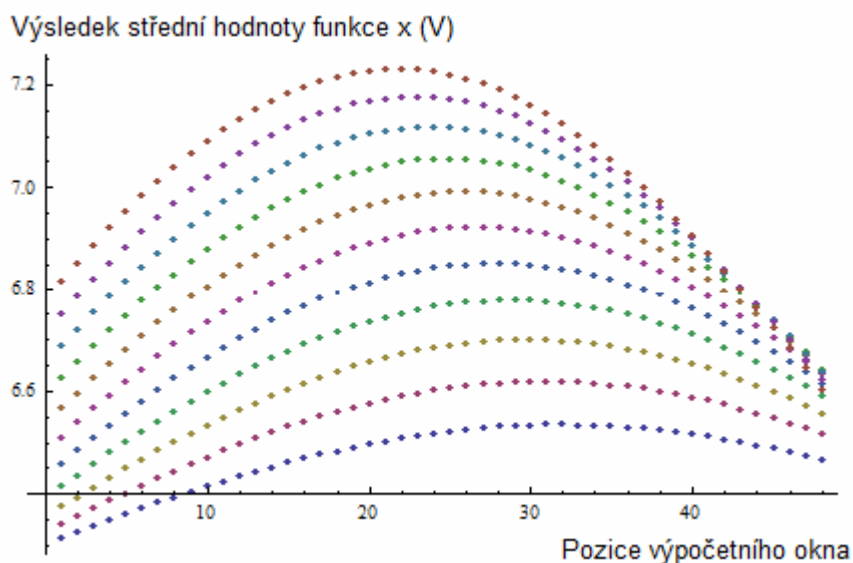


Obr. 3.4 Posun výpočetního okna pro jednu konkrétní periodu

Uvedený vyhodnocovací proces pokračuje přes všechny předpokládané periody až do periody T_n . Příklad získaných dat je zobrazen na obr. 3.5. Na němž je zřejmé, jak se pro konkrétní zvolenou periodu mění výsledek střední hodnoty zkoumaného signálu (jednotlivé po sobě jsou body dané periody T). V zobrazených datech je možné pozorovat, že část vynesných dat má konvexní charakter ($T_1, T_2, \dots$) a část má konkávní charakter ($T_n, T_{n-1}, \dots$). To je známka, že hledaná perioda T^* měřeného signálu se pohybuje mezi zvolenými periodami T_1 a T_n . Máme-li signál o jmenovité periodě sítě (20 ms) a zvolený počet vzorků T_k této periodě taktéž odpovídá, výsledné střední hodnoty jsou nezávislé na pozici výpočetního okna. Mají tedy konstantní hodnotu. Na obrázku 3.6 je jako příklad uveden výsledek při nevhodné volbě testovaných period. Signál má kratší periodu než je nejkratší předpokládaná perioda. Všechna vykreslená data vykazují konkávní charakter, tím pádem v nich nelze najít body s konstantní střední hodnotou.



Obr. 3.5 Výsledky středních hodnot signálu pro různě zvolené periody $T_1, T_2, \dots, T_n$ a různé pozice výpočetního okna



Obr. 3.6 Příklad souboru dat při nevhodně zvoleném rozsahu hledaných period T_1-T_n

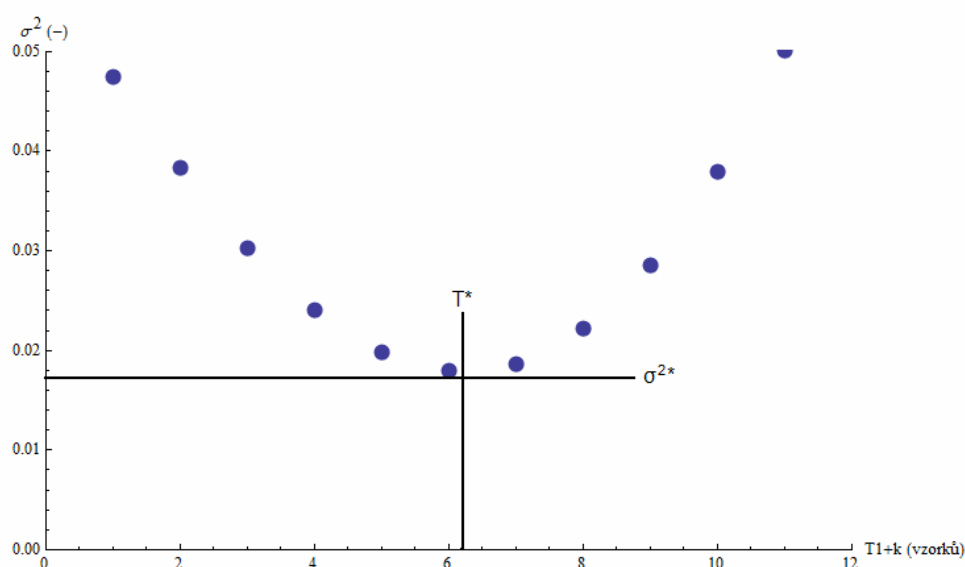
Nyní jsou získána potřebná data pro výpočet skutečné periody a konečně i hledané frekvence. Na obr. 3.8 jsou vyneseny body, jejichž souřadnice značí vzorky odpovídající periodě a jim úměrná hodnota rozptylu. V případě správně nastavených mezí je patrné minimum patřící hledané periodě. Osa x je vynášena v hodnotách volených period, tedy vzorek 1 odpovídá periodě T_1 , vzorek 2 periodě T_2 , atd. Jak je zřejmé hledaná perioda ve většině případů neodpovídá nejnižšímu vykreslenému bodu, proto je nutné použít vhodnou metodu hledání minima nespojitě funkce, která zároveň dopočte odpovídající hodnotu rozptylu.

Ze získaných materiálů, viz zdroj [6] byla využita metoda hledání minima spočívající v prokladu parabolou. Tři nejnižší body jsou dosazeny do rovnice 3.7 a zpětné dosazení výsledku do rovnice paraboly pro získání odpovídající hodnoty rozptylu generuje rovnice 3.8. V naprosté většině případů jsou tyto tři nejnižší body okolo sebe.

Rovnice 3.7 pro hledání minima odpovídající hledané periodě, v níž x_1 , x_2 a x_3 jsou tři nejnižší vypočtené hodnoty period, kde x_2 je z nich nejnižší. Proměnné y_1 , y_2 a y_3 jsou vypočteným periodám odpovídající rozptyly, kde y_2 je z nich nejnižší.

$$T^* = \frac{y_1 + 2x_2y_1 - 4x_2y_2 - y_3 + 2x_2y_3}{2(y_1 - 2y_2 + y_3)} \quad (3.7)$$

$$\sigma^{2*} = -\frac{y_1^2 + (-4y_2 + y_3)^2 - 2y_1(4y_2 + y_3)}{8(y_1 - 2y_2 + y_3)} \quad (3.8)$$



Obr. 3.7 Zobrazení vypočtených dvojic zvolené frekvence a odpovídající rozptyl

Získáním hledané periody T^* a jí odpovídající rozptyl σ^{2*} je proces získání frekvence téměř u konce. Je však nutné si uvědomit, že získaná perioda má charakter počtu vzorků, odpovídajících skutečné periodě. Pro získání skutečné periody T je nutné znásobit vzorkovací periodou T_{vzork} , jak zobrazuje následující rovnice 3.9.

$$T = T^* \cdot T_{\text{vzork}} \quad (3.9)$$

Hledaná frekvence je pak

$$f = \frac{1}{T}. \quad (3.10)$$

Jak je zřejmé, údaj výsledného rozptylu již do žádných dalších výpočtů nevstupuje, ale je to velmi vhodný kvalitativní parametr pro určení věrohodnosti výsledné frekvence. Tato metoda zpracovává navzorkovaný signál. S vyšší přesností výsledku souvisí větší počet vzorků na jednu periodu. Je tedy výhodné minimalizovat vzorkovací periodu a tím získat velký objem dat, proti tomuto faktu však vstupuje časová náročnost prováděných operací. Časová náročnost zvyšujícím se počtem vzorků neúměrně rychle narůstá.

3.2 Získání fázoru

Nyní bude uveden proces získání fázoru, jako výchozího produktu této metody. V této části algoritmu je rovněž použito statistických metod pro vyhodnocení dat. Vstupními informacemi jsou, stejně jako v předchozí kapitole, navzorkovaná data měřeného signálu, dále pak číslo vzorku odpovídající skutečné periodě získaného výše uvedeným procesem a tabelované hodnoty funkce sinus a cosinus. Tato část algoritmu je rozdělena na dvě části. První část se zabývá transformací signálu do mezí, které jsou využity v následném výpočtu fázoru. Ve druhé části je pak získán vlastní výsledný fázor.

Tabelované funkce sinus a cosinus mají shodný počet vzorků. Má-li být výpočet jednotlivých složek fázoru (rovnice 3.14 a 3.15) korektní, je nutné aby také navzorkovaný měřený signál měl shodný počet vzorků. K tomuto účelu slouží transformační algoritmus. V tomto algoritmu je nejprve vyjádřeno přepočtení měřítko viz rovnice 3.11.

$$meritko = \frac{T^*}{T_{nom}}, \quad (3.11)$$

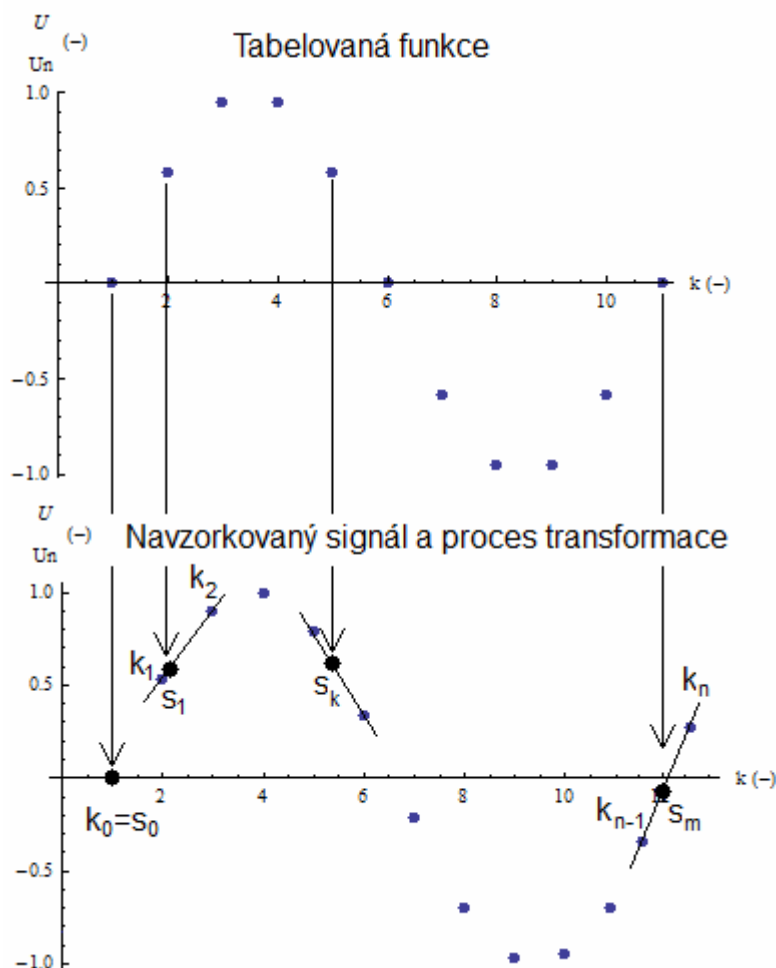
kde T_{nom} označuje počet vzorků již uvedených tabelovaných funkcí a T^* je výsledné číslo vzoru, které je vstupem do této metody. Následuje převzorkování celého vstupního signálu na požadovaný počet vzorků, při čemž využijeme lineární interpolační vzorec 3.12.

$$y_{sk} = y_{ki} + (s_k - k_i) \cdot \frac{y_{ki+1} - y_{ki}}{k_i - k_{i+1}}, \quad (3.12)$$

kde y_{sk} je funkční hodnota hledaného bodu, y_{ki} a y_{ki+1} jsou funkční hodnoty interpolačních mezí původního signálu, k nim odpovídající čísla vzorků k_i a k_{i+1} a konečně s_k , což je nové umístění vzorku na x-ové ose. Hodnotu s_k získáme z rovnice 3.13.

$$s_k = k_i * \text{meritko} \quad (3.13)$$

Proces interpolace je znázorněn na obr. 3.8. V jeho horní části je vykreslena tabelovaná funkce sinus, která zde však slouží pouze pro odvození požadovaného počtu vzorků výsledku transformace. Spodní část obrázku naznačuje princip lineární interpolace, kde původní vzorky $k_0, k_1, \dots, k_n$ přechází na s_0, s_1, s_m . Obecně platí $n \neq m$.



Obr. 3.8 Transformace vstupních dat interpolací

První hodnota nově transformovaného signálu je shodná s původní, tedy $k_0 = s_0$. Pro všechny ostatní platí výše uvedený popis přepočtu. V této chvíli jsou k dispozici uložené hodnoty tabelovaných funkcí a transformovaného signálu, které mají shodný počet vzorků a je možné přistoupit k výpočtu fázoru.

Fázor (např. napětí) je získán v algebraickém tvaru, tedy $U = a + jb$. Jakmile je získána frekvence první harmonické složky spočteme koeficienty fourierovy řady dle diskrétních vztahů 3.16 a 3.17. Tyto vztahy vycházejí ze spojitých rovnic viz vztahy 3.14 a 3.15

$$a = \frac{2}{T} \int_{t=0}^T s(t) \cdot \cos\left(2\pi \frac{t}{T}\right) \cdot dt \quad (3.14)$$

$$b = \frac{2}{T} \int_{t=0}^T s(t) \cdot \sin\left(2\pi \frac{t}{T}\right) \cdot dt \quad (3.15)$$

$$a = \frac{2}{kT} \sum_{k=0}^T s_k \cdot \cos\left(2\pi \frac{k}{T}\right) \quad (3.16)$$

$$b = \frac{2}{kT} \sum_{k=0}^T s_k \cdot \sin\left(2\pi \frac{k}{T}\right), \quad (3.17)$$

kde a je reálná složka fázoru a b je imaginární složka, kT je počet vzorků periody. Funkce sinus a cosinus jsou již zmíněné tabelované hodnoty. Tabelované hodnoty jsou uvedeny, neboť je časově výhodnější načítat data pro výpočet v mikrokontroléru z paměti, než když jsou hodnoty funkcí pokaždé počítány. Funkce s_k jsou k -té vzorky transformovaného vstupního signálu.

Méně častou podobou je goniometrický tvar, který má formu rov. 3.18. Další varianta zápisu fázoru (napětí) je exponenciální tvar, jehož zápis vyjadřuje rovnice 3.19 a přímo vychází z goniometrického tvaru.

$$\mathbf{U} = |U| \cdot (\cos(\varphi) + j \cdot \sin(\varphi)), \quad (3.18)$$

$$\mathbf{U} = |U| \cdot e^{j \cdot \varphi}, \quad (3.19)$$

kde $|U|$ je absolutní hodnota fázoru, též modul a φ je argument či fázový úhel. Převodní vztahy z algebraického a goniometrického tvaru zobrazují následující rovnice 3.20 pro výpočet modulu

$$|U| = \sqrt{a^2 + b^2} \quad (3.20)$$

a 3.21 pro výpočet argumentu, kterým předchází převodní vztahy mezi algebraickým a goniometrickým vztahem

$$\cos(\varphi) = \frac{a}{|U|}, \sin(\varphi) = \frac{b}{|U|}$$

$$\varphi = \arccos \frac{a}{|U|} \text{ obdobně } \varphi = \arcsin \frac{b}{|U|} \quad (3.21)$$

Tímto byly získány všechny použitelné tvary výsledného fázoru, které jsou vhodné pro následnou práci s těmito fázory. Vstupují-li fázory do vztahů kde se vyskytuje sčítání/odčítání, je vhodné použít algebraický tvar, neboť se jednotlivé složky přímo sčítají/odčítají. Pokud se ve vztazích objevuje násobení/dělení je vhodný exponenciální tvar, protože moduly se mohou přímo násobit/dělit a argumenty sčítat/odčítat.

4 Vlastní realizace algoritmu

Tato kapitola se zabývá již konkrétní realizací a aplikací výše popsaného algoritmu. Bude zde popsáno konkrétní vývojové prostředí a jeho základní vlastnosti. Následuje popis a důvody použití konkrétního typu mikrokontroléru. V závěru bude popsán postup implementace programu vytvořeného v jazyce C pro zmíněný typ mikrokontroléru.

4.1 Popis vývojového prostředí

Pro tuto práci bylo využito vývojové prostředí od firmy ATMEL, a to **AVR Studio 5**. Tento editor je jednak uživatelsky snadno ovladatelný, a dále nabízí velké množství funkcí k podpoře vývoje programů pro mikrokontroléry již zmiňované firmy, viz ilustrační obr. P1 obsažený v tištěné příloze.

Při tvorbě algoritmu byly využívány zejména funkce prediktivní psaní programového kódu, což velmi urychluje a zjednodušuje práci. Další významnou součástí je simulace běhu programu včetně možností vkládání záchytných bodů, náhled viz obr. P2 v příloze. Lze zde nastavit taktování mikrokontroléru a měřit časy mezi jednotlivými úseky programu. Toho bylo využito při časové optimalizaci vyvíjeného kódu a pro konečné vyhodnocení délky trvání výpočtu fázoru. Software rovněž umožňuje automatickou optimalizaci při překladu, a to v pěti režimech. Aplikace rovněž umožňuje ladění programu přímo na čipu (mikrokontroléru), této možnosti však nebylo využito z důvodu nutnosti použití jiného typu programovacího rozhraní. AVR Studio rovněž podporuje velké množství knihoven, z nichž byly využity knihovny pro operace s řetězcí, pro složitější matematické operace a pro výpis výsledků na LCD displej.

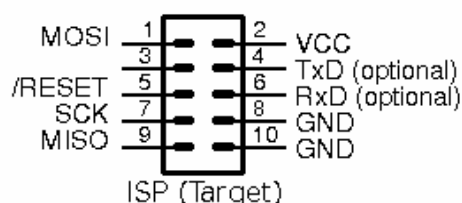
Další nedílnou součástí programovacího řetězce je softwarová aplikace spolupracující s programátorem (hardwarový-HW přípravek). Tuto funkci sice AVR Studio 5 podporuje, avšak pouze s jinými typy programátorů. Zmíněný program se nazývá **AVR8 Burn-O-Mat**. Tato aplikace, mimo přenosu kódu do programovaného čipu podporuje jeho základní konfigurace v podobě nastavení pojistek mikrokontroléru a práci s pamětí EEPROM.

Hardwarový přípravek k programování mikrokontrolérů s názvem USBasp spolupracuje s výše uvedeným programem. Tato jednotka viz obr. 4.1 převádí informaci z USB portu na rozhraní ISP (Serial Peripheral Interface) jejíž programovací piny jsou zobrazeny na obr. 4.2.

Označení těchto pinů koresponduje s popisem pinů na čipu, viz následující kapitola. Tento programátor vychází z koncepce popsané ve zdroji [8]. Výhodou tohoto způsobu programování je možnost aktualizace programu v již nainstalované aplikaci, bez nutnosti vyjímát z ní mikrokontrolér.



Obr. 4.1 USBasp programátor s rozhraním ISP



Obr. 4.2 Zapojení pinů USBasp programátoru; zdroj [7]

K programování je dále možné použít rozhraní JTAG (Joint Test Action Group). Programátory podporující tento protokol umožňují ladění aplikace přímo na čipu, což je funkce kterou podporuje i AVR Studio 5.

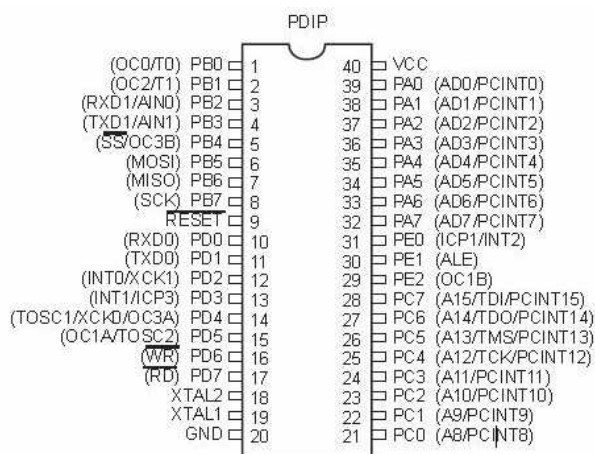
4.2 Popis zvolených mikrokontrolérů

V této kapitole budou popsány základní vlastnosti mikrokontrolérů, které byly využity při tvorbě aplikace. Nastavení periferních obvodů daného mikrokontroléru bude popsáno v kapitole zabývající se implementací. Procesory této řady byly použity jednak z důvodu vyhovujícího programovacího prostředí s řadou funkcí, a také proto že s nimi mám již řadu zkušeností.

4.2.1 Mikrokontrolér ATMEL ATmega162

Tento mikrokontrolér, konkrétně typ ATmega162-16PU slouží pouze jako pomocná jednotka pro zobrazování výsledků. Konfigurace pinů je na obr. 4.3. Jeho vstupem jsou data, která jsou přijímána přes sériovou linku USART (Universal Synchronous / Asynchronous Receiver and Transmitter). Výstupem je výpis vypočtených dat na znakovém LCD displeji.

Jedná se o osmi-bitovou jednotku, která obsahuje 16kb programové paměti Flash, 1kb datové paměti SRAM a 512b paměti EEPROM. Taktovací frekvence externím krystalem je možná až do frekvence 16MHz, která byla také pro taktování zvolena. Napájecí napětí 2,7-5,5 V. Z obsažených periférií byly využity osmi nebo šestnácti bitové časovače a jednotka USART pro přenos dat.



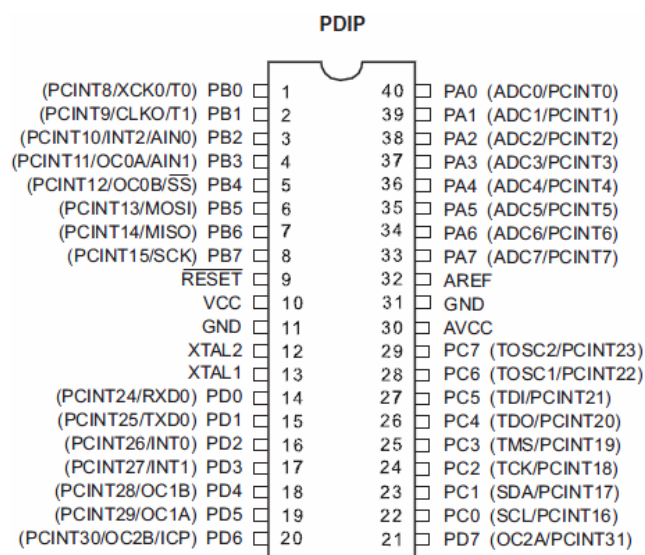
Obr. 4.3 Konfigurace pinů pro ATmega 162

Tento obvod byl zvolen zejména proto, že obsahuje dvojici USART jednotek a mohl by být použit pro další externí komunikační zařízení (PC, bezdrátový přenosový modul, atd.).

4.2.2 Mikrokontrolér ATMEL ATmega644

Mikrokontrolér ATMEL ATmega644-20PU slouží jako hlavní výpočetní jednotka PMU, pro měření fázorů napětí a proudu. Označení vývodů jsou na obr. 4.4. Vstupem jsou dva analogové signály na integrovaném A/D (analogově/digitálním) převodníku a výstupem je kódovaná sériová informace obsahující veškeré výsledky. Dále je zde signalizace průběhu výpočtu pomocí LED a signalizace odesílání dat.

Taktéž se jedná o osmi bitový čip jehož programová paměť Flash má velikost 64kb a datová paměť SRAM má 4kb, paměť EEPROM dosahuje 2kb. Taktovací frekvence krystalu je až 20MHz a rovněž byla zvolena pro dosažení maximálního výpočetního výkonu. Napájecí napětí je nutné konkrétně pro tuto frekvenci udržet v rozsahu 4,5-5,5 V. Jako periferní obvody byly využity osmi či šestnácti bitové časovače, jednotka USART a deseti bitový A/D převodník.



Obr. 4.4 Konfigurace pinů pro ATmega644

Zmiňovaný obvod byl vybrán, neboť se jedná o jeden z nejvýkonnějších mikrokontrolérů v řadě ATmega s dostatkem datové a programové paměti i relativně vysoké taktovací frekvenci. Právě datová paměť je v této aplikaci hojně využívána, protože zařízení pracuje s velkým objemem dat, které je nutné mít po většinu výpočtu k dispozici.

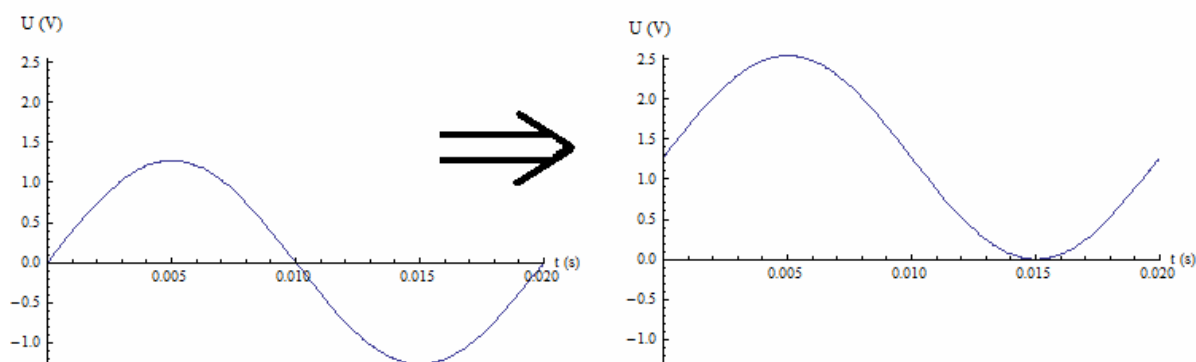
4.3 Implementace algoritmu

Tato kapitola je zaměřena na konkrétní implementaci výše popsaného algoritmu. Parametry byly voleny vzhledem k přesnosti očekávaného výsledku a rychlosti běhu programu. V příloze na přiloženém CD je vytvořený projekt v jazyce C. Složka nazvaná **DP_Synchrofazory** obsahuje podsložky **synchrofazory**, kde je program pro jednotku PMU a složka **zobrazeni.**, kde je uložen kód pro zobrazovací jednotku. Program je rozdělen na jednotlivé funkční bloky, které budou blíže popsány. Schéma zapojení jednotky PMU a zobrazovací jednotky jsou obsaženy v kapitole 4.4 Hardwarová realizace.

4.3.1 Načtení dat

S kvalitou načtených dat úzce souvisí i přesnost výsledku. Není zde kladen tak významný důraz na přesnost A/D převodníku, spíše na přesně definovanou vzorkovací frekvenci. Vzorkovací kmitočet má zásadní vliv na přesnost výsledné frekvence, neboť při výpočtu je tento kmitočet dosazován ve výpočtu. Vzhledem k tomu, že jsou načítána data odpovídající napětí a proudu, je nutné aby si jednotlivé vzorky vzájemně časově odpovídaly. Tento mikrokontrolér však obsahuje pouze jeden nezávislý převodník a je tedy nutné vzorkovat signály postupně tak, aby nedocházelo k příliš velké chybě fázového posunu mezi oběma signály.

Mikrokontrolér obsahuje jednopolaritní osmi kanálový A/D převodník s rozlišením deseti bitů. Využity byly dva kanály, a to konkrétně ADC0 pro napěťový vstup a ADC1 pro proudový vstup, viz obr. 4.4. Vzhledem k tomu, že převodníky jsou jednopolaritní a měřený signál je střídavý, byl vytvořen jednoduchý obvod, jehož funkci znázorňuje obr. 4.5 a jeho schéma je uvedeno v kapitole 4.4. Tento obvod posouvá nulovou úroveň signálu na napětí přibližně poloviny rozsahu převodníku, tedy +1,27 V. Toto jednoduché zapojení s sebou nese jednu základní nevýhodu, a tou je nutnost odděleného záporného pólu napájení od nulové úrovně signálu. Rozkmit signálu tedy musí být maximálně 2,56 V, aby nedocházelo ke zkreslení měřeného napětí.



Obr. 4.5 Princip převodníku napěťové úrovně

A/D převodník obsahuje tři základní registry pro konfiguraci (ADMUX, ADCSRA a ADCSRB) a registr výsledku převodu ADC skládající se z registrů ADCH a ADCL. V registru ADMUX jsou nastaveny následující parametry:

- referenční hodnota napětí pro převodník (zvolena vnitřní reference 2,56 V)

- zarovnání výsledků převodu⁴ (zvolena vlevo)
- používaný kanál (střídavě ADC0, ADC1)

Registr ADCSRA obsahuje parametry:

- spuštění převodníku
- start převodu v případě jednorázové konverze
- start převodu od definovaného zdroje
- spuštění systému přerušení při dokončení převodu
- nastavení předděličky frekvence z hlavního oscilátoru pro převodník (nastaveno $f_{osc}/64$)

Registr ADCSRB obsahuje:

- definování zdroje spuštění převodu (při přetečení časovače Timer/Counter1)

Proces spouštění převodníku je tedy volen od časovače 1. Dokončí-li se převod jednoho kanálu je vyvoláno přerušení, kde se obsah registru uloží do pomocné proměnné a nastaví se registr ADMUX na druhý kanál. Po každém převodu se kanály střídají, aby bylo dosaženo co nejmenšího chybového fázového posuvu. Jednotlivé hodnoty se ukládají do dvou polí.

Rychlost převodu závisí na nastavení předděličky (f_{AD}) pro časování převodníku. Katalog udává, že by se tato hodnota měla pohybovat mezi 50 až 200 kHz pro maximální rozlišení 10 bitů. Je zde však možnost zvolit vyšší vstupní frekvenci do převodníku za cenu nižší přesnosti. Pro dosažení lepších parametrů je nutné doplnit zařízení o externí A/D převodníky. Dobu převodu v závislosti na typu spouštění převodníku zobrazuje tab. 1. V našem případě platí hodnota konverzního času 13,5 cyklu. Pro určení vzorkovací frekvence (f_{vz}) slouží následující vztahy

$$f_{AD} = \frac{f_{osc}}{64} = 312,5 \text{ kHz}, \quad (4.1)$$

kde 64 je dělicí poměr předděličky a f_{osc} je frekvence oscilátoru

$$f_{vz} = \frac{f_{AD}}{13,5} = 23,148 \text{ kHz}, \quad (4.2)$$

což odpovídá periodě vzorkování 43,2 μ s. Je však nutné si uvědomit, že jsou načítány dva kanály a reálná vzorkovací frekvence je maximálně poloviční. Velikost proměnných obsahující načtený

⁴ volba vpravo: nejnižších osm bitů je uloženo v registru ADCL zbylé dva v registru ADCH; volba vlevo: horních osm bitů je uloženo v ADCH, zbylé dva v registru ADCL

signál, byla zvolena na 160 vzorků v periodě při jmenovité frekvenci sítě (50 Hz). Z principu algoritmu je nutné vzorkovat delší úsek, a proto byl zvolen 1,6 násobek periody, to je 256 vzorků. Tato hodnota odpovídá vzorkovací frekvenci 8 kHz na kanál. Celková vzorkovací frekvence je tedy 16 kHz. Jak je patrné vzniká zde ještě malý prostor pro navýšení množství vzorků. Tento prostor prozatím nebyl využit. Je však nutné zde ponechat malou rezervu.

Tab. 1 Konverzní čas A/D převodníku pro různé nastavení; zdroj [9]

Condition	Sample & Hold (Cycles from Start of Conversion)	Conversion Time (Cycles)
First conversion	14.5	25
Normal conversions, single ended	1.5	13
Auto Triggered conversions	2	13.5
Normal conversions, differential	1.5/2.5	13/14

Spouštění převodníku je tedy voleno přetečením šestnácti bitového časovače. Časovač je nastaven v režimu volného běhu s povolením přerušení při přetečení. Během tohoto přerušení se střídavě ukládá obsah načtené proměnné do pole vzorků proudu a napětí. Pro dosažení přesného spouštěcího času (přetečení časovače) je nutné zvolit vhodnou velikost předděličky časovače n_{cs} (registr TCCR1B), v tomto případě signál dělen není. Následně se po signálu přerušení zapíše do registru TCNT1 taková hodnota, aby do přerušení scházelo právě tolik vzorků, kolik odpovídá zvolenému času, dle vztahu 4.3.

$$TCNT1 = \frac{65535 \cdot n_{cs} \cdot f_{vz} - f_{osc}}{n_{cs} \cdot f_{vz}} = 64285 \quad (4.3)$$

Při simulaci však bylo zjištěno, že reálná perioda přerušení se liší. Je to dáno tím, že k předpokládanému času se započítává zpoždění, které nastává při ukládání návratových hodnot po generaci přerušovacího signálu. Bylo tedy nutné korigovat výslednou hodnotu tak, aby reálná perioda zůstala stejná jako požadovaná. Obsah registru je po korekci $TCNT1=64339$.

4.3.2 Výpočet frekvence

Tato kapitola je zaměřena na proces výpočtu frekvence. Výpočet je zahájen ihned po načtení obou vstupních dat. Proces je rozdělen na několik po sobě jdoucích operací s daty. Tyto operace jsou reprezentovány funkcemi v algoritmu, jejichž princip bude následně přiblížen.

Funkce **zvecPer()** slouží k postupnému navyšování délky periody viz princip výpočtu v kapitole 3.1. Hledaná frekvence se pohybuje v těsné blízkosti jmenovité hodnoty 50 Hz, a proto byla počáteční hodnota periody zvolena na 150-tém vzorku a koncová na 170-tém, což odpovídá frekvencím 53,33 Hz a 47,06 Hz dle následujícího vztahu

$$f = \frac{f_N \cdot 160}{150} = 53,33 \text{ Hz.}$$

Tyto dvě hodnoty dávají reálný měřitelný rozsah zařízení. Z této funkce je volána další funkce **integrujPer()**, která realizuje vztah 3.4, což je integrace navzorkovaného signálu. Pro numerickou integraci signálu lze použít různé metody: Obdélníková, Lichoběžníková a Simpsonova, které jsou popsány např. v [11]. V tomto případě byla zvolena Simpsonova metoda.

Tato metoda spočívá v nahrazení integrantu polynomem druhého stupně. A dosahuje přesnějších výsledků než ostatní metody. Výpočet znázorňuje rovnice 4.3 pro sudé výpočetní okno a 4.4 pro liché výpočetní okno.

$$\int_a^b f(x)dx = \frac{h}{3} \left[f(x_0) + 2 \cdot \sum_{i=1}^{n-1} f(x_{2i}) + 4 \cdot \sum_{i=1}^n f(x_{2i-1}) + f(x_{2n}) \right], \quad (4.3)$$

$$\int_a^b f(x)dx = \frac{h}{3} \left[f(x_1) + 2 \cdot \sum_{i=1}^{n-1} f(x_{2i+1}) + 4 \cdot \sum_{i=1}^n f(x_{2i}) + f(x_{2n+1}) \right], \quad (4.4)$$

kde h je krok vypočtený

$$h = \frac{b - a}{2 \cdot n},$$

n je polovina počtu vzorků a celkový počet vzorků $2n$ tedy musí být sudé číslo.

Rovnice byla rozdělena na čtyři sčítance (v hranaté závorce), které bylo nutné zvlášť vyčíslit. První/čtvrtý sčítanec přímo odpovídá první/poslední hodnotě zkoumaného souboru dat. Druhý sčítanec vyjadřuje součet všech prvků se sudým indexem. Třetí sčítanec potom sčítá členy s lichým indexem. Jak je zřejmé pro zjištění jediného výsledku střední hodnoty signálu pro zvolenou periodu je nutné velké množství operací a tudíž je tato část náročná na strojový čas. Máme-li tento prvotní výsledek lze výpočet následujícího výsledku zjednodušit. Výsledek se totiž liší v posunu výpočetního okna o jeden vzorek. Tím je možné následující hodnotu získat tak, že za první/poslední sčítanec dosadíme členy o prvek větší. Zbývající dva sčítance obsahující sudé a liché součty je nutné rozlišit na sudé či liché výpočetní okno. Posunuté sudé součty jsou vypočteny odečtením prvního sudého sčítance a připočtením vzorku a dva prvky větší než poslední sudý sčítanec. S lichými sčítanci je situace obdobná. Pro urychlení běhu programu byl výpočet sudého a lichého okna proveden v jednom kroku.

Výsledky jednotlivých integrací jsou ukládány pro výpočet celkové střední hodnoty dané periody. Tomuto účelu slouží funkce **vysledekInt()**, která ještě vyhodnocuje rozptyl předchozí periody, pokud zrovna není počítána první perioda. Vyhodnotit kvalitativní parametr dat je možné pouze po jeho celkovém zpracování. Následuje zvýšení periody a celý proces se opakuje až je výpočet proveden s poslední uvažovanou periodou. Po té je volána funkce **doplnSigma()**, která dopočte poslední kvalitativní parametr a výpočet přechází do funkce **spoctiMiny()**. Funkce **spoctiMiny()** projede všechny vypočtené hodnoty rozptylů a najde tři nejmenší hodnoty, které uloží do proměnných (sigMin1, sigMin2, sigMin3). Vždy od nejmenšího po největší. K těmto hodnotám je uložena odpovídající hodnota velikosti periody (Tmin1, Tmin2, Tmin3). S pomocí vhodně použitých vztahů 3.7 a 3.8 je dopočten výsledek hledané periody T^* a jemu odpovídající rozptyl σ^{2*} . Následuje přepočet výsledné periody na skutečnou frekvenci dle vztahů 3.9 a 3.10.

4.3.3 Výpočet fázoru

Získáním výsledku frekvence algoritmus pokračuje dále a volá funkci **dejFazor()**. Tato funkce obsahuje několik částí. V první fázi je transformováno načtené pole vzorků pomocí interpolačního vzorce. Výsledek je uložen do lokálního pole (newSig), protože výsledky interpolace jsou následně zpracovány a není proto nutné je déle uchovávat. Po té následuje aplikace vztahů 3.16 a 3.17, kde funkce s_k odpovídá proměnné *newSig* a periodické funkce jsou tabelované proměnné *sinus* a *cosinus*. Princip výpočtu integrálu je totožný s postupem uvedeným v předchozí kapitole, tedy použitím numerického integračního vzorce Simpsonova pravidla. Výsledkem je reálná a imaginární část (proměnné *re* a *im*) výsledného fázoru v algebraickém tvaru. Pro úplnost je ještě výsledek převeden do exponenciálního tvaru s proměnnými modulem (*z*) a fází (*fii*). Pro získání fáze ze vztahů 3.21 je použito funkcí *asin()* a *acos()*, jež obsahuje knihovna Math.h. Výsledný úhel je vydán v radiánech a pro názornost následně převeden na stupně.

Vzhledem k tomu, že jsou požadovány výsledky dvou kanálů (proudu a napětí). Je celý uvedený proces výpočtu včetně získání frekvence aplikován jak na data napětí, tak na data odpovídající proudovému vstupu. Po druhém průchodu algoritmu je ještě vypočten rozdílový fázor (**U-I**) a uložen do proměnných *Dz* a *Dfi* pro snazší porovnání výsledků.

Jak již bylo popsáno výše, signál napětí a proudu je načítán postupně. Časový rozdíl mezi těmito vzorky při vzorkovací frekvenci 16 kHz odpovídá prodlevě 62,5 μ s. Dle vztahu 4.5

bylo spočteno, že tento čas odpovídá úhlové chybě 1,125°, což je velmi výrazná chyba a v algoritmu byla provedena korekce. Výsledný úhel proudového signálu je tedy o tento korekční úhel menší.

$$\varphi_{kor} = 62,5 \cdot 10^{-6} \cdot \frac{180}{10 \cdot 10^{-3}} = 1,125^\circ, \quad (4.5)$$

kde $10 \cdot 10^{-3}$ je čas, který odpovídá 180°.

Po celkovém proběhnutí výpočtu jsou k dispozici následující proměnné:

- fázor odpovídající napětovému vstupu v algebraickém tvaru re , im a v exponenciálním tvaru z , fii
- fázor odpovídající proudovému vstupu v alg. tvaru reI , imI a v exp. tvaru zI , fii
- fázor rozdílu ($U-I$) v alg. tvaru Dre , Dim a v exp. tvaru Dz , Dfi

vzhledem k následnému odesílání těchto dat, viz následující kapitola, jsou výsledky převedeny pomocí funkce `dtostrf(hodnota, šířka, des, kam)`. Tato funkce, obsažená v knihovně `stdlib.h`, slouží k převodu proměnné typu `double/float` na jednotlivé digity v ASCII kódu. Uvedené parametry funkce mají následující význam. Proměnná *hodnota* symbolizuje převáděnou veličinu, *šířka* udává počet digitů výsledku⁵, *des* značí počet míst za desetinnou čárkou a *kam* odkazuje na pole, kde bude zapsán výsledek převodu. Počet takovýchto polí je shodný s počtem všech výsledků.

Zde v principu funkce PMU jednotky končí. Každá jednotka však musí obsahovat nějakou formu reprezentace výsledků. Tomuto tématu je věnována následující kapitola.

4.3.4 Reprezentace a kódování výsledků

Nyní je k dispozici velké množství výsledků a tyto hodnoty je nutné nějakým způsobem reprezentovat. Ze všech ohledů vychází nejlépe znakový LCD displej, o kterém bude podrobněji psáno v následující kapitole. Pro praktické využití tohoto zařízení je však vhodný strojově zpracovatelný výstup. Pro tento případ bylo zvoleno odesílání dat po sériové lince USART. Vzhledem k velkému množství dat a také k základnímu zabezpečení přenosu byl vytvořen jednoduchý datový protokol, který bude následně popsán.

Nastavení jednotky USART zajišťují registry `UCSR0A`, `UCSR0A`, `UCSR0A`, `UBRR0H` a `UBRR0L`. Registr `UCSR0A` není v tomto případě nastaven a je celý vynulován.

Registr `UCSR0B` obsahuje parametry:

⁵ včetně desetinného místa a případného znaménka záporné hodnoty

- zapnutí přerušení při přijetí dat (vypnuto)
- zapnutí přerušení při odeslání dat (zapnuto)
- zapnutí přerušení v případě prázdného datového registru (vypnuto)
- zapnutí funkce příjem dat (vypnuto)
- zapnutí funkce odesílání dat (zapnuto)
- nastavení počtu datových bitů (nastaveno 8 bitů)
- zapnutí devátého bitu (vypnuto)

Registr UCSR0C obsahuje parametry:

- mód jednotky USART (nastaven asynchronní USART)
- nastavení paritního bitu (vypnuto)
- délka stop bitu (1 bit)
- polarita hodinového signálu (v tomto případě nevyužito)

Registry UBRR0H a UBRR0L obsahují parametry rychlosti datového přenosu. Pro tento případ byla zvolena standardní rychlost 9600 b/s, která pro krystal 20MHz dosahuje dle katalogu chyby 0,2%. Tato chyba má větší důležitost pokud mikrokontrolér propojujeme např. s počítačem, kde je přenosová rychlost daleko přesnější. Vlivem asynchronnosti přenosu by čas od času docházelo k chybně přeneseným datům. Tyto nekorektní data jsou však v přijímači eliminována. V testovaném případě mají vysílač i přijímač chybu stejnou a proto k tomuto jevu téměř nedochází.

Použitý protokol vychází ze zdroje [12], je upraven dle požadavků aplikace. Délka protokolu je jedenáct bajtů. Všechny obsažené hodnoty jsou vyjádřeny ve formě ASCII kódu. Význam jednotlivých bajtů zobrazuje tab. 2. Prvním odesílaným bajtem je START bajt, který obsahuje znak '*'. Po té je odesílán TYP proměnné, který signalizuje jaká součást fázoru je momentálně odesílána. Součástí fázoru jsou reprezentovány znaky 'r'-reálná část výsledku, 'i'-imaginární část, 'f'-vypočtená frekvence, 's'-rozptyl z výpočtu frekvence, 'm'-modul exponenciálního tvaru výsledku, 'a'-vypočtený fázový úhel. Následuje prázdný bajt REZ s hodnotou '0', jenž slouží jako rezerva pro případné využití. Bajt DRUH obsahuje informaci pro jaký typ měřené veličiny data slouží. Může nabývat následujících hodnot: '0'-data patří napětíovému vstupu, '1'-data náleží proudovému vstupu a '2'-data mají význam rozdílového vektoru napětí a proudu. Po té následuje šest bajtů, které nesou informaci o konkrétní výsledné hodnotě. V závěru je odeslán jeden bajt (též v ASCII kódu) sloužící jako kontrolní součet ASCII hodnot bajtů 1 až 9. Proměnná pro výpočet kontrolního součtu je však typu *unsigned char* a tudíž

se může pohybovat pouze v rozsahu 0-255. Z tohoto důvodu může dojít k situaci, kdy součet bude větší než 255 a dojde k přetečení⁶. S tímto jevem je nutné počítat při dekódování přijatých dat.

Algoritmus kódování a odesílání dat je synchronizován přerušením od časovače 0. Perioda přerušení je nastavena přibližně na 2 ms. Nastavení časovače je podobné, jako již popsané v kapitole 4.3.1. Po přerušení narazí program na testovací podmínku, zda-li došlo k odeslání předchozího bajtu. Pokud ano, je volána funkce **naplnData()**, která naplní datový protokol a v jeho závěru dopočte kontrolní součet. Nyní je odesílaný bajt uložen do registru UDR0, čímž dojde automaticky k odeslání po sériové lince. Po dokončení přenosu generuje jednotka USART přerušení při kterém je posunut ukazatel odesílaného bajtu na bajt následující a algoritmus čeká na vznik přerušení od časovače 0.

Protokol je při každé generaci přerušení plněn daty postupně tak, aby se přenesly všechny vypočtené výsledky. Přenesení kompletní informace tedy trvá téměř 400 ms. Při každém přerušení je odeslán jeden bajt a pro jednu informaci je bajtů jedenáct, to je 22 ms na jeden odeslaný protokol. Protokol nabývá šesti typů a tří druhů, což je osmnáct kombinací. Časování je samozřejmě možné ještě zrychlit, ale pro tento typ aplikace je doba přenosu dostačující.

Tab. 2 Forma datového protokolu

bajt	0	1	2	3	4	5	6	7	8	9	10
obsah	'*'										
význam	START	TYP	REZ	DRUH	DATA	DATA	DATA	DATA	DATA	DATA	Kontrolní součet

Tab. 3 Kódování typu informace

funkce	START	TYP	REZ	DRUH	DATA	DATA	DATA	DATA	DATA	DATA	Kontrolní součet
Reálná část	'*'	'r'	'0'	'0'	' '	' '	' '	' '	' '	' '	' '
Imaginární část	'*'	'i'	'0'	'0'	' '	' '	' '	' '	' '	' '	' '
Frekvence	'*'	'f'	'0'	'0'	' '	' '	' '	' '	' '	' '	' '
Rozptyl	'*'	's'	'0'	'0'	' '	' '	' '	' '	' '	' '	' '
Modul	'*'	'm'	'0'	'0'	' '	' '	' '	' '	' '	' '	' '
Fáze	'*'	'a'	'0'	'0'	' '	' '	' '	' '	' '	' '	' '

Tab. 4 Kódování druhu informace pro typ reálného čísla

funkce	START	TYP	REZ	DRUH	DATA	DATA	DATA	DATA	DATA	DATA	Kontrolní
--------	-------	-----	-----	------	------	------	------	------	------	------	-----------

⁶ např. 255+1=0

											součet
napětí	'*'	'r'	'0'	'0'	' '	' '	' '	' '	' '	' '	' '
proudu	'*'	'r'	'0'	'1'	' '	' '	' '	' '	' '	' '	' '
rozdíl (U-I)	'*'	'r'	'0'	'2'	' '	' '	' '	' '	' '	' '	' '

Na jednotce PMU není zařazen LCD displej, i když by se tato varianta nabízela. Důvodem je použitá knihovna pro ovládání displeje, která obousměrně komunikuje s displejem. Toto chování sice samo o sobě není na závadu až do okamžiku, kdy je nutné přesně časovat program v ladícím módu. Program se zacyklí při čekání na zpětnou informaci a nelze dál pokračovat. Ten samý důvod však nepůsobí problémy při ladění programu. V případě naprogramování a spuštění běhu programu v aplikaci se celý běh programu zasekne dojde-li k odpojení displeje od mikrokontroléru. Tento jev je nepříjemný a proto byl displej přesunut do externího obvodu, který plní funkci zobrazovací jednotky.

4.3.5 Dekódování a výpis na displej

Proces dekodování a výpis na LCD displej je realizován v druhém mikrokontroléru. Pro tuto aplikaci byl vytvořen jednoduchý program, který čeká dokud není na sériové lince zaznamenáno vysílání dat. Pokud ano, zachytí tuto informaci, otestuje ji a pokud je platná vypíše obsah na displej. Program obsahuje dva zdroje přerušení (časovač 0 a jednotku USART) a dekodovací funkci.

Nastavení jednotlivých řídicích registrů je podobné, jako u PMU jednotky.

Registr UCSR0B USART jednotky obsahuje parametry:

- zapnutí přerušení při přijetí dat (zapnuto)
- zapnutí přerušení při odeslání dat (vypnuto)
- zapnutí přerušení v případě prázdného datového registru (vypnuto)
- zapnutí funkce příjem dat (zapnuto)
- zapnutí funkce odesílání dat (vypnuto)
- nastavení počtu datových bitů (nastaveno 8 bitů)
- zapnutí devátého bitu (vypnuto)

Nastavení registru UCSR0C je shodné jako v předchozí kapitole.

Registr TCCR0 časovače 0 je nastaven v režimu volného běhu s přerušením při přetečení. Dělicí poměr předděličky je nastaven na 1024. Do registr TCNT0 je uložena hodnota 100, která doladuje periodu přerušení na přibližně 10 ms.

Funkce programu je následující. Po zaznamenání signálu na sériové lince je vyvoláno přerušení od jednotky USART. Následně je přepsán přijatý bajt do pomocné proměnné a otestován, zda obsahuje znak '*'. Pokud ano, jedná se o počáteční start bajt datového protokolu a znak je uložen do proměnné *datovaSekvence* na první místo. Po uložení je posunut ukazatel na další pozici v datové sekvenci. Pokud START bajt neobsahuje, nastávají dvě možnosti. První možnost je, že v protokolu ještě není uložen start bajt, nejedná se tedy o legitimní data a program na ně nereaguje. V případě, že již v protokolu je obsažen znak '*' dá se předpokládat, že bajt patří do obsahu protokolu a je uložen na místo ukazatele, který se opět posune o jednu pozici. Po naplnění celé proměnné *datovaSekvence* (11 bajtů) je volána funkce **otestujAVypis()**.

Funkce **otestujAVypis()** nejprve spočte kontrolní součet z přijatého protokolu a porovná ho s posledním bajtem v protokolu. Pokud se tyto hodnoty shodují, je přistoupeno k dekódování obsažené informace. Podle typu a druhu dat se roztřídí datový obsah do odpovídajících proměnných a aktivuje se povolení pro výpis na displej.

Výpis na displej je řízen od přerušení časovače 0. Perioda přerušení časovače je sice 10ms, to je však na displej příliš krátká doba. Výrazněji delší periodu přerušení není možné nastavit, a proto byla vytvořena programová smyčka, která tuto periodu desetkrát prodlouží. Obnovovací perioda displeje je tedy přibližně 100 ms. Po uplynutí této doby jsou postupně aplikovány instrukce pro práci s displejem, které udávají grafický vzhled reprezentovaných výsledků. Instrukce pro práci s displejem obsahuje knihovna *lcd.h*. tato knihovna není standardně implementována do editoru AVR Studio 5, ale je nutné ji dodatečně přidat. Uvedená knihovna je dostupná ze zdroje [13].

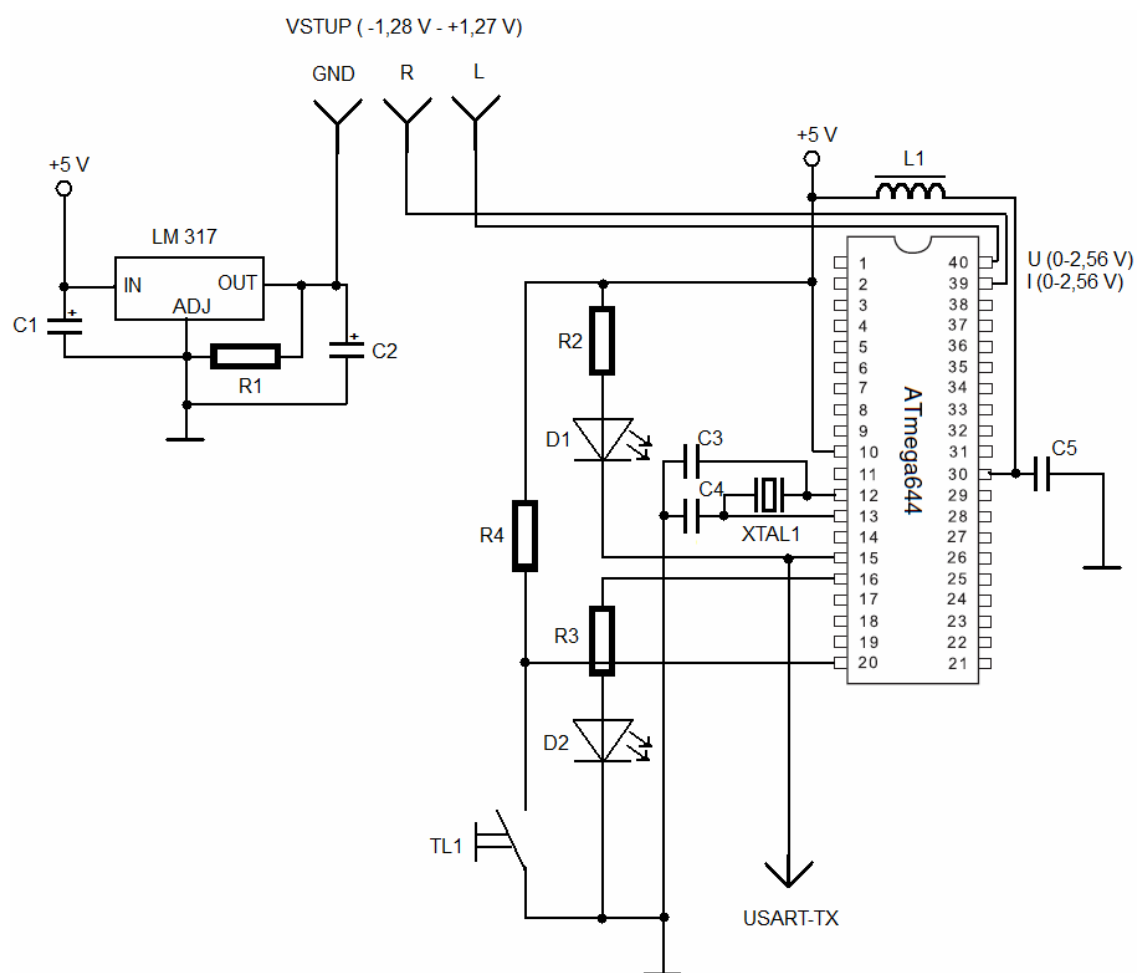
Vzhledem k velkému množství dat byl zvolen displej se čtyřmi řádky po šestnácti znacích. Konkrétní typ displeje, který byl použit je MC1604CH-SYL. Grafická úprava bude předvedena v kapitole Ověření funkčnosti.

4.4 Hardwarová realizace

Výše popsané zařízení bylo prakticky vytvořeno na nepájovém kontaktním poli. Skládá se ze tří částí. Jedná se o měřící jednotku PMU, zobrazovací jednotku a pomocný převodník

napětových úrovní. Vzhledem k testovací funkci tohoto obvodu obsahuje zapojení pouze nezbytně nutné obvody.

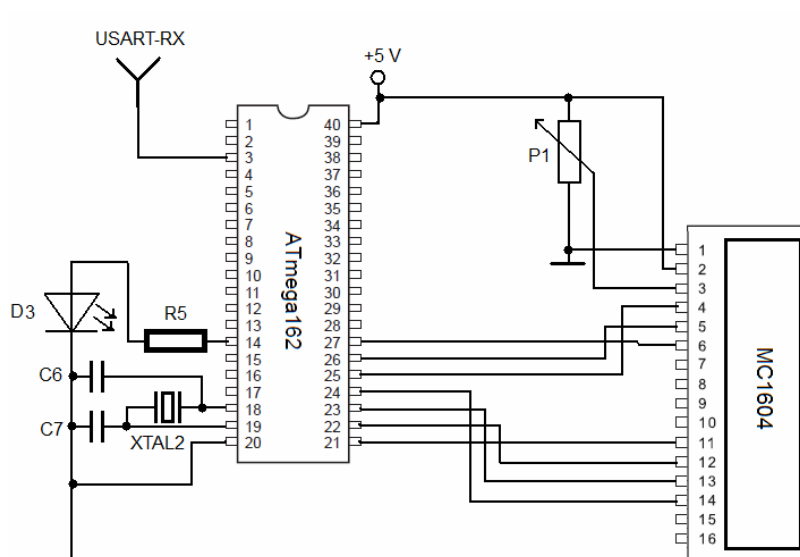
Zapojení jednotky PMU je zobrazeno na obr. 4.6. Pro úplnost je obvod doplněn i převodníkem napětových úrovní, protože ve většině případů bez něj nelze pracovat. Vstupními objekty, jsou dva signálové vstupy vedoucí na piny A/D převodníku a tlačítko sloužící k zahájení výpočtu. Výstupy jsou dvě signalizační LED diody a pin sériového portu. LED dioda D1 slouží k signalizaci vysílání sériového signálu a dioda D2 je rozsvícena po dobu, kdy probíhá výpočet. Výpočet je spouštěn při detekci sestupné hrany tlačítka, po té je funkce tlačítka blokována po dobu 1s a následně je tlačítko odblokováno. Pokud je tlačítko soustavně stisknuto výpočet je realizován po každém odblokování, tj. po 1s.



Obr. 4.6 Schéma jednotky PMU

Obvod zobrazovací jednotky je zobrazen na obr. 4.7. Jeho vstupem je signál sériové linky a výstupem je informace vypsaná na displeji. Obvod obsahuje ještě signalizační LED D3.

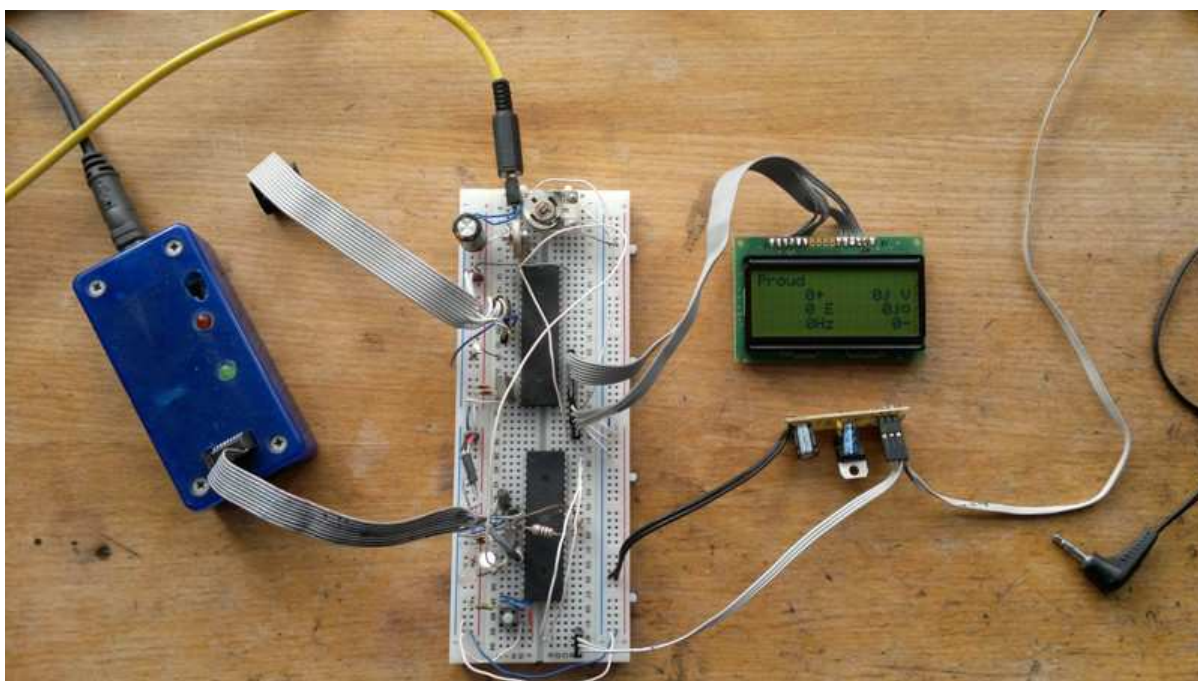
Tato dioda blikne po každém úspěšném dekódování přijatého protokolu. Seznam použitých součástek je uveden v tištěné příloze č. 4. Nutno dodat, že ke správné funkci přenosu po sériové lince musí být propojeny záporné napájecí póly PMU a zobrazovací jednotky.



Obr. 4.7 Schéma zobrazovací jednotky

5 Ověření funkčnosti

Tato kapitola se zabývá ověřením funkce a otestováním vytvořeného zařízení simulovanými průběhy měřených veličin. Na obr. 5.1 je zobrazeno celé zařízení. V levé části se nachází programátor, uprostřed je na nepájivém poli zapojena jednotka PMU i zobrazovací jednotka a v pravé části obrázku se nachází LCD zobrazovač a pod ním převodník napět'ových úrovní. V pravém dolním rohu je konektor pro připojení do zvukové karty počítače.



Obr. 5.1 Celé zařízení na nepájivém kontaktním poli

5.1 Softwarový test

Ověření funkčnosti programu bylo nejprve odzkoušeno v AVR Studiu, kde místo reálného signálu byly do proměnné uloženy hodnoty odpovídající reálnému signálu. Příklad testovaného signálu je zobrazený na obr. 5.2. Parametry testovaného signálu charakterizuje následující rovnice:

$$U = 127,5 + 128 \cdot (0,8 \cdot \sin(\omega \cdot t + \varphi) + 0,06 \cdot \sin(5 \cdot \omega \cdot t + \varphi) + 0,03 \cdot \sin(7 \cdot \omega \cdot t + \varphi) + 0,05 \cdot \sin(17 \cdot \omega \cdot t + \varphi) + 0,05 \cdot \sin(29 \cdot \omega \cdot t + \varphi))$$

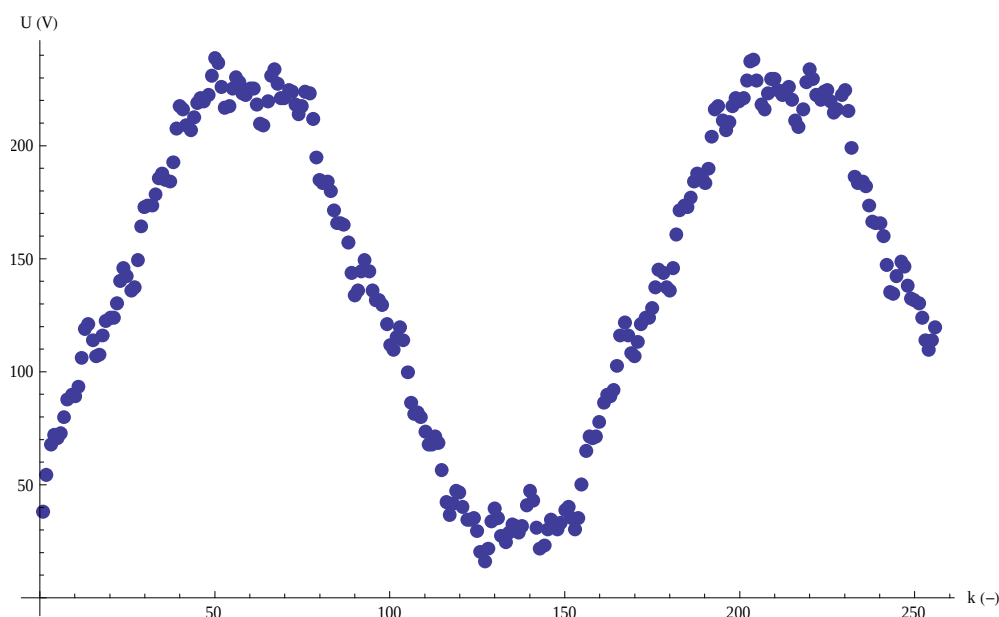
kde $\omega=2\pi.52$ a $\varphi= -45^\circ$. Následující obrázek 5.3 ukazuje výpis výsledku na displej. První řádek popisuje jaký typ dat je zobrazován, druhý řádek zobrazuje algebraický tvar výsledku ve formě $a+b.j$. Na třetím řádku je tento tvar převeden do exponenciální formy ve tvaru $|Z|.E^{+ \varphi.j}$. Poslední řádek zobrazuje údaj o naměřené frekvenci a jemu odpovídající výsledek rozptylu, který odpovídá věrohodnosti výsledku frekvence. Tento signál byl komponován tak, aby postihl nejčastější a nejsilnější vyšší harmonické v síti (5. a 7.) a zároveň jsou obsaženy velmi vysoké harmonické (17. a 29.) mající funkci šumu obsaženého v signálu. V tomto případě lze velmi snadno dle vztahu 6.1 spočítat celkové harmonické zkreslení (THD) měřeného průběhu.

$$THD = \frac{\sqrt{U_2^2 + U_3^2 + \dots + U_n^2}}{U_1} \quad (6.1)$$

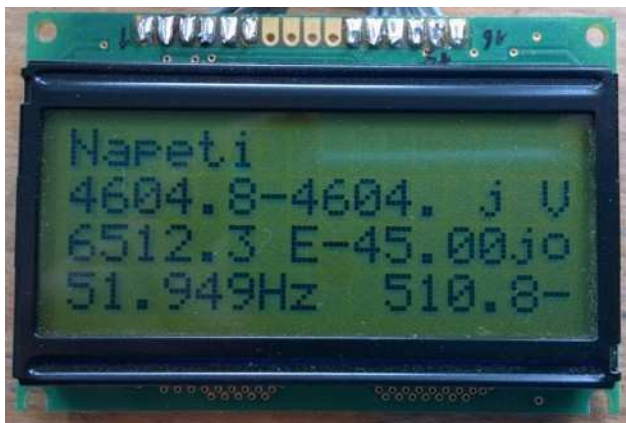
Spočteme-li si výsledné harmonické zkreslení tohoto signálu

$$THD = \frac{\sqrt{0,06^2 + 0,03^2 + 0,05^2 + 0,05^2}}{0,8} \cdot 100 = 12,18\%,$$

což signalizuje velmi nízkou kvalitu signálu, která by se v běžné síti vyskytovat určitě neměla.



Obr. 5.2 Softwarový testovací signál



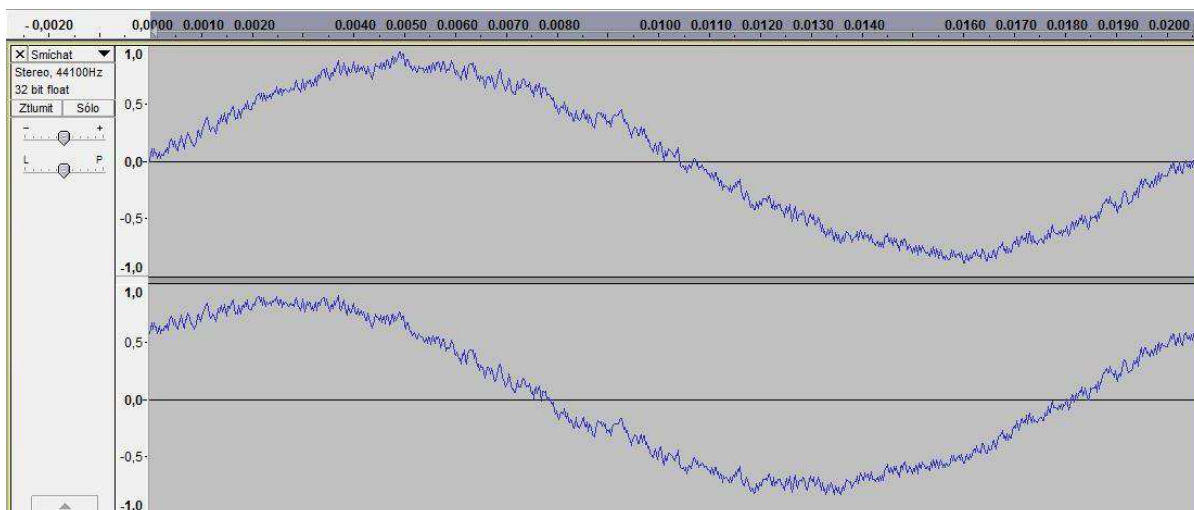
Obr. 5.3 Výsledek softwarového testovacího signálu

I přes takto rušený signál dává metoda uspokojivé výsledky viz obr. 5.3, které se příliš neliší od teoretického předpokladu. Čím více je však signál rušný, tím narůstá hodnota rozptylu vypočtené frekvence. Tento výstup můžeme porovnat s dalšími softwarově testovanými signály uvedenými v tištěné příloze č. 3.

5.2 Hardwarový test

Hardwarový test byl proveden přímo na zařízení. Zdrojem signálu byl vygenerovaný průběh ze zvukové karty počítače, který byl vytvořen v programu Audacity 2.0.5. Tento program umožňuje vytvářet sinusové průběhy, fázově posouvat signály a uměle přidávat šumový signál. Je zde také možnost přidávat další průběhy např. vyšších harmonických, které jsou superponovány na původní signál.

Zařízení bylo otestováno množstvím různých frekvencí, vzájemných fázových posuvů bez šumu či s přidaným šumem. Příklad testovacího signálu je na obr. 5.4. Napětíový vstup obsahuje funkci sinus s modulací šumu, průběh simulující proud je posunut o -45° a je taktéž zašumělý. Frekvence obou signálů je zvolena 48,461 Hz. Výsledky jsou zobrazeny na obrázcích 5.5 (napětí), 5.6 (proud) a 5.7 (dopočtený rozdíl). Proud je označen pro odpovídající hodnotu proudového vstupu, která je úměrná měřenému napětí proto je vyjádřena ve voltech. Rozdílový signál (U-I) není korektní vyjádření, je zde zařazen zejména proto, aby bylo hned jasné, jaký je reálný fázový posuv mezi oběma signály.



Obr. 5.4 Testovaný signál včetně šumu



Obr. 5.5 Výsledek napětí



Obr. 5.6 Výsledek proudu



Obr. 5.7 Výsledek rozdílu

Poslední hodnotící fází byla délka výpočtu. S ohledem na rychlost výpočtu bylo voleno menší množství načítaných dat. To se samozřejmě projevuje v přesnosti výsledků. Pro co možná nejrychlejší výpočty byly v programu použity především celočíselné proměnné s pevnou řádovou čárkou (typ char, int, long int). Tyto proměnné jsou zejména využity v algoritmu výpočtu frekvence. V další fázi programu (získání fázoru) však již nebylo možné celočíselné proměnné využít, zejména pak z důvodu použití interpolačního vzorce. Proto byly použity proměnné typu float s pohyblivou řádovou čárkou. Tento fakt zapříčinil velmi výrazné zpomalení programu. Orientační časy k odpovídajícím operacím jsou uvedeny v následující tabulce. Časy jsou uvedeny pouze pro jeden zpracovaný průběh. Reálná délka výpočtu je složena ze součtu doby načítání dat a dvojnásobku času výpočtu, tedy $32+2 \cdot 196,28=424,56$ ms.

Tab. 5 Trvání běhu programu pro jednu zaznamenanou veličinu

Operace	časová náročnost (ms)
Načítání dat	32
Výpočet frekvence	43,49
Interpolace	85,29
Výpočet fázoru	67,5
Celkový čas výpočtu	196,28
Celkový čas procesu	228,28

6 Závěr

Tato práce byla zaměřena na problematiku synchronního měření v elektrických sítích používaného zejména v přenosové soustavě. Jsou zde uvedeny hlavní důvody využití synchronního měření pro získání údajů o okamžitém stavu sítě. Princip zpracování těchto získaných parametrů systémem WAMS, který tyto data vyhodnocuje a na jejich základě předvídá neobvyklé jevy a chování v síti. Tímto je možné včas přizpůsobit konfiguraci sítě aby byla zajištěna požadovaná kvalita síťových parametrů a stabilita provozu sítě.

Následující kapitola je věnována principu získání fázoru z měřeného signálu. Byl zvolen princip algoritmu využívající statistických operací s naměřenými daty. V kapitolách je popsán proces získání frekvence z navzorkovaných dat včetně kvalitativního parametru rozptylu a proces získání fázoru, který je spolu s frekvencí výstupním produktem této metody.

Cílem praktické části práce bylo naprogramovat popsany algoritmus do zvoleného typu mikrokontroléru a ověřit jeho funkčnost. Vzhledem ke zkušenostem byl zvolen mikrokontrolér firmy ATMEL, řady ATmega. Z této řady byl vybrán jeden z nejvýkonnějších typů, konkrétně typ ATmega644. Programovacím prostředím bylo AVR Studio 5, které obsahuje velké množství funkcí zjednodušujících proces vývoje algoritmu. Programovacím jazykem byl jazyk C.

Simulace funkce algoritmu byla složena ze dvou částí. První část, uvedená pod názvem softwarová simulace, simulovala vlastní funkčnost algoritmu přímo ve vývojovém prostředí. Pro tento případ byl do programu přidán simulující průběh odpovídající reálnému signálu.

Druhá část simulace, hardwarová simulace, byla realizována přímo ve fyzickém zařízení (PMU jednotka) a zahrnovala načítání dat pomocí A/D převodníku. Testovaný signál byl vytvořen zvukovou kartou počítače. Přestože zvolený mikrokontrolér obsahuje převodník s rozlišením 10 bitů, pro simulaci bylo využito horních 8 bitů. Takto „hrubé“ rozlišení bylo zvoleno, protože s použitou vzorkovací periodou není možné dosáhnout plného rozlišení. Zároveň takto nízké rozlišení slouží jako demonstrace odolnosti zvolené metody vzhledem ke kvalitě naměřených dat. Vzorkovací perioda byla zvolena tak, aby při jmenovité frekvenci měřeného signálu obsahovala perioda 160 vzorků. Tento počet vzorků se při simulaci ukazuje jako nedostatečný, avšak jeho zvýšení by znamenalo použití externího A/D převodníku a komplexní úpravu vytvořeného programu.

Interpretace dat je realizována na LCD displeji. Z tohoto důvodu byla vytvořena pomocná zobrazovací jednotka, která přijímá kódované vypočtené výsledky přes sériovou linku

od PMU jednotky. Použitím sériového rozhraní se otvírá možnost nejen naměřené výsledky číst na displeji, ale také je dále elektronicky zpracovávat. Na displeji je zobrazena informace k jakému vstupu patří výsledné hodnoty, konkrétní hodnoty fázoru v algebraickém i exponenciálním tvaru a údaj o frekvenci včetně hodnoty rozptylu signalizující věrohodnost výsledku.

Na displeji přeskakují výsledky odpovídající napěťovému a proudovému vstupu a jejich rozdíl mající pouze informativní charakter. Každý údaj je na displeji zobrazen po dobu přibližně 4s. a cyklicky se opakuje. Navzdory malému počtu načtených dat se naměřené vzájemné fázové posuvy velmi blíží těm teoretickým. Tím se nabízí možnost realizace byť jen ukázkového měření charakteru zátěže (z napětí a proudu), či sledu fází (dvou napětí).

Celé zařízení bylo zrealizováno na nepájivém kontaktním poli. Vstupní signál by se měl pohybovat v rozsahu $\pm 1,27$ V s frekvencí 50 ± 3 Hz.

Použitá metoda byla shledána jako velmi zajímavá alternativa k již používaným zařízením, jenž by mohla dosahovat požadovaných přesností měřených parametrů a současným zachováním nízkých výrobních nákladů, neboť je zde využito nižší úrovně mikrokontrolérů. Tím by aplikace synchronního měření dosáhla většího významu i v podružných sítích distribuční soustavy.

7 Seznam použité literatury

- [1] IEEE standard c.37.118.1 2011. IEEE Standard for Synchrophasor Measurements for Power Systems. USA: IEEE Power & Energy Society, 2011.
- [2] AIS SPOL. S R.O. Měřicí a informační technologie pro energetiku česky english [online]. [2011] [cit. 2014-05-04]. Dostupné z: <http://www.ais-brno.cz/?lang=cz>
- [3] Problémy estimace stavu ES. Česká energetika [online]. 23. září 2008 [cit. 2014-05-04]. Dostupné z: http://www.ceskaenergetika.cz/nezarazene_clanky/problemy_estimace_stavu_es.html
- [4] WAMS systémy pro monitoring elektrizační soustavy. *WAMS systémy pro monitoring elektrizační soustavy* [online]. 03/2013, roč. 7, č. 3 [cit. 2014-05-05]. Dostupné z: <http://www.allforpower.cz/clanek/wams-systemy-pro-monitoring-elektrizacni-soustavy/>
- [5] Kyncl, J.; Hariram, A.; Novotny, M., "On measurement of synchronous phasors in electrical grids," Circuits and Systems (ISCAS), 2013 IEEE International Symposium on , vol., no., pp.2972,2975, 19-23 May 2013
- [6] Materiály poskytnuté katedrou K13115
- [7] USBasp. LoetLabor [online]. 24. Mai 2013 [cit. 2014-05-04]. Dostupné z: <http://www.loetlabor.org/USBasp>
- [8] USBasp - USB programmer for Atmel AVR controllers. Thomas Fischl [online]. © 1998 - 2014 [cit. 2014-05-04]. Dostupné z: <http://www.fischl.de/usbasp/>
- [9] Uživatelský manuál procesoru ATMEGA644-20PU, ATMEL;

- [10] Uživatelský manuál procesoru ATmega 162-16PU, ATMEL;
- [11] SLEZSKÁ UNIVERZITA V OPAVĚ MATEMATICKÝ ÚSTAV V OPAVĚ.
Numerické metody – studijní opora [online]. © 2011 [cit. 2014-05-04]. Dostupné z:
<http://www.slu.cz/math/cz/knihovna/ucebni-texty/Numericke-metody/Numericke-metody.pdf>
- [12] HUMLÍČEK, Jiří. Řízení otáček stejnosměrného stroje procesorem z řady PIC18F. Praha, 2012. Bakalářská práce (Bc.). České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra elektrických pohonů a trakce. Vedoucí práce Miroslav Lev.
- [13] AVR-GCC Source Examples. Peter Fleury [online]. © 2006 [cit. 2014-05-04].
Dostupné z: <http://homepage.hispeed.ch/peterfleury/avr-software.html#libs>
- [14] Uživatelský manuál LCD displej MC1604CH-SYL, Everbouquet;
- [15] VÁŇA, Vladimír. *Mikrokontroléry ATMEL AVR: programování v jazyce C : popis a práce ve vývojovém prostředí CodeVisionAVR C*. 1. vyd. Praha: BEN - technická literatura, 2003, 215 s. ISBN 80-730-0102-0.

8 Seznam použitých zkratek

UTC - Coordinated Universal Time

PMU - Phasor Measurement Unit

PDC - Phasor Data Concentrator

WAMS - Wide Area Monitoring System

SCADA - Supervisory control and data acquisition

FACTS - Flexible alternating current transmission system

ES - Elektrická soustava

GPS - Global Positioning System

USB - Universal Serial Bus

ISP - Serial Peripheral Interface

JTAG - Joint Test Action Group

USART - Universal Synchronous / Asynchronous Receiver and Transmitter

ASCII - American Standard Code for Information Interchange

THD - Total harmonic distortion

9 Seznam obrázků

Obr. 2.1 Značení synchronních fázorů; zdroj [1].....	12 -
Obr. 2.2 Průnik jmenovité frekvence a signálu s $f > f_0$; zdroj [1].....	13 -
Obr. 2.3 Vzorky odebrané při odlišné frekvenci; zdroj [1].....	13 -
Obr. 2.4 Tok dat při estimaci stavu sítě; zdroj [2]	15 -
Obr. 2.5 Schéma pro vyhodnocování provozu ES; zdroj [3].....	16 -
Obr. 2.6 Princip monitorování stability sítě; zdroj [2].....	17 -
Obr. 2.7 Náhradní schéma vedení pomocí π -článku; zdroj [2].....	18 -
Obr. 2.8 Grafický výstup fázorů napětí; zdroj [2]	19 -
Obr. 2.9 Systém WAMS; zdroj [4]	20 -
Obr. 3.1 Vliv volby kper na velikosti střední kvadratické odchylky; zdroj [5]	23 -
Obr. 3.2 Spojitá funkce sinus ($f=50\text{Hz}$; $k_{\text{per}}=1,6$)	24 -
Obr. 3.3 Diskretizovaná funkce sinus ($f=50\text{Hz}$; $k_{\text{per}}=1,6$; $N=128$ vzorků)	25 -
Obr. 3.4 Posun výpočetního okna pro jednu konkrétní periodu	26 -
Obr. 3.5 Výsledky středních hodnot signálu pro různě zvolené periody $T_1, T_2, \dots, T_n$..	27 -
Obr. 3.6 Příklad souboru dat při nevhodně zvoleném rozsahu hledaných period T_1-T_n ...	27 -
Obr. 3.7 Zobrazení vypočtených dvojic zvolené frekvence a odpovídající rozptyl	28 -
Obr. 3.8 Transformace vstupních dat interpolací.....	30 -
Obr. 4.1 USBasp programátor s rozhraním ISP	33 -
Obr. 4.2 Zapojení pinů USBasp programátoru; zdroj [7]	33 -
Obr. 4.3 Konfigurace pinů pro ATmega 162	34 -
Obr. 4.4 Konfigurace pinů pro ATmega 644	35 -
Obr. 4.5 Princip převodníku napěťové úrovně	36 -
Obr. 4.6 Schéma jednotky PMU	46 -
Obr. 4.7 Schéma zobrazovací jednotky	47 -
Obr. 5.1 Celé zařízení na nepájivém kontaktním poli	48 -
Obr. 5.2 Softwarový testovací signál	49 -
Obr. 5.3 Výsledek softwarového testovacího signálu.....	50 -
Obr. 5.4 Testovaný signál včetně šum	51 -
Obr. 5.5 Výsledek napětí	51 -
Obr. 5.6 Výsledek proudu	51 -
Obr. 5.7 Výsledek rozdílu	51 -

10 Seznam tabulek

Tab. 1 Konverzní čas A/D převodníku pro různé nastavení; zdroj [9]	- 38 -
Tab. 2 Forma datového protokolu.....	- 43 -
Tab. 3 Kódování typu informace	- 43 -
Tab. 4 Kódování druhu informace pro typ reálného čísla	- 43 -
Tab. 5 Trvání běhu programu pro jednu zaznamenanou veličinu	- 52 -

11 Seznam příloh

Příloha 1 - Náhled vývojového prostředí - editor*

Příloha 2 - Náhled vývojového prostředí - simulátor*

Příloha 3 - Doplnující výsledky testování*

Příloha 4 - Seznam použitých komponent zapojení*

Příloha 5 - Projekt DP_Synchrofazory – jazyk C, AVR studio 5

Příloha 6 – Katalogový list k mikroprocesoru ATMEL ATmega644-20PU

Příloha 7 - Katalogový list k mikroprocesoru ATMEL ATmega162-16PU

Příloha 8 – Knihovna pro zobrazení na LCD displej

Příloha 9 - Katalogový list k LCD displeji MC1604CH-SYL

* Obsaženo i v tištěné verzi této práce.

12 Obsah příloženého CD

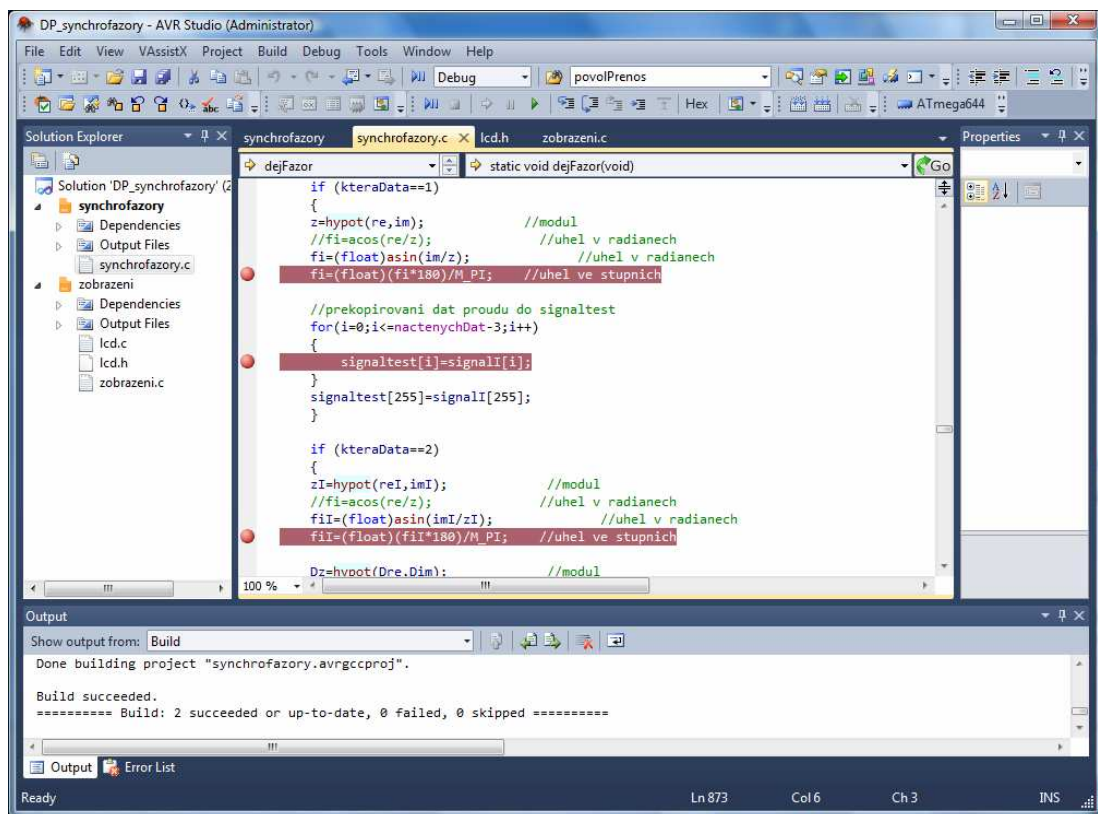
Složka **Přílohy** - obsahuje všechny výše uvedené přílohy

Složka **Obrázky** - obsahuje všechny použité obrázky s příslušným číslováním

Složka **Dokumenty** - obsahuje elektronickou podobu této práce ve formátu .pdf

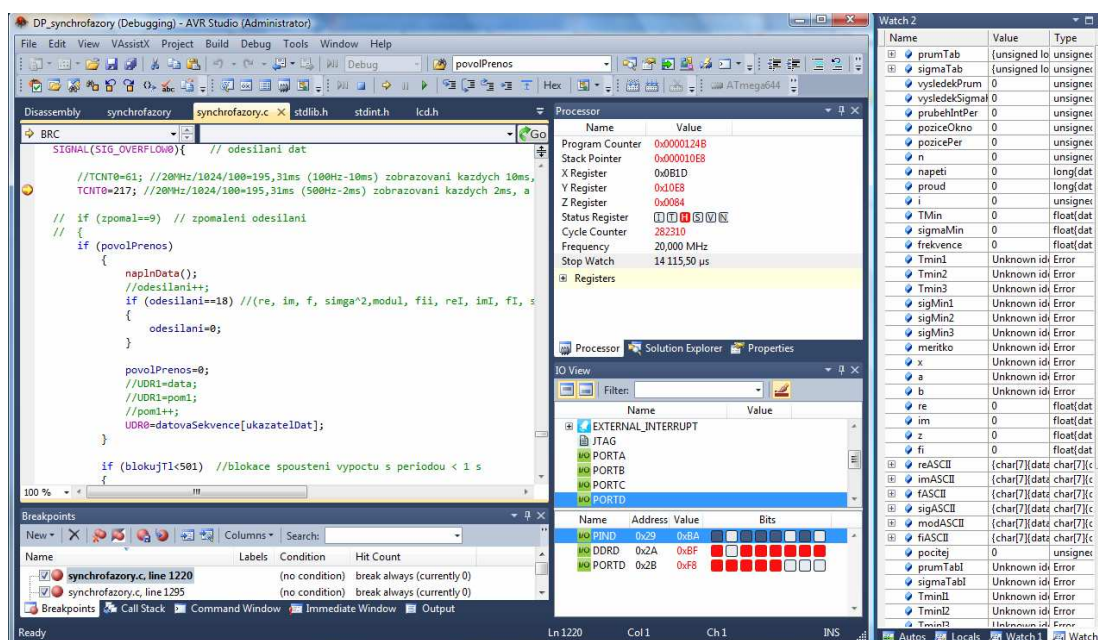
13 Tištěné přílohy

Příloha 1 Náhled vývojového prostředí - editor



Obr. P 1 Vývojové prostředí AVR Studio 5, programovací mód

Příloha 2 Náhled vývojového prostředí - simulátor



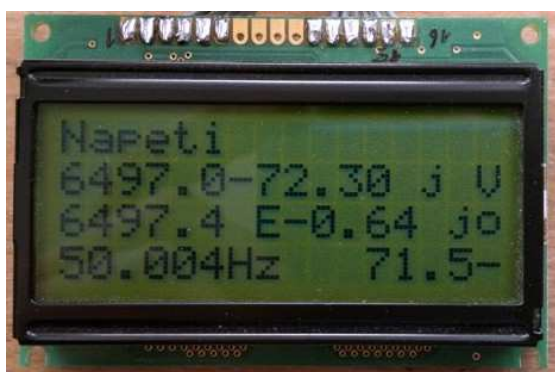
Obr. P 2 Vývojové prostředí AVR Studio 5, ladící mód

Příloha 3 Doplnující výsledky testování

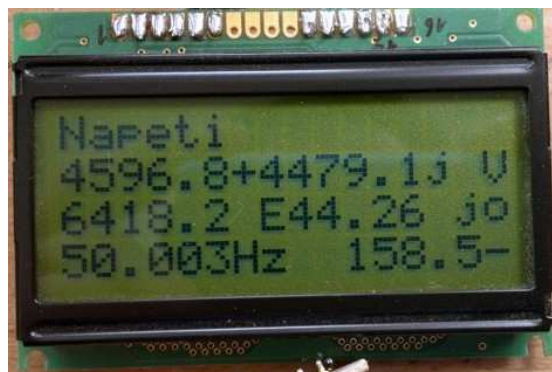
Softwarový test

Tabulka popisuje signál, jehož výsledky jsou uvedeny na následujících obrázcích s příslušným označením.

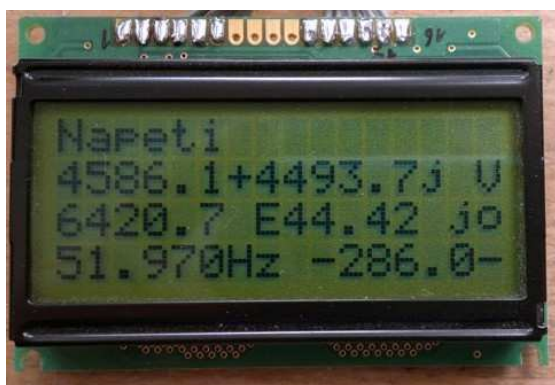
Softwarový test				
označení	amplituda*255 (-)	f (Hz)	φ (°)	doplňující informace
a)	0,8	50	0	sinusový signál bez fázového posuvu
b)	0,8	50	45	sinusový signál s posunutím o úhel φ
c)	0,8	52	45	sinusový signál s posunutím pro odlišnou frekvenci
d)	0,8	52	-45	sinusový signál s posunutím pro odlišnou frekvenci s obsahem vyšších harmonických, viz kapitola 5.1



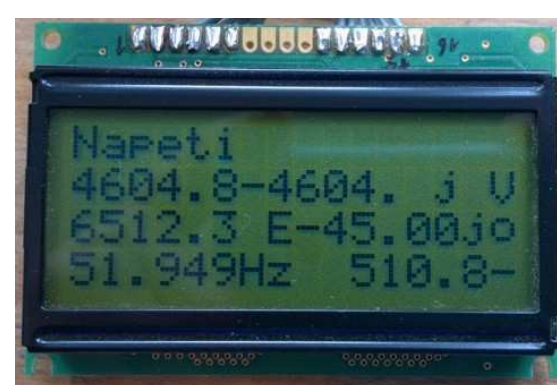
a)



b)



c)



d)

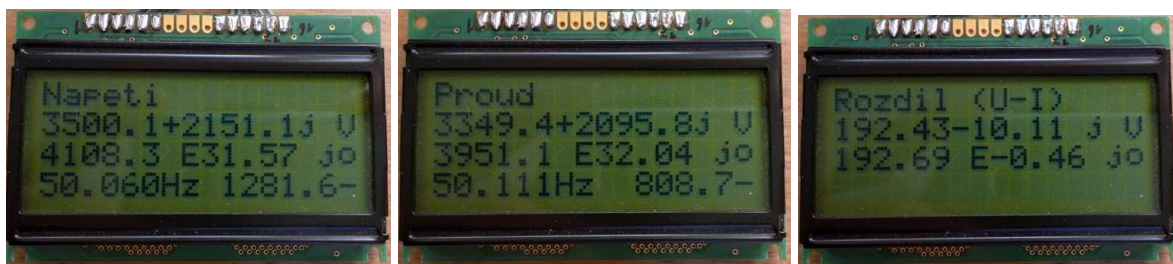
Hardwarový test

Tabulka popisuje signál generovaný zvukovou kartou počítače, jehož fázový posuv mezi levým (odpovídající napětí) a pravým (odpovídající proudu) kanálem charakterizuje úhel φ . Pod tabulkou jsou umístěny vyobrazené výsledky k odpovídajícím signálům.

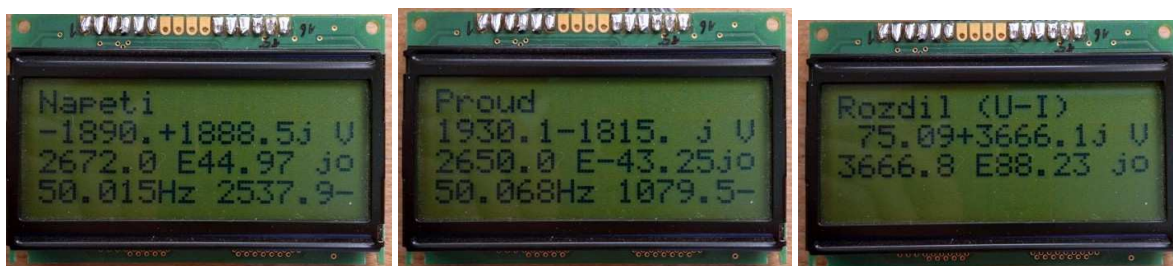
Hardwarový test			
označení	f (Hz)	φ (°)	doplňující informace
a)	50	0	sinusové signály bez vzájemného fázového posunu
b)	50	0	sinusové signály bez vzájemného fázového posunu s namodulovaným šumem
c)	50	90	sinusové signály simulující čistou indukční zátěž
d)	50	-90	sinusové signály simulující kapacitní zátěž s namodulovaným šumem
e)	48,461	45	sinusové signály s fázovým posunem a namodulovaným šumem



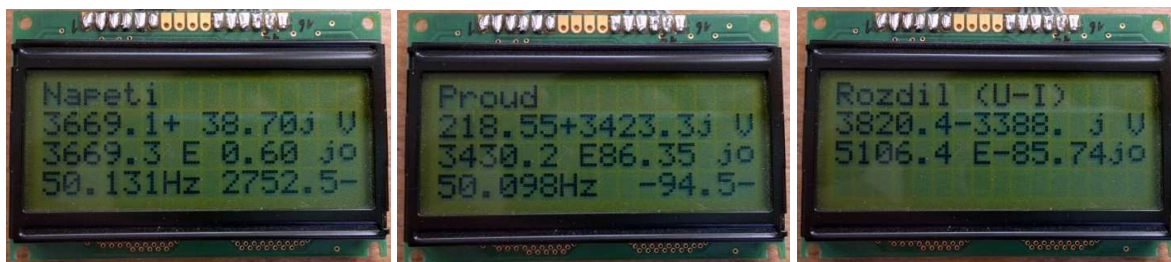
a)



b)



c)



d)



e)

Příloha 4 - Seznam použitých součástek zapojení

Integrované obvody: ATMega644-20PU

ATMega162-16PU

LM317T

Displej MC1604CH-SYL

Rezistory: R1 – 240 Ω

R2, R3, R4, R5 - 470 Ω

Odporový trimr: P1 – 2,2 k Ω

Kondenzátory: C1, C2 - 100 μ F

C3, C4, C6, C7 - 22 pF

C5 - 100 nF

Cívka L1 - 10 μ H

Diody D1, D2, D3 - standardní diody s proudem do 20 mA

Krystaly XTAL1 - 20 MHz

XTAL2 - 16 MHz