

ZÁPADOČESKÁ UNIVERZITA V PLZNI

---

Fakulta elektrotechnická  
Katedra elektroniky a informačních technologií

# DIPLOMOVÁ PRÁCE

Elektronický wattmetr

Autor práce: **Bc. Jan Rada**  
Vedoucí práce: **Prof. Ing. Milan Štork, CSc.**

---

2023

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2022/2023

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Jan RADA**  
Osobní číslo: **E20N0053P**  
Studijní program: **N0714A060013 Elektronika a informační technologie**  
Specializace: **Elektronika**  
Téma práce: **Elektronický wattmetr**  
Zadávající katedra: **Katedra elektroniky a informačních technologií**

## Zásady pro vypracování

1. Návrh a realizace elektronického wattmetru s integrovaným obvodem pro měření výkonu s mikrokontrolérem pro řízení, zobrazení případně přenos naměřených dat.
2. Přehled integrovaných obvodů pro měření výkonu.
3. Návrh měřícího přístroje.
4. Realizace e testy.

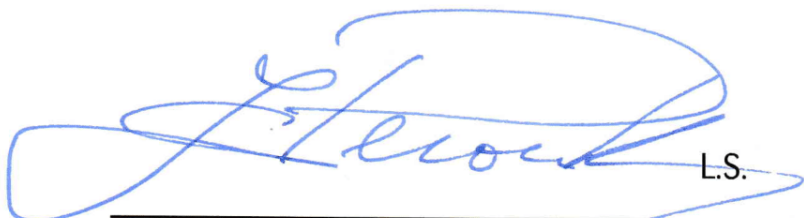
Rozsah diplomové práce: **40 – 60**  
Rozsah grafických prací: **dle doporučení vedoucího**  
Forma zpracování diplomové práce: **elektronická**

Seznam doporučené literatury:

1. <https://datasheets.maximintegrated.com/en/ds/78M6613.pdf>
2. <https://www.analog.com/media/en/technical-documentation/data-sheets/ADE7768.pdf>
3. <http://ww1.microchip.com/downloads/en/devicedoc/21948b.pdf>
4. [https://www.versasense.com/files/datasheets/CS5463\\_F3.pdf](https://www.versasense.com/files/datasheets/CS5463_F3.pdf)

Vedoucí diplomové práce: **Prof. Ing. Milan Štork, CSc.**  
Katedra elektroniky a informačních technologií

Datum zadání diplomové práce: **7. října 2022**  
Termín odevzdání diplomové práce: **26. května 2023**



L.S.

---

**Prof. Ing. Zdeněk Peroutka, Ph.D.**  
děkan



---

**Doc. Ing. Jiří Hammerbauer, Ph.D.**  
vedoucí katedry

V Plzni dne 7. října 2022

## **Abstrakt**

Práce se zabývá problematikou integrovaných obvodů pro měření výkonu, návrhem elektronického wattmetru, jeho realizací a na závěr je tento wattmetr ověřen různými měřeními. Jsou zde podrobně popsány různé integrované obvody pro měření výkonu od různých výrobců, které jsou aktuálně na trhu. Wattmetr byl sestaven z vývojové desky *Nucleo32-F303K8* s mikrokontrolerem, obvodu *MCP39F511A* pro měření výkonu a zobrazovače v podobě čtyřřádkového displeje *EA DOGS164x-A*.

## **Klíčová slova**

elektronický wattmetr, integrovaný obvod pro měření výkonu, návrh měřicího zařízení, mikrokontroler

## **Abstract**

The thesis deals with the issue of integrated circuits for power measurement, the design of an electronic wattmeter, its implementation, and at the end, this wattmeter is verified by various measurements. Various power measurement ICs from various manufacturers currently on the market are detailed here. The wattmeter was assembled from a *Nucleo32-F303K8* development board with a microcontroller, an *MCP39F511A* circuit for power measurements and an *EA DOGS164x-A* four-line display.

## **Key Words**

electronic wattmeter, integrated circuit for power measurements, design of measuring device, microcontroller

# Obsah

Úvod.....	- 1 -
1 Přehled integrovaných obvodů pro měření výkonu.....	- 2 -
1.1.1 Měřicí integrovaný obvod CS5463 .....	- 3 -
1.1.2 Měřicí integrovaný obvod 78M6613 .....	- 4 -
1.2 Měřicí integrované obvody firmy Analog Devices.....	- 6 -
1.2.1 Měřicí integrovaný obvod ADE9153A.....	- 6 -
1.2.2 Měřicí integrovaný obvod ADE7759 .....	- 8 -
1.2.3 Měřicí integrovaný obvod ADE7758 .....	- 9 -
1.3 Měřicí integrované obvody firmy Microchip.....	- 10 -
1.3.1 Měřicí integrovaný obvod MCP3905 .....	- 10 -
1.3.2 Měřicí integrovaný obvod MCP39F511A .....	- 12 -
2 Návrh elektronického wattmetru s využitím integrovaného obvodu pro měření výkonu .....	- 14 -
2.1 Cíle návrhu elektronického wattmetru .....	- 14 -
2.2 Komponenty použité při návrhu elektronického wattmetru.....	- 14 -
2.2.1 MCP39F511A .....	- 14 -
2.2.2 Vývojová deska PWR Meter click.....	- 14 -
2.2.3 Nucleo32-F303K8 .....	- 16 -
2.2.4 Proudová sonda s Hallovou sondou ACHS-7121 .....	- 17 -
2.2.5 LCD displej EADOGS164W-A.....	- 18 -
2.2.6 Optočlen HCPL-070L-000E .....	- 18 -
2.2.7 DC/DC měnič IES0103S05 .....	- 19 -
2.2.8 Nábojová pumpa ICL7660CPAZ .....	- 20 -
2.2.9 Operační zesilovač TC7650CPA .....	- 20 -
3 Realizace elektronického wattmetru s využitím integrovaného obvodu pro měření výkonu .....	- 22 -
3.1 Práce s integrovaným obvodem pro měření výkonu MCP39F511A (a s deskou PWR Meter click).....	- 22 -
3.1.1 Převodník USB to UART CH340.....	- 22 -

3.1.2	Zapojení desky PWR Meter click .....	- 23 -
3.1.3	Komunikace s deskou PWR Meter click .....	- 23 -
3.1.4	Příkazy pro měřicí obvod MCP39F511A .....	- 24 -
3.1.5	Single-Wire Transmission Mode .....	- 28 -
3.1.6	Kalibrace .....	- 29 -
3.2	Práce s deskou Nucleo32-F303K8 .....	- 31 -
3.3	Měření napětí a proudu .....	- 32 -
3.4	Galvanické oddělení a zdroje napájení.....	- 34 -
3.5	Práce s displejem.....	- 36 -
3.5.1	Zapojení displeje.....	- 36 -
3.5.2	Komunikace s displejem .....	- 37 -
3.5.3	Inicializace displeje.....	- 38 -
3.6	Návrh schématu a desky plošných spojů.....	- 39 -
3.7	Naprogramování zařízení .....	- 42 -
3.8	Oživení přístroje.....	- 42 -
3.9	Konstrukční krabice .....	- 43 -
3.10	Kompletace výrobku.....	- 44 -
4	Ověření funkcí navrženého a realizovaného elektronického wattmetru .....	- 46 -
4.1	Proudový senzor s Halloovou sondou.....	- 46 -
4.2	Schéma zapojení měření.....	- 47 -
4.3	Stejnoseměrné měření na vyrobeném přístroji.....	- 47 -
4.4	Stejnoseměrné měření na vyrobeném přístroji s externím napájením proudového senzoru -	50 -
4.5	Stejnoseměrné měření po kalibraci .....	- 52 -
4.6	Střídavé měření na vyrobeném přístroji .....	- 56 -
5	Závěr a diskuse výsledků.....	- 60 -
	Literatura.....	- 61 -
1	Příloha – doporučené zapojení obvodu MCP39F511A [22] .....	I
2	Příloha – schéma zapojení desky PWR Meter click [25] .....	II
3	Příloha – schéma zapojení navrženého měřidla .....	III
4	Příloha – návrh desky plošných spojů .....	V
5	Příloha – výkres navržené konstrukční krabice.....	VI

6	Příloha – potisk krabičky .....	VIII
7	Příloha – programový kód v jazyku C .....	IX



## Seznam symbolů a zkratek

Značka	Popisek	Jednotka
$P$	Činný výkon	W
$Q$	Jalový výkon	var
$S$	Zdánlivý výkon	VA
$\cos \varphi$	Účinnost	–
$f$	Frekvence	Hz
$U$	Napětí	V
$I$	Proud	A
$R$	Odpor	$\Omega$
SNR	Signal-to-noise ratio, tj. Poměr signálu k šumu	dB
IO	Integrovaný obvod	
OZ	Operační zesilovač	
AC	Alternating current, tj. střídavý proud	
DC	Direct current, tj. stejnosměrný proud	
U(S)ART	Universal (Synchronous) Asynchronous Receiver and Transmitter, tj. Univerzální (synchronní) asynchronní sériové rozhraní	
SPI	Serial Peripheral Interface, tj. sériové periferní rozhraní	
I <sup>2</sup> C	Inter-Integrated Circuit	
IoT	Internet of things, tj. internet věcí	
RAM	Random Access Memory	
EEPROM	Electrically Erasable Programmable Read-Only Memory, tj. Elektricky vymazatelná paměť pouze pro čtení)	
LDO	Low-dropout regulátor	
PC	Personal computer, tj. osobní počítač	
USB	Universal serial bus, tj. univerzální sériová sběrnice	
RMS	Efektivní hodnota	
ADC	Analogově digitální převodník	
TTL	Transistor-transistor-logic, tj. tranzistorově-tranzistorová logika	
LED	Light-Emitting Diode, tj. svítivá dioda	
LCD	Liquid crystal display, tj. displej z tekutých krystalů	
VPC	Virtual COM port	
EMI	Elektromagnetická interference	
PLA	Polylactic acid, tj. kyselina polylactická	
CAD	Computer Aided Design, tj. počítačem podporované navrhování	
LSB	Least Significant Bit, tj. nejméně významný bit	
MSB	Most Significant Bit, tj. nejvýznamnější bit	
SSI	Simple Sensor Interface	
PGA	Programmable Gain Amplifier, tj. zesilovač s programovatelným zesílením	

## Úvod

Vzhledem ke stále se zvyšujícím nárokům na snižování spotřeby elektrické energie a se vzniklou energetickou, krizí způsobenou ruskou agresí na Ukrajině, vznikl námět na tuto diplomovou práci, která se zabývá integrovanými obvody pro měření výkonu. Cílem je sestavit přehled těchto integrovaných obvodů. Informace byly čerpány zejména z katalogových listů jednotlivých integrovaných obvodů a z internetových stránek jednotlivých výrobců.

V praktické části se práce zabývá návrhem, realizací a testováním vlastního měřicího přístroje – elektronického wattmetru. Tento byl sestaven z integrovaného obvodu *MCP39F511A*, vývojové desky *Nucleo32-F303K8* s mikrokontrolerem *STM32F303K8T6* a čtyřřádkového displeje *EA DOGS164x-A*. Nejprve jsou popsány jednotlivé komponenty a popsány důvody, proč byly použity. Dále je v práci detailně popsána práce s obvodem *MCP39F511A*, zejména komunikace po rozhraní UART, nastavování registrů a vyčítání naměřených hodnot. Následuje popis realizace výrobku, konstrukce a testování. Část praktického úseku se také věnuje programování mikrokontroleru v jazyku C.

### 1 Přehled integrovaných obvodů pro měření výkonu

V této kapitole jsou popsány a blíže představeny některé vybrané integrované obvody pro měření výkonů. Integrované obvody pro měření výkonů jsou speciální obvody, které umožňují měřit různé parametry související s energetikou a kvalitou napájení.

Tyto obvody pracují na principu měření proudu a napětí, které prochází elektrickým obvodem. Napětí se je měřeno pomocí odporového děliče a diferenciálního zesilovače a proud je měřen pomocí Hallova senzoru nebo pomocí shunt rezistoru. Po získání napěťového a proudového signálu se signály digitalizují a pomocí matematických výpočtů se vypočítá spotřeba energie a výkon. Tyto výpočty zahrnují integraci proudu a násobení výsledné hodnoty napětí. [1, 2]

Měřicí obvody obvykle obsahují několik analogově-digitálních převodníků (ADC), které převádějí analogové signály napětí a proudu na digitální hodnoty. Tyto hodnoty jsou pak zpracovány pomocí digitálního signálového procesoru (DSP), který provádí např. výpočty energie, výkonu, frekvence, účinníku atd. Tyto výpočty jsou pak uloženy v registrech obvodu a jsou dostupné pro čtení nebo zápis pomocí komunikačního rozhraní jako je SPI, UART nebo I<sup>2</sup>C. Integrované obvody pro měření výkonů také podporují různé funkce kvality napájení jako je detekce průchodu nulou, poklesu a zvýšení napětí, špičkového proudu atd. [1, 2]

Integrované obvody pro měření výkonů jsou často nazývány jako „power meter ICs“ nebo „energy meter ICs“. Tyto obvody slouží k měření spotřeby energie v elektronických zařízeních, jako jsou například domácí spotřebiče, průmyslové stroje a elektromobily. Mohou také obsahovat různé funkce pro kalibraci a komunikaci s dalšími elektronickými zařízeními, jako jsou například mikrokontrolery. Integrované obvody pro měření výkonu se používají ve smart metrech, které umožňují monitorovat a řídit spotřebu energie domácností nebo průmyslových zařízení. Tyto smart metry také umožňují komunikaci s distribuční sítí a poskytují informace o stavu napájení. [1, 2]

Třídy měřičů energie jsou definovány jako třídy přesnosti měření spotřeby elektrické energie. Třída přesnosti měření spotřeby elektrické energie je definována jako maximální chyba měření v procentech. Třída 1 je nejpřesnější a má maximální chybu měření 1 %. Třída 2 má maximální chybu měření 2 %. Třída 3 má maximální chybu měření 5 %. Třída 4 má maximální chybu měření 10 %. [3]

### 1.1.1 Měřicí integrovaný obvod CS5463

Tento obvod byl vyvinut firmou *Cirrus Logic* a je navržen pro použití v aplikacích s vysokými nároky na přesnost měření, jako jsou například výroba a distribuce energie, solární systémy nebo průmyslové aplikace. Integrovaný obvod *CS5463* je vysoce výkonný měřicí integrovaný obvod, který se používá k měření činného, zdánlivého a jalového výkonu, napětí, proudu, účinníku ( $\cos \varphi$ ) a dalších parametrů elektrického signálu. Mezi klíčové vlastnosti *CS5463* patří vysoká přesnost měření, vysoká odolnost proti rušení, nízká spotřeba energie a široký teplotní rozsah provozu. Obvod je schopen měřit činný výkon s přesností až 0,1 % napětí a proud s přesností až 0,05 %. [4, 5]

Další výhodou *CS5463* je jeho odolnost proti rušení. Obvod je navržen tak, aby odolal vysokému šumu a interferenci, které se vyskytují v průmyslových aplikacích. Tato odolnost proti rušení zajišťuje spolehlivý a přesný výkon obvodu. *CS5463* také spotřebovává velmi málo energie, což je velmi užitečné v aplikacích s omezenou energetickou spotřebou. Obvod spotřebuje méně než 10 mW, což ho činí vhodným pro bateriové aplikace nebo aplikace s omezenou energetickou spotřebou. [4, 5]

Frekvenční rozsah měření obvodu *CS5463* je závislý na konkrétní aplikaci a nastavení obvodu. Nicméně, v dokumentaci obvodu je uváděno, že obvod může měřit v rozsahu od 20 Hz do 2 kHz. Použití tohoto obvodu je tedy omezeno pouze na střídavé průběhy. Tento frekvenční rozsah je vhodný pro měření v oblasti průmyslové automatizace, zpracování dat z obnovitelných zdrojů energie a v dalších aplikacích, kde jsou měřeny střídavé signály v uvedeném frekvenčním rozsahu. Pro měření vyšších frekvencí je vhodné použít jiný typ měřicího obvodu. [4, 5]

Vzhledem k vysokému výkonu a přesnosti je *CS5463* vhodný pro širokou škálu průmyslových aplikací, jako jsou například měření spotřeby elektrické energie, monitorování napájecích systémů, řízení motorů a další aplikace, kde je kladen důraz na vysokou přesnost měření a spolehlivost v různých prostředích. Tyto vlastnosti jsou však doprovázeny vyššími náklady a složitějším návrhem desky plošných spojů. [4, 5]

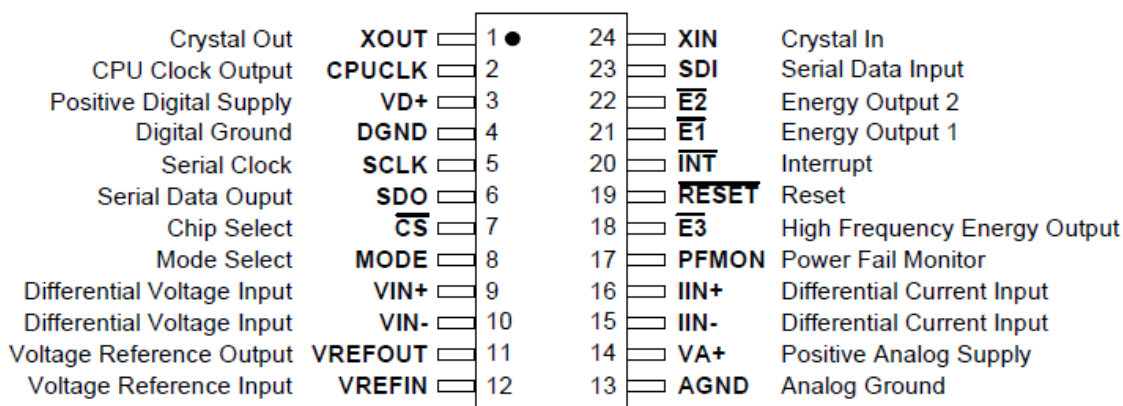
Na *Obr. 1* je vyfocený integrovaný obvod *CS5463* a na *Obr. 2* jsou popsány jednotlivé vývody.

## Shrnutí [5]:

- Frekvenční rozsah měření: 20 Hz až 2 kHz
- Rozlišení A/D převodníku: 24 bitů
- Přesnost měření: 0,1 %
- Rozhraní: I<sup>2</sup>C, SPI, pulzní výstup
- Napájení: 2,7 V až 5,5 V
- Doba odezvy: 20 ms
- Odolnost proti rušení a elektrostatickým výbojům
- Pouzdro: SSOP-24



Obr. 1: Fotografie integrovaného obvodu CS5463 [4]



Obr. 2: Pinout integrovaného obvodu CS5463 [5]

### 1.1.2 Měřicí integrovaný obvod 78M6613

Obvod *78M6613* je vysoce integrovaný IO pro zjednodušenou implementaci jednofázového měření AC do zdrojů napájení, chytrých spotřebičů a dalších aplikací s vestavěným monitorováním a řízením zátěže AC. Je dodáván v malém pouzdru.

5 mm x 5 mm a obsahuje 32bitový procesor *CEVA-DSP1*. Má také rozhraní UART, SPI a I<sup>2</sup>C pro komunikaci s hostitelským procesorem. [6–8]

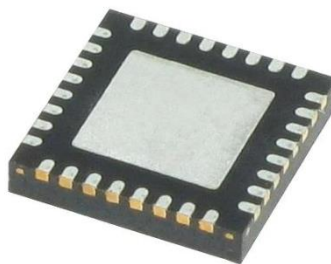
Tento obvod je vyvinut společností *Maxim Integrated* a je součástí jejich portfolia produktů pro správu energie. Hlavní výhodou *78M6613* je jeho jednoduchost a snadná integrace do existujících elektronických systémů. Obvod *78M6613* umožňuje měření proudu a napětí a vypočítává výkon pomocí interních výpočetních algoritmů. [7]

Obvod *78M6613* dokáže měřit napětí, proud, činný výkon, jalový výkon, zdánlivý výkon, účinník a harmonické složky signálu AC. Může také provádět kalibraci a kompenzaci teploty. Obvod *78M6613* je navržen tak, aby minimalizoval počet externích součástek a snížil náklady na systém. [6–8]

Obvod *78M6613* je vhodný pro různé typy aplikací, jako jsou inteligentní zásuvky, inteligentní osvětlení, inteligentní termostaty, inteligentní ventilátory a další spotřebiče s funkcemi monitorování a řízení spotřeby energie. Obvod *78M6613* umožňuje vytvářet energeticky efektivní a ekologické produkty pro domácí i průmyslové použití. Díky svému nízkému proudu spotřeby a vestavěnému teplotnímu senzoru je obvod vhodný pro použití v bateriových aplikacích, jako jsou například bezdrátové senzory a internet věcí. [6, 7]

### **Shrnutí [7]:**

- Frekvenční rozsah: 45 Hz až 65 Hz
- Rozlišení A/D převodníku: 22 bitů
- Přesnost měření: 0,5 %
- Vestavěný mikrokontroler s programovatelnou pamětí Flash a RAM
- Rozhraní UART, SPI, I<sup>2</sup>C
- Napájení 3 V až 3,6 V
- Pouzdro: QFN-32



Obr. 3: Fotografie integrovaného obvodu 78M6613 v pouzdře QFN-32 [8]

### 1.2 Měřicí integrované obvody firmy Analog Devices

Tato kapitola bude více zaměřena jen na jednoho výrobce integrovaných obvodů pro měření výkonů. *Analog Devices* nabízí širokou škálu obvodů pro měření výkonu a kvality napájení v různých aplikacích, jako jsou průmyslové, energetické, spotřebitelské nebo automobilové. Některé z klíčových vlastností a výhod těchto obvodů, jako jsou vysoká přesnost, nízká spotřeba energie, snadná integrace a flexibilita. *Analog Devices* nabízí jak izolované, tak neizolované měřicí obvody. [9, 10]

Izolované měřiče jsou takové, které oddělují měřený signál od zbytku obvodu pomocí nějaké formy izolační bariéry. Tím se zabrání nežádoucím rušivým proudům nebo napětím a zvyšuje se bezpečnost a spolehlivost měření. Neizolované měřiče jsou takové, které nemají izolační bariéru a jsou propojeny s měřeným signálem přímo nebo prostřednictvím nízkohmových rezistorů. [11, 12]

Příkladem izolovaného měřiče od *Analog Devices* je *ADE9153A*, který používá izolovaný sigma-delta modulátor pro přenos analogových signálů přes galvanickou izolaci. Tento obvod je vhodný pro aplikace jako inteligentní elektroměry nebo monitorování kvality napájení. [12]

Příkladem neizolovaného měřiče od *Analog Devices* je *ADE7758*, který používá shunt rezistory pro snímání proudu v třífázových systémech. Tento obvod je vhodný pro aplikace jako je průmyslová automatizace nebo energetické řídicí systémy. [12]

#### 1.2.1 Měřicí integrovaný obvod ADE9153A

Obvod *ADE9153A* je určen pro měření energie v jednofázových systémech. Může se použít například pro jednofázové měřiče energie, měření výkonu a energie, osvětlení ulic, inteligentní distribuci energie nebo monitorování stavu strojů. [13, 14]

Obvod *ADE9153A* měří několik parametrů souvisejících s energetikou a kvalitou napájení. Mezi ně patří napětí a proud, činná energie, jalová energie a zdánlivá energie, činný výkon, jalový výkon a zdánlivý výkon, frekvence sítě, doba periody a účinník. Také má detekci průchodu nulou, poklesu a zvýšení napětí, špičkového proudu a účinníku. Všechny tyto hodnoty jsou uloženy v registrech obvodu a jsou aktualizovány každou periodu fáze. [13, 14]

*ADE9153A* se hodí pro měření kvality napájení, jako jsou detekce nulového přechodu, výpočet délky linie, měření úhlu, detekce výpadků a přepětí, detekce vrcholů a přetížení a měření faktoru výkonu. Každý vstupní kanál podporuje nezávislé a flexibilní ziskové stupně. Obvod není primárně určen pro přesné měření výkonu a energie, ale spíše na kvalitu napájení. [13, 14]

Obvod *ADE9153A* může být napájen z měřené fáze nebo z vnějšího zdroje. Pokud je napájen z měřené fáze, používá se interní lineární regulátor (LDO), který snižuje vstupní napětí na 3,3 V. Pokud je napájen z vnějšího zdroje, musí být dodrženo rozsah napájecího napětí 2,97 V až 3,63 V. Obvod *ADE9153A* má nízkou spotřebu proudu: typicky 13 mA při plném provozu a 1  $\mu$ A při režimu spánku. [13, 14]

Obvod *ADE9153A* obsahuje funkci *mSure*, která umožňuje automatickou kalibraci a zajišťuje vysokou přesnost měření energie. *mSure* autokalibrace podporuje měřiče třídy 1 a třídy 2. Obvod *ADE9153A* obsahuje tři vysokovýkonné analogově-digitální převodníky (ADC), které poskytují poměr signálu k šumu (*SNR*) 88 dB [14, 15].

Obvod *ADE9153A* má několik rozhraní pro komunikaci s vnějšími zařízeními. Jedním z nich je sériové periferní rozhraní (SPI), které umožňuje čtení a zápis do registrů obvodu. Dalším je UART rozhraní, které umožňuje přenos dat mezi obvodem a hostitelským procesorem. Obvod *ADE9153A* také podporuje I<sup>2</sup>C rozhraní pro komunikaci s externími paměťovými zařízeními jako je EEPROM. Všechna tato rozhraní jsou konfigurovatelná pomocí registrů obvodu. [13]

Obvod *ADE9153A* je dostupný v pouzdru LFCSP (lead frame chip scale package) s 32 vývody. Rozměry pouzdra jsou 5 mm x 5 mm x 0,75 mm. Rozteč vývodů je 0,5 mm.

### **Shrnutí [13, 14]:**

- Frekvenční rozsah: 45 Hz až 65 Hz
- Rozlišení A/D převodníku: 24 bitů, 32 bitů pro akumulovanou energii
- Přesnost měření: 0,1 %



- Rozhraní UART, SPI, I<sup>2</sup>C
- Napájení 2,97 V až 3,63 V nebo z měřené fáze
- Spotřeba: 13 mA, 1  $\mu$ A při režimu spánku
- Pouzdro LFCSP32



Obr. 4: Integrovaný obvodu ADE9153A v pouzdře LFCSP32 [15]

### 1.2.2 Měřicí integrovaný obvod ADE7759

*ADE7759* je přesný integrovaný obvod pro měření činného výkonu a energie se sériovým rozhraním a pulzním výstupem. *ADE7759* obsahuje dva sigma-delta ADC druhého řádu, digitální integrátor (na kanálu 1), referenční obvody, teplotní senzor a veškeré zpracování signálu potřebné k provádění měření činného výkonu a energie. Digitální integrátor na čipu poskytuje přímé přístup k proudovým snímačům, jako je Rogowského cívka, které mají výstup  $di/dt$ . To vytváří i galvanické oddělení. Digitální integrátor eliminuje potřebu externího analogového integrátoru a toto řešení poskytuje vynikající dlouhodobou stabilitu a přesné fázové přizpůsobení mezi proudovými a napěťovými kanály. Integrátor lze zapínat a vypínat podle aktuálního zvoleného snímače. [16]

Poskytuje měření s přesností až 0,1 %, což umožňuje měřit energii a výkon s vysokou přesností. Tento obvod je schopen měřit napětí a proud v rozsahu až 5 kV a 5 A, což umožňuje měřit vysoké úrovně energie a výkonu. *ADE7759* obsahuje vestavěný 16bitový ADC pro měření napětí, proudu a fázového posunu s kmitočtem vzorkování až 8 kHz. Obvod je schopen detekovat zpětný tok energie, což umožňuje detekovat případné problémy v napájení nebo výkonových systémech. Obsahuje rozhraní pro komunikaci s mikrokontrolerem pomocí sériového rozhraní SPI. [16]

Podle datasheetu obvodu *ADE7759* je frekvenční rozsah měřených výkonů až do 2,5 kHz. Tento rozsah by měl být dostatečný pro většinu běžných aplikací, jako jsou průmyslové systémy řízení a měření energie, avšak při některých speciálních aplikacích, jako jsou například některé druhy motorů nebo vysokofrekvenční systémy, může být nutné použít jiný typ energetického měřiče s vyšším frekvenčním rozsahem. [16]

*ADE7759* je k dispozici ve 20vývodovém SSOP pouzdře a je pin-to-pin kompatibilní s *ADE7753* a *ADE7756*. [16]

Celkově lze říci, že obvod *ADE7759* poskytuje vysokou přesnost a stabilitu při měření energie a výkonu v širokém rozsahu a různých měřicích režimech. Je vhodný pro mnoho různých aplikací, jako jsou například průmyslová automatizace, energetické systémy a řízení spotřeby energie. [16]

### **Shrnutí [16]:**

- Frekvenční rozsah: 45 Hz až 65 Hz, v režimu kvality sítě až do 2,5 kHz
- Rozlišení A/D převodníku: 16 bitů
- Přesnost měření: 0,1 %
- Rozhraní: SPI
- Napájení: 5 V
- Spotřeba: 25 mW
- Pouzdro SSOP-20

### **1.2.3 Měřicí integrovaný obvod ADE7758**

*ADE7758* je integrovaný obvod pro měření aktivní, reaktivní a zdánlivé energie v různých třífázových konfiguracích, jako jsou hvězda nebo trojúhelník, se třemi nebo čtyřmi vodiči. *ADE7758* poskytuje kalibrační funkce systému pro každou fázi, tj. korekci offsetu RMS, kalibraci fáze a kalibraci výkonu. [17, 18]

Frekvenční rozsah měřeného signálu pomocí tohoto obvodu závisí na zvoleném režimu měření. V režimu měření spotřeby energie, obvod *ADE7758* měří střídavý signál v rozsahu 45 Hz až 65 Hz. To odpovídá standardní frekvenci elektrické sítě v mnoha zemích. Jak v těch, kde je používána frekvence 60 Hz (například v USA, Kanadě či Mexiku), tak v zemích, kde se používá frekvence 50 Hz (například v Evropě a Asii). Tento rozsah však může být rozsah frekvencí měřeného signálu obvodu *ADE7758* upraven. V režimu měření kvality energie, obvod *ADE7758* může měřit signály s frekvencí až do 2,5 kHz. Tento režim měření umožňuje detekovat různé typy poruch v napájecí síti, jako jsou například vysokofrekvenční rušení a špičky napětí. Je důležité poznamenat, že frekvenční rozsah měřeného signálu obvodem *ADE7758* může být ovlivněn externími komponentami a konkrétními nastaveními obvodu. [17]

*ADE7758* obsahuje registr vzorků z průběhu, který umožňuje přístup k výstupům ADC. Obvod také obsahuje detekční obvod pro krátkodobé změny poklesu napětí nebo přepětí. Prahové napěťové úrovně a doba trvání překročení (počet půlperiodových cyklů) jsou uživatelsky programovatelné. Detekce průchodu nulou je synchronizována s bodem průchodu nulou síťového napětí kterékoli ze tří fází. Tyto informace lze použít k měření periody kteréhokoli ze tří napěťových vstupů. Detekce průchodu nulou se používá uvnitř čipu pro režim akumulace energie fáze. Tento režim umožňuje rychlejší a přesnější kalibraci synchronizováním akumulace energie s celočíselným počtem period fáze.[17]

Data jsou čtena z *ADE7758* přes sériové rozhraní SPI. Interrupt request (IRQ) je typu open-drain a je aktivní v log. 0. IRQ výstup se aktivuje nízkou úrovní, dojde-li k jedné nebo více událostem přerušení v *ADE7758*. Stavový registr označuje povahu přerušení. *ADE7758* je k dispozici ve 24vývodovém pouzdře SOIC. [17]

### **Shrnutí [17]:**

- Frekvenční rozsah: 45 Hz až 65 Hz, v režimu kvality sítě až do 2,5 kHz
- Počet fází: 3
- Rozlišení A/D převodníku:
- Přesnost měření: 0,1 %
- Rozhraní: SPI, pulzní výstup
- Napájení: 5 V
- Spotřeba: 70 mW
- Pouzdro SOIC-24

### **1.3 Měřicí integrované obvody firmy Microchip**

#### **1.3.1 Měřicí integrovaný obvod MCP3905**

*MCP3905* je integrovaný obvod pro měření energie navržený tak, aby podporoval specifikaci mezinárodního standardu měření *IEC 62053*. Je to jednofázový měřicí obvod, který poskytuje jak frekvenční výstup úměrný průměrnému činnému výkonu, tak vyšší frekvenční výstup úměrný okamžitému výkonu pro kalibraci elektroměru. Obsahuje dva 16bitové, delta-sigma ADC pro široký rozsah  $I_B$  a  $I_{MAX}$  proudů a/nebo provedení s malým bočnickem (<200  $\mu\Omega$ ). Zahrnuje ultranízky drift napětí s <15 ppm/°C prostřednictvím

speciálně navržené teplotní křivky zakázaného pásu pro minimální gradient v průmyslovém teplotním rozsahu. Pro výpočet aktivního výkonu je na čipu blok DSP s pevnou funkcí. Tento přesný integrovaný obvod pro měření energie s vysokou spolehlivostí je k dispozici v průmyslovém standardním pinoutu. [19, 20]

*MCP3905* je integrovaný obvod pro měření energie, který má frekvenční výstup úměrný činnému výkonu a vyšší frekvenční výstup úměrný okamžitému výkonu pro kalibraci měřiče. Oba kanály používají 16bitové ADC druhého řádu delta-sigma, které převzorkují vstup na frekvenci rovné  $MCLK/4$  (tj.  $3,58 \text{ MHz}/4$ ), což umožňuje široký dynamický rozsah vstupních signálů. Programovatelný zesilovač zesílení (PGA) zvyšuje použitelný rozsah na aktuálním vstupním kanálu (kanál 0). Výpočet činného výkonu a filtrování spojené s tímto výpočtem se provádí v digitální formě, což zajišťuje lepší stabilitu driftu. Dvě digitální horní propusti odfiltrují systémový offset na obou kanálech, takže do výpočtu reálného výkonu se nezahrnuje žádný obvodový nebo systémový offset. Po filtraci horní propustí se napětové a proudové signály vynásobí, aby poskytly signál okamžitého výkonu. Tento signál neobsahuje složky stejnosměrného offsetu, takže může být efektivně použita technika průměrování k získání požadovaného výsledku činného výkonu. Signál okamžitého výkonu obsahuje informace o skutečném výkonu; je to stejnosměrná složka okamžitého výkonu. [19, 20]

Techniku průměrování lze použít jak pro sinusové, tak i nesinusové průběhy, stejně tak pro účinník. Okamžitý výkon je tedy filtrován dolní propustí, aby se vytvořil okamžitý signál skutečného výkonu. Digitální převodník na frekvenci shromažďuje informace o okamžitém činném výkonu a vytváří výstupní impulsy s frekvencí úměrnou průměrnému činnému výkonu. [19, 20]

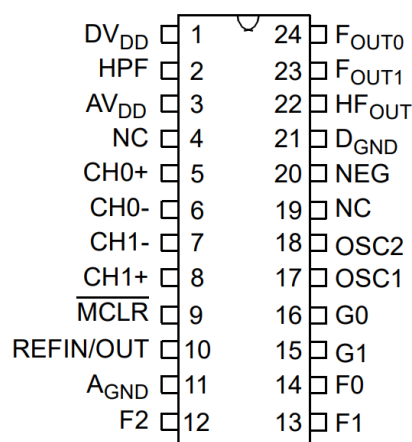
Nízkofrekvenční impulsy na výstupech  $F_{OUT0}$  a  $F_{OUT1}$  jsou navrženy tak, aby poháněly elektromechanické čítače a dvoufázové krokové motory zobrazující činnou spotřebu energie. Každý puls odpovídá pevnému množství činné energie, zvolenému nastavení logiky  $F_2$ ,  $F_1$  a  $F_0$ . Výstup  $HF_{OUT}$  má nastavenou vyšší frekvenci a kratší dobu integrace, takže může představovat okamžitý signál činného výkonu. Díky kratší době akumulace umožňuje uživateli přistoupit k rychlejší kalibraci za podmínek stálého zatížení. [19, 20]

## Shrnutí [20]:

- Frekvenční rozsah: 45 Hz až 65 Hz
- Rozlišení A/D převodníku: 16 bitů
- Přesnost měření: 0,1 %
- Rozhraní: frekvenční výstupy
- Napájení: 4,5 V až 5,5 V
- Spotřeba: 4 mA
- Pouzdro SSOP-24



Obr. 5: Fotografie integrovaného obvodu MCP3905 [20]



Obr. 6: Pinout integrovaného obvodu MCP3905 [20]

### 1.3.2 Měřicí integrovaný obvod MCP39F511A

Všechny výše popsané obvody byly určeny pouze pro střídavé průběhy. V systémech, které využívají střídavý i stejnosměrný proud je implementace monitorování napájení ve dvou režimech (tedy AC i DC) obvykle vyžaduje více integrovaných obvodů, aby byl zajištěn vynikající výkon a přesnost. Rostoucí aplikace, jako jsou solární invertory, inteligentní osvětlení a cloudové servery, často využívají oba režimy k udržení bezpečného provozu, přičemž střídavý proud je hlavním zdrojem energie a stejnosměrný proud jako záloha nebo naopak. [21]

Pro optimalizaci výkonu a usnadnění vývoje těchto systémů vyvinula společnost *Microchip Technology Inc.* flexibilní dvourežimový integrovaný obvod pro monitorování napájení, který měří střídavý i stejnosměrný režim se špičkovou přesností 0,1 % v širokém dynamickém rozsahu 4000:1. Výpočty napájení a monitorování událostí jsou součástí jediného integrovaného obvodu, což snižuje náklady na kusovníky a čas na vývoj firmwaru. [21, 22]

*MCP39F511A* je vysoce integrované zařízení, které řeší rostoucí potřebu přesnějších měření výkonu ve vysoce výkonných konstrukcích. Pro zjednodušení kalibračních procedur a uspokojení většiny požadavků na přesnost jsou k dispozici dva 24bitové delta-sigma analogově-digitální převodníky (ADC) s poměrem signálu k šumu 94,5 dB plus zkreslení (SINAD) a 16bitový výpočetní engine. zahrnuta. [21, 22]

Mezi další výhody používání *MCP39F511A* patří jeho flexibilita a snadná implementace. Zařízení poskytuje standardní výpočty výkonu, jako je činný, jalový a zdánlivý výkon, činná a jalová energie, efektivní hodnota proudu a napětí (RMS), frekvence fáze a účinník, což umožňuje návrhářům snadno přidávat vysoce přesné funkce monitorování výkonu. koncové aplikace s minimálním vývojem firmwaru. [21]

Obvod je vhodný pro řadu spotřebitelských aplikací, internet věcí (IoT) i průmyslové aplikace, automaticky snímá typy napájecích zdrojů a přepíná mezi střídavým a stejnosměrným režimem, čímž optimalizuje výsledky měření. Zařízení také pomáhá vývojářům odstraňovat problémy díky EEPROM na čipu, do které se zaznamenávají kritické události. Integrovaná referenční hodnota napětí s nízkým driftem a interním oscilátor snižují náklady na implementaci. [21]

Typické zapojení je součástí *1. Příloha – doporučené zapojení obvodu MCP39F511A*.

### **Shrnutí [22]:**

- Frekvenční rozsah: 0 až 4 kHz
- Rozlišení A/D převodníku: 24 bitů
- Přesnost měření: 0,1 %
- Rozhraní: UART
- Napájení: 3,3 V
- Spotřeba: 75 mA
- Pouzdro QFN-28



Obr. 7: Fotografie obvodu MCP39F511A [21]

## 2 Návrh elektronického wattmetru s využitím integrovaného obvodu pro měření výkonu

V této kapitole je popsán návrh elektronického wattmetru s využitím integrovaného obvodu pro měření výkonu. Je zde uvedena motivace a cíle návrhu zařízení, a také jsou zde popsány jednotlivé komponenty navrhovaného zařízení. Konstrukce, schémata zapojení, návrh desky plošných spojů, popis funkce zařízení a popis funkce kódu jsou popsány v následující kapitole č. 3: *Realizace elektronického wattmetru s využitím integrovaného obvodu pro měření výkonu*.

### 2.1 Cíle návrhu elektronického wattmetru

Za cíle návrhu byly stanoveny následující vlastnosti: kompaktní měřicí přístroj s displejem, který bude zobrazovat v reálném čase měřené výkony (činný, jalový, zdánlivý), napětí a proud, případně frekvenci. Přístroj by měl měřit jak stejnosměrný, tak střídavý výkon. Případné vylepšení by mohlo zahrnovat propojení s PC buď pomocí USB kabelu nebo pomocí bezdrátové technologie, např. *Bluetooth*.

### 2.2 Komponenty použité při návrhu elektronického wattmetru

#### 2.2.1 MCP39F511A

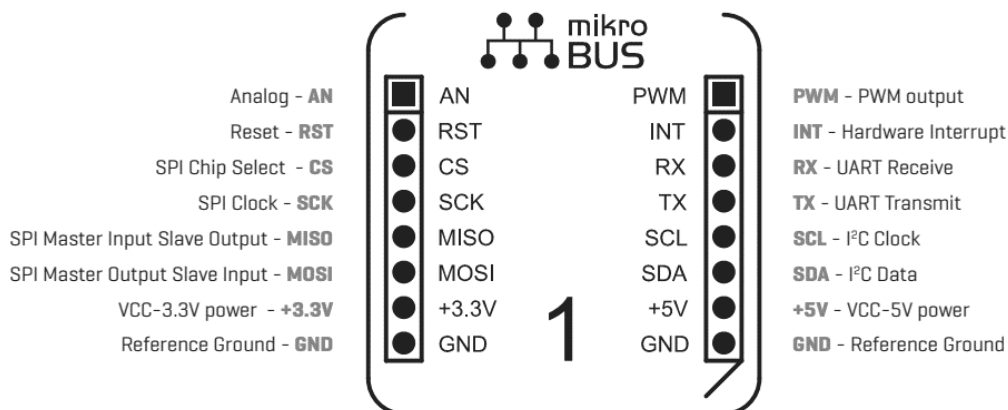
Přístroj by měl měřit jak stejnosměrný, tak střídavý výkon. Tomuto požadavku vyhovoval jediný ze zkoumaných integrovaných obvodů pro měření výkonu, a to *MCP39F511A*. Hlavní vlastnosti obvodu byly popsány v kapitole č. 1.3.2 *Měřicí integrovaný obvod MCP39F511A*. Komunikace s obvodem a další podrobnosti jsou popsány dále v kapitole č. 3 *Realizace elektronického wattmetru s využitím integrovaného obvodu pro měření výkonu*.

#### 2.2.2 Vývojová deska PWR Meter click

Firma *MikroElektronika* vyrábí vývojovou desku s integrovaným obvodem *MCP39F511A* pro měření výkonu. Deska s názvem *PWR Meter click* (viz *Obr. 9*) je součástí konceptu *Click Boards* od tohoto výrobce. Koncept je určený pro rychlý vývoj embedded systémů, kdy se *Click Boards* destička zasune do jednotné patice a modul je tak ihned připravený k použití bez jakéhokoliv pájení, navrhování a výroby pomocných obvodů. Ukázka je na *Obr. 10*. Všechny desky v této rodině mají konektor pro sběrnici *mikroBUS™*.

## 2. NÁVRH ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Sběrnice je takovým „kontejnerem“, který obsahuje sběrnice USART, SPI, I<sup>2</sup>C, dále PWM, napájení 5 V a 3,3 V, analogový kanál, přerušení a reset. Nákres konektoru je na *Obr. 8*. [23]

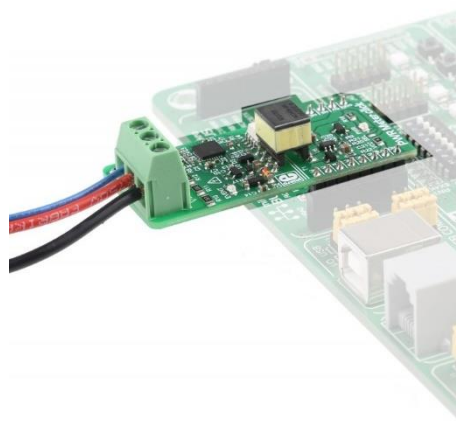


Obr. 8: Nákres konektoru sběrnice mikroBUS™ [23]

V našem případě je z tohoto konektoru a z této sběrnice využito jen velmi málo. V levé části byly využity jen napájecí piny  $+3,3V$  a  $GND$ , v pravé části piny  $RX$  a  $TX$  pro komunikaci a napájecí piny  $+5V$  a  $GND$ . Ostatní piny zůstávají nepřipojeny.



Obr. 9: PWR Meter click [24]



Obr. 10: PWR Meter click připojený do základové desky [24]

Měření proudu na této desce je řešeno pomocí shunt odporu  $200\text{ m}\Omega$ . Vzhledem k rozsahům ADC na vstupu  $MCP39F511A \pm 600\text{ mV}$  je možné měřit proud  $\pm 3\text{ A}$ . Napětí je měřeno přes napěťový dělič  $100:1$ , tzn. maximální možné měřené napětí je  $60\text{ V}$ . Celé



## 2. NÁVRH ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

---

schéma desky *PWR Meter click* je součástí 2. Příloha – schéma zapojení desky *PWR Meter click*. [24, 25]

Deska byla na začátku návrhu použita pro zkoušku komunikace s obvodem a pro obecné vyzkoušení si práce s obvodem. Práce s deskou je dále popsána v kapitole č. 3 *Realizace elektronického wattmetru s využitím integrovaného obvodu pro měření výkonu*.

### 2.2.3 Nucleo32-F303K8

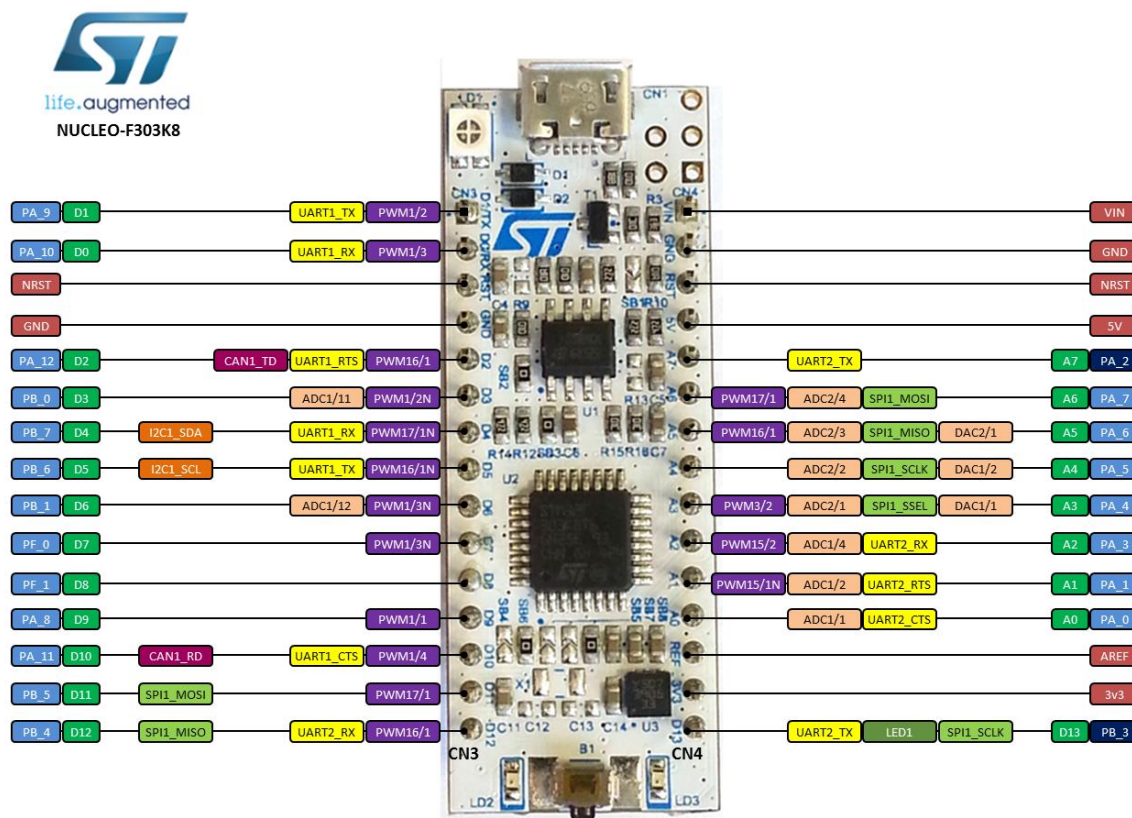
Dále je zapotřebí mikrokontroler, který bude celý měřicí přístroj řídit. K tomu byla zvolena deska *Nucleo32-F303K8*. Tato vývojová deska od společnosti *STMicroelectronics* byla vybrána pro svoji jednoduchost ovládání, komunikace a připojení, svoje malé rozměry, postačující výkon a relativně nízkou cenu, především ale pro moji předchozí zkušenost s deskou při tvorbě bakalářské práce. Srdcem desky je mikrokontroler *STM32F303K8T6* v pouzdře *LQFP32*. Fotografie desky a její pinout je na *Obr. 11*. [26]

**Hlavními parametry tohoto jednočipového počítače [26]:**

- ARM®32-bit Cortex®-M4 CPU s FPU
- maximální taktovací frekvence 72 MHz
- 64 kB Flash paměti
- 12 kB SRAM paměti
- 12bitový A/D převodník s 9 kanály
- 12bitový D/A převodník se 3 kanály
- SPI a I<sup>2</sup>S rozhraní
- USART rozhraní
- CAN rozhraní
- 25 GPIO portů s možností externího přerušení
- RTC krystal
- Časovače

Desku je možné připojit k PC pomocí USB-mikro B konektoru. Zařízení se pak pro počítač tváří jako 3 zařízení: VCP, velkokapacitní uložení a debug port. Součástí desky je i druhý mikrokontroler, který řídí komunikaci po USB, debug, nahrávání programu apod. [26]

## 2. NÁVRH ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU



Obr. 11: Deska Nucleo32-F303K8 a její pinout [26]

### 2.2.4 Proudová sonda s Hallovou sondou ACHS-7121

Pro měření proudu bylo využito integrovaného obvodu *ACHS-7121*. Tento obvod využívá Hallovu sondu pro snímání proudu, která nabízí proudovou cestu s nízkým odporem (přibližně  $0,7 \text{ m}\Omega$ ) a elektrickou izolaci až do  $3 \text{ kV RMS}$ . Největšími výhodami jsou možnost snímání jak stejnosměrného, tak střídavého proudu, a zároveň galvanické oddělení od měřeného obvodu. Tato verze měří proud o velikosti od  $-10 \text{ A}$  do  $+10 \text{ A}$  a na výstupu je proporcionální analogové napětí s citlivostí  $185 \text{ mV/A}$  se středem v  $2,5 \text{ V}$  s typickou chybou  $\pm 1,5 \%$ . Obvod pracuje s napájecím napětím od  $4,5 \text{ V}$  do  $5,5 \text{ V}$  a je určen pro použití v  $5\text{V}$  systémech. [27]

Pro jednodušší práci s tímto SMD integrovaným obvodem byla použita deska *POLOLU-4030*, na které je tento obvod už zapájen a k němu i dva kondenzátory – filtrační a blokovací. Další výhodou je, že deska má připravené pokovené díry, do kterých lze přišroubovat přívodní vodiče proudu. [27]

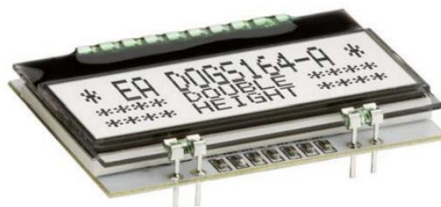
### 2.2.5 LCD displej EADOGS164W-A

Pro zobrazování měřených hodnot je samozřejmě zapotřebí nějaký zobrazovač. Pro tento návrh zařízení byl zvolen LCD displej *EA DOGS164W-A*.

Je to vysokokontrastní LCD displej s volitelnými jednotkami LED podsvícení v různých barvách. Displej je v konfiguraci 4x16 znaků s výškou znaku 3,2 mm, což by mělo být pro účely měřicího přístroje dostačující, nabízí však i nastavitelnou variantu 2x16 znaků s výškou znaku 6 mm, případně třířádkové kombinace větší a menší výšky písma. V ROM paměti jsou k dispozici 3 různé znakové sady – cyrilice, anglicko-japonská a evropská. Je zde také nastavitelný směr pohledu – shora nebo zdola. [28]

Tento displej je FSTN pozitivní transflektivní, tzn. že může být provozován jak s podsvícením, tak bez něj. Zde byl použit přídatný modul s třibarevným podsvícením *EA LED40x33-ERW*. Displej se jednoduše zapájí do DPS modulu podsvícení. Barvu podsvícení lze složit z barev: bílá, zelená a červená. [28]

LCD je řízen ovladačem *SSD1803A* s rozhraními SPI (tři nebo čtyřvodičová varianta) a I<sup>2</sup>C (dvouvodičový). Displej je napájen 3,3 V s typickou spotřebou 440  $\mu$ A, LED podsvícení se spotřebou až 45 mA (tj. samozřejmě nastavitelné omezujícím odporem). [28]



Obr. 12: Fotografie displeje EA DOGS164W-A s LED podsvícením EA LED40x33-ERW [28]

### 2.2.6 Optočlen HCPL-070L-000E

V případě konstrukce měřicího zařízení takového charakteru, jako je tento elektronický wattmetr, je velmi vhodné měřicí obvod galvanicky oddělit. V tomto návrhu je třeba oddělit napájení a komunikaci mikrokontroleru s měřicím obvodem po UART. Právě signály *Rx* a *Tx* byly galvanicky odděleny pomocí optočlenu *HCPL-070L-000E*.

Tento vazební člen s vysokým ziskem využívá svítivou diodu (LED) a integrovaný fotodetektor s vysokým ziskem pro zajištění extrémně vysokého poměru přenosu proudu mezi vstupem a výstupem. Samostatné piny pro fotodiodu a koncový stupeň zajišťují vysokorychlostní provoz a saturační napětí kompatibilní s hladinami LVTTTL/LVCMOS.

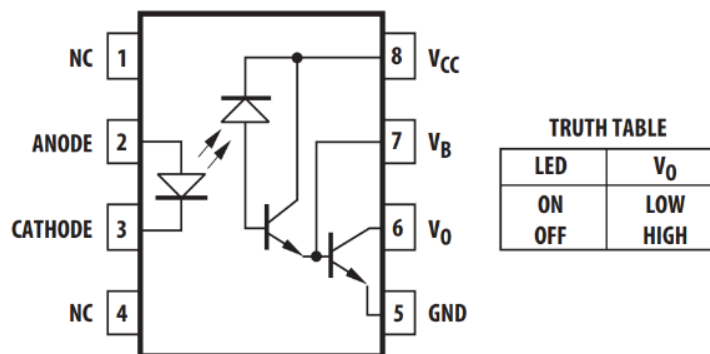
## 2. NÁVRH ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Tam, kde je to žádoucí, mohou být piny  $V_{CC}$  a  $V_O$  svázaný dohromady, aby se dosáhlo konvenčního fotodarlingtonova provozu. [29]

Tyto optočleny jsou určeny pro použití v LVTTL/LVCMOS technologiích nebo jiných aplikacích s nízkou spotřebou. Minimální proudový přenosový poměr 400 % je zaručen v provozním rozsahu 0 až  $+70\text{ }^{\circ}\text{C}$  při proudu LED pouze 0,5 mA. HCPL-070L je IO pro povrchovou montáž zapouzdřená v průmyslovém standardu SOIC-8. Funkční zapojení optočlenu je znázorněno na *Obr. 13*. [29]

### Vlastnosti optočlenu [29]:

- 3,3 V / 5 V duální napájecí napětí
- malá spotřeba energie
- vysoký přenosový poměr proudu
- nízké požadavky na vstupní proud: 0,5 mA
- LVTTL/LVCMOS kompatibilní výstup
- vlastnosti zaručeny při teplotách  $0\text{ }^{\circ}\text{C}$  až  $+70\text{ }^{\circ}\text{C}$
- vysoký výstupní proud až 60 mA



Obr. 13: Funkční zapojení optočlenu HCPL-070L-000E [29]

### 2.2.7 DC/DC měnič IES0103S05

Další komponenta, která zajišťuje galvanické oddělení měřicího obvodu je izolovaný DC/DC měnič *IES0103S05*. Měnič byl použit pro oddělení stejnosměrného napájení.

Jednovýstupová řada měničů *IES01* od výrobce *XP Power* je ideálním řešením pro izolaci napěťových rozvodů napájení, jako jsou analogové, digitální, datové a reléové obvody. Tato řada produktů nabízí kompaktní design s vysokou účinností, izolací 1,5 kV, volitelně 3,0 kV,

## 2. NÁVRH ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

---

ochranou proti zkratu a vysokou provozní teplotou. Největší výhodou je beze sporu jednočipové řešení celého DC/DC měniče. Pouzdro obvodu má pouze vstupní a výstupní napěťové piny, ke kterým jsou připojeny pouze EMI filtry a celé zapojení je funkční. [30]

Z této řady byl do návrhu vybrán měnič se vstupním napětím 3,3 V a výstupním 5 V s maximálním výstupním proudem 200 mA. Výstupní napětí 5 V bylo zvoleno proto, aby bylo možné napájet měřicí senzor *POLOLU-4030* a operační zesilovač *TC7650CPA*, které potřebují právě 5V napájení. Napájecí napětí pro měřicí obvod *MCP39F511A* bylo opět sníženo na 3,3 V pomocí LDO regulátoru *LE33CZ-AP* s maximálním výstupem 100 mA. [30, 31]



Obr. 14: Fotografie DC/DC měniče řady IES01 [30]

### 2.2.8 Nábojová pumpa ICL7660CPAZ

Pro potřebu vytvoření záporného napětí pro vytvoření symetrického napájení pro operační zesilovač *TC7650CPA* byla použita nábojová pumpa *ICL7660CPAZ*. Nábojová pumpa byla zapojena tak, aby z napájecího napětí +5 V vytvořila výstupní napětí -5 V. Měnič je schopen dodávat proud až 20 mA. Vzhledem k předpokladu, že operační zesilovač nebude mít téměř žádný odběr, protože bude pouze v režimu napěťového výstupu bez zatížení, je tento proud dostačující. [32]

### 2.2.9 Operační zesilovač TC7650CPA

*TC7650CPA* je univerzální operační zesilovač s nízkým šumem, což znamená, že se hodí pro použití v široké škále aplikací, včetně zesilovačů signálu, komparátorů, oscilátorů a dalších obvodů. Tento operační zesilovač má velmi nízký vstupní proud, což umožňuje snížit šum a zkreslení signálu. Jedna z hlavních předností *TC7650CPA* je jeho schopnost pracovat s vysokým zesílením. Díky tomu je tento zesilovač vhodný pro aplikace, které vyžadují vysokou přesnost a stabilitu signálu. Vysoké zesílení zajišťuje vysoký výstupní signál při velmi nízké vstupní úrovni signálu. Operační zesilovač *TC7650CPA* nabízí také

## 2. NÁVRH ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

---

vysokou rychlost odezvy, což je důležité pro aplikace, které vyžadují rychlé odezvy na změny signálu. Tento zesilovač se hodí pro použití v aplikacích, jako jsou různé typy měření, řízení motorů a další. [33]

### **Hlavní vlastnosti [33]:**

- Nízké vstupní offsetové napětí:  $0,7 \mu\text{V}$
- Nízký drift vstupního offsetového napětí: max.  $0,05 \mu\text{V}/^\circ\text{C}$
- Nízký vstupní klidový proud: max.  $10 \text{ pA}$
- Vysokoimpedanční diferenciální vstupy CMOS:  $10^{12} \Omega$
- Vysoký zisk v otevřené smyčce: min.  $120 \text{ dB}$
- Nízké vstupní šumové napětí:  $2,0 \mu\text{V}_{\text{P-P}}$
- Vysoká rychlost přeběhu:  $2,5 \text{ V}/\mu\text{s}$
- Nízký příkon:  $20 \text{ mW}$
- Pouzdro: DIP-8 a DIP-14

### 3 Realizace elektronického wattmetru s využitím integrovaného obvodu pro měření výkonu

V této kapitole je podrobně popsána konstrukce, schémata zapojení, návrh desky plošných spojů, popis funkce zařízení a popis funkce programového kódu pro navržené měřicí zařízení. Jednotlivé komponenty byly popsány v předchozí kapitole č. 2. *Návrh elektronického wattmetru s využitím integrovaného obvodu pro měření výkonu.*

#### 3.1 Práce s integrovaným obvodem pro měření výkonu MCP39F511A (a s deskou PWR Meter click)

Stavba zařízení začala nejprve vyzkoušením již hotového řešení v podobě desky *PWR Meter click* od společnosti *MikroElektronika*.

##### 3.1.1 Převodník USB to UART CH340

Díky tomu, že obvod *MCP39F511A* komunikuje po rozhraní UART, je možné s ním komunikovat pomocí terminálu na PC. K tomu byl použit převodník USB to UART s označením *CH340*, viz *Obr. 15*. Napěťové úrovně komunikace lze jednoduše zvolit pomocí jumperu na 3,3 V nebo na 5 V. Pro účely komunikace s obvodem *MCP39F511A* musí být nastaveno 3,3 V. [22, 24, 34]

Bylo by teoreticky možné komunikovat přímo bez převodníku přes sériové rozhraní např. RS-232 jen s úpravou napěťových hladin na 0 a 3,3 V, ale takto to nebylo vyzkoušeno.



Obr. 15: Převodník USB to UART CH340 [34]

CH340	PWR Meter click
5V	5V
3V3	3V3
TXD	RX
RXD	TX
GND	GND

Tab. 1: Propojení CH340 a PWR Meter click

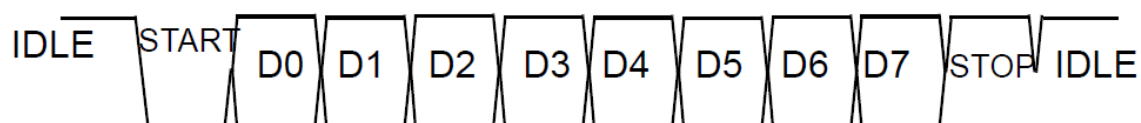
#### 3.1.2 Zapojení desky PWR Meter click

Díky tomu, že má převodník i napěťové piny 3,3 V a 5 V, bylo možné je využít jako napájení pro desku *PWR Meter click* a nic dalšího již nebylo potřeba připojovat. Propojení pinů převodníku s piny *PWR Meter click* je v tabulce *Tab. 1*. **Pozor, opravdu je nutné připojit 5V i 3,3V napájení.** Na desce *PWR Meter click* je sice možnost výběru napájení 5 V nebo 3,3 V pomocí solderbridge, který je defaultně nastavený na 3,3 V, ale při bližším průzkumu schématu zapojení desky je možné se přesvědčit, že flyback transformátor je přímo připojen na pin 5V bez ohledu na přepínač. Není to na první pohled zřejmé a výrobce na to v dokumentaci neupozorňuje, což je poměrně nevhodné. Bez připojení 5V napájení by nebyl obvod *MCP39F511A* napájen. [24, 25]

Dále je potřeba desku *PWR Meter click* zapojit do měřeného obvodu. K tomu je zde svorkovnice, do které se zapojí zdroj a zátěž. Zdroj se připojí mezi svorky V+ a V-, zátěž mezi svorky L a V-. V našem případě byl připojen laboratorní stejnosměrný zdroj 0 až 30 V a 0 až 3 A známý pod označením *G400*, který byl vyroben v rámci výuky v předmětu *Napájecí zdroje a systémy (KEI/NZAS)*. Jako zátěž byl připojený výkonový neproměnný rezistor 10  $\Omega$  / 20 W. [24]

#### 3.1.3 Komunikace s deskou PWR Meter click

Hardware je tímto připraven. Nyní je potřeba na PC otevřít terminál (např. *PuTTY* nebo *Hercules*) a otevřít VCP na který je převodník *CH340* připojen. Komunikační rychlost je potřeba nastavit na 9600 Bd, což je výchozí nastavení obvodu *MCP39F511A*. Měřicí obvod umožňuje nastavit komunikační rychlost na různé hodnoty, maximálně však 115200 Bd. Komunikační protokol pro zařízení *MCP39F511A* je založen na protokolu *Simple Sensor Interface* (SSI). Tento protokol se používá pro komunikaci point-to-point z mikrokontroléru do zařízení typu single-slave *MCP39F511A*. Zpráva musí být ve formátu viz *Obr. 16*. Začíná Start bitem, následují 8bitová data (začíná se LSB) a nakonec jeden Stop bit. Takto je potřeba terminál v PC nakonfigurovat. [22, 24]



Obr. 16: Formát zprávy pro komunikaci po UART s obvodem *MCP39F511A* [22]

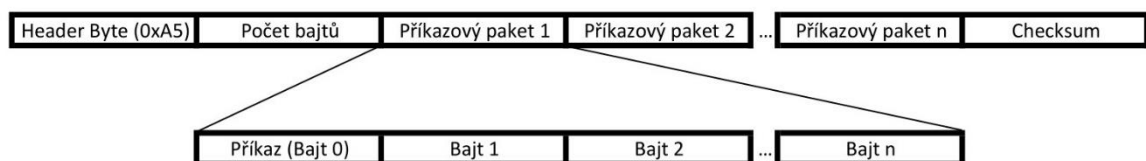


### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Veškerá komunikace se zařízením probíhá v rámcích. Každý rámeček se skládá z hlavičkového bajtu (header byte), počtu bajtů v rámci, příkazového paketu (nebo paketů) a kontrolního součtu (checksum). Náčrt rámečku je na *Obr. 17*. Maximální počet bajtů v přijímacím nebo vysílacím rámci je 35. Po přijetí komunikačního rámečku odpoví zařízení *MCP39F511A* třemi možnými odpověďmi, které budou vyslány nazpět společně s daty (nebo bez nich, záleží na charakteru přijatého rámečku). Tyto odpovědi jsou: [22]

- *Acknowledge* (ACK, 0x06): Potvrzení – rámeček přijatý úspěšně, příkazy pochopeny a příkazy provedeny úspěšně.
- *Negativní potvrzení* (NAK, 0x15): Rámeček přijatý úspěšně, ale příkazy nebyly úspěšně provedeny, příkazy nebyly pochopeny nebo se vyskytla nějaká jiná chyba v příkazových bajtech.
- *Selhání kontrolního součtu* (CSFAIL, 0x51): Rámeček přijat úspěšně, avšak kontrolní součet rámečku neodpovídal bytům v rámci.

Kontrolní součet (checksum) je generován pomocí jednoduchého sčítání bajtů. Výsledný checksum je zbytek po dělení součtu číslem 256. Tato operace se provádí za účelem získání 8bitového kontrolního součtu. V kontrolním součtu jsou zahrnuty všechny bajty rámečku, včetně header bajtu i počtu bajtů. Pokud rámeček obsahuje více rámečků příkazů, nebude v případě selhání kontrolního součtu rámečku proveden žádný z příkazů. V tomto případě zařízení *MCP39F511A* odpoví odpovědí CSFAIL (0x51). U příkazů, které požadují vyslání dat nazpět ze zařízení *MCP39F511A*, se rámeček a kontrolní součet vytvoří stejným způsobem, přičemž bajt záhlaví se stane potvrzovacím (0x06). [22]



Obr. 17: Vizualizace komunikačního rámečku vysílaného do MCP39F511A [22]

#### 3.1.4 Příkazy pro měřicí obvod MCP39F511A

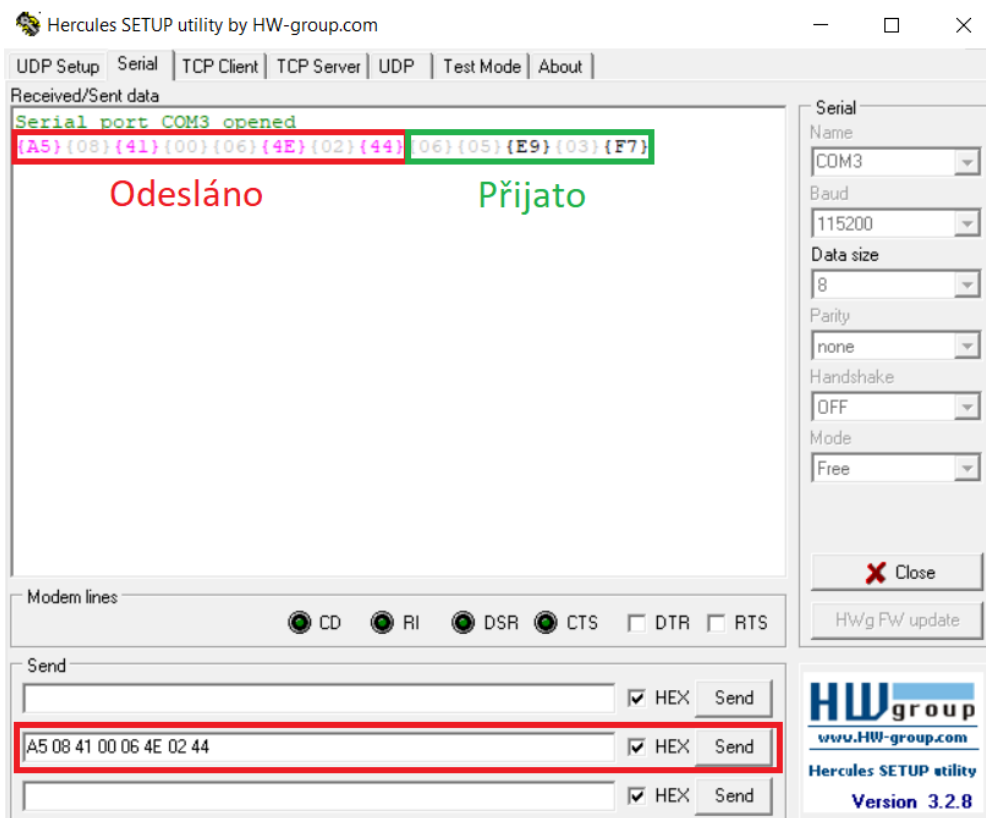
Následující tabulka *Tab. 2* je seznam všech akceptovaných příkazových bajtů pro zařízení *MCP39F511A*. Existuje 11 možných akceptovaných příkazů. Ve sloupci „Parametr“ je uvedeno, co musí být obsahem následujícího bajtu (bajtů).

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Příkaz	Číslo příkazu	Parametr	Počet bajtů	Odpověď
Přečti z registru $N$ bajtů	0x4E	Počet bajtů	2	ACK, počet bajtů, data, checksum
Zapiš do registru $N$ bajtů	0x4D	Počet bajtů	1+N	ACK
Nastav adresový ukazatel	0x41	Adresa	3	ACK
Ulož registry do Flash paměti	0x53	Žádný	1	ACK
Přečti stránku EEPROM	0x42	Stránka	2	ACK, počet bajtů, data, checksum
Zapiš stránku EEPROM	0x50	Stránka	18	ACK
Celkové vymazání EEPROM	0x4F	Žádný	1	ACK
Autokalibrace zisku	0x5A	Žádný	1	ACK
Autokalibrace reaktivního zisku	0x7A	Žádný	1	ACK
Autokalibrace frekvence	0x76	Žádný	1	ACK
Ulož počítadlo energie do EEPROM	0x45	Žádný	1	ACK

Tab. 2: Instrukční sada obvodu MCP39F511A [22]

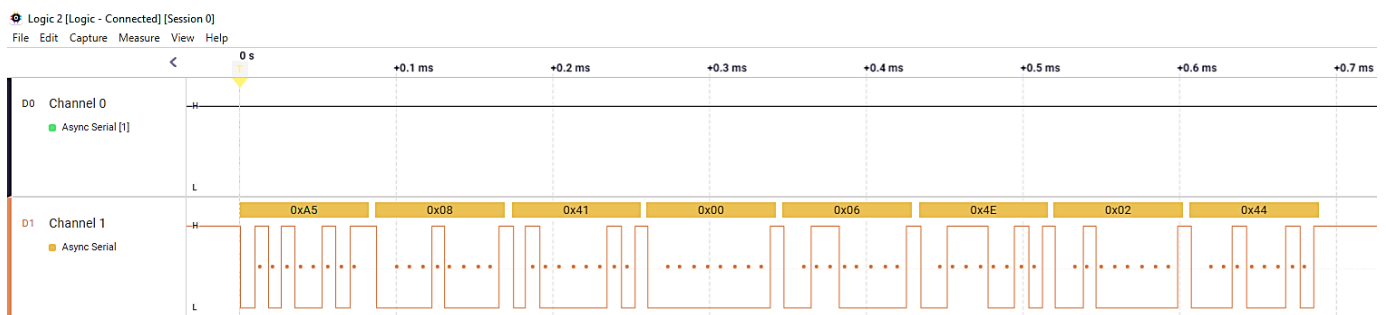
Následuje ukázka komunikace s deskou *PWR Meter click*, jak přečíst z registrů naměřené napětí. K ukázce byl použit terminál *Hercules*, který byl nastaven na zobrazování a vysílání znaků v hexadecimální podobě. Komunikační rychlost s měřicí deskou byla přenastavena na 115200 Bd. Snímek terminálu je na *Obr. 18*. Zdroj byl nastaven na hodnotu napětí 10 V, podle Ohmova zákona potoče při zátěži 10  $\Omega$  proud 1 A.



Obr. 18: Snímek terminálu Hercules při komunikaci s deskou PWR Meter click

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Ze snímku terminálu na *Obr. 18* lze vidět jak odeslaná data, tak přijatá data jako odpověď. Současně byl k signálům *TX* a *RX* připojen logický analyzátor *SL-USB8CH* pro vizualizaci fyzické vrstvy v logických úrovních na vodičích. Na kanál *CH0* byl připojen signál *TX* a na kanál *CH1* signál *RX* (z pohledu PC). Snímky se zachycením komunikace logickým analyzátořem jsou na obrázcích ***Chyba! Nenalezen zdroj odkazů.*** a *Obr. 20*. Analýza byla provedena programem *Logic 2*. Logické úrovně byly během analýzy dekodovány a jednotlivé bajty komunikace zobrazeny v hexadecimálním tvaru.



Obr. 19: Zachycení vysílací sekvence logickým analyzátořem



Obr. 20: Zachycení přijímací sekvence logickým analyzátořem

V tabulce *Tab. 3* a *Tab. 4* jsou rozepsány a vysvětleny jednotlivé bajty, které byly vyslány do desky *PWR Meter click* a bajty, které byly následně přijaté jako odpověď.

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Bajt	Význam bajtu
A5	Header bajt (hlavička.)
08	Počet bajtů ve zprávě = 8
41	Příkaz „Nastav adresový ukazatel“
00	Vyšší bajt adresy registru naměřeného napětí (celá adresa je 0x0006)
06	Nižší bajt adresy registru naměřeného napětí
4E	Příkaz „Přečti z registru $N$ bajtů“
02	$N = 2$ (přečti 2 bajty)
44	Kontrolní součet (checksum)

Tab. 3: Sekvence vyslaných bajtů do desky PWR Meter click pro vyčtení hodnoty naměřeného napětí

Bajt	Význam bajtu
A5	Header bajt (hlavička.)
05	Počet bajtů ve zprávě = 5
E9	Nižší bajt registru Voltage RMS
03	Vyšší bajt registru Voltage RMS
F7	Kontrolní součet (checksum)

Tab. 4: Sekvence přijatých bajtů z desky PWR Meter click pro vyčtení hodnoty naměřeného napětí

Z měřicího obvodu *MCP39F511A* osazeného na desce *PWR Meter click* jsme vyčetli hodnotu napěťového registru 0x03E9. Po převedení čísla do desítkové soustavy dostaneme hodnotu 1001, kterou je nutné vydělit konstantou 100, abychom dostali hodnotu napětí ve voltech. Naměřené napětí tedy odpovídá hodnotě 10,01 V. Konstanta reprezentuje nastavení PGA zesilovače a nastavení Range registru. Hodnota konstanty 100 je výchozí, nebyla nijak upravována. [22, 24]

Podobným způsobem lze vyčíst všechny další měřené hodnoty, jen se musí zadat adresa příslušného registru a počet požadovaných bajtů – záleží na velikosti konkrétního registru. V *Tab. 5* jsou popsány registry s měřenými hodnotami a v *Tab. 6* jsou popsány některé důležité konfigurační a kalibrační registry, kde  $u$  = neznaménková hodnota,  $s$  = znaménková hodnota. Znaménka hodnot (např. napětí a proudu v případě DC módu, nebo znaménka výkonů) jsou uloženy v *System Status* registru.

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Adresa registru	Velikost registru [bit]	Popis registru
0x0002	u16	Systém Status register
0x0006	u16	Napětí RMS
0x0008	u16	Frekvence
0x000E	u32	Proud RMS
0x0012	u32	Činný výkon $P$
0x0016	u32	Jalový výkon $Q$
0x001A	u32	Zdánlivý výkon $S$
0x000C	s16	Účinník $\cos \varphi$
0x001E	u64	Akumulátor aktivní energie, import
0x0026	u64	Akumulátor aktivní energie, export
0x002E	u64	Akumulátor reaktivní energie, import
0x0036	u64	Akumulátor reaktivní energie, export

Tab. 5: Tabulka registrů s naměřenými hodnotami [22]

Adresa registru	Velikost registru [bit]	Popis registru
0x0094	u16	Hlavní konfigurační registr – zde je nastavená rychlost komunikace po UART, zesílení PGA, Single Wire Mode apod.
0x0050	u16	Kalibrační faktor pro proud RMS (AC režim)
0x0052	u16	Kalibrační faktor pro napětí RMS (AC režim)
0x0054	u16	Kalibrační faktor pro činný výkon (AC režim)
0x0056	u16	Kalibrační faktor pro jalový výkon (AC režim)
0x006C	u16	Kalibrační faktor pro proud (DC režim)
0x006E	u16	Kalibrační faktor pro napětí (DC režim)
0x0070	u16	Kalibrační faktor pro činný výkon (DC režim)
0x0072	s16	Kalibrační faktor offsetu pro proud (DC režim)
0x0074	s16	Kalibrační faktor offsetu pro činný výkon (DC režim)

Tab. 6: Tabulka konfiguračních a kalibračních registrů [22]

#### 3.1.5 Single-Wire Transmission Mode

Obvod *MCP39F511A* umožňuje tzv. *Single-Wire Transmission Mode*, což jednosměrný přenos, kdy nemusím každý registr naměřených hodnot vyčítat zvlášť, ale kdy po nastavení příslušného bitu v konfiguračním registru bude zařízení automaticky kontinuálně na konci každého výpočetního cyklu vysílat rámec naměřených dat. Takový rámec dat v *Single-Wire* režimu se skládá z 20 bajtů. Tři bajty záhlaví (Header Byte), 16 bajtů naměřených dat (efektivní hodnota napětí a proudu, činný výkon, jalový výkon a frekvence) a kontrolního

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

součtu (checksum). Formát rámce je znázorněn v tabulce *Tab. 7*. Tento režim může být výhodné použít právě při použití obvodu *MCP39F511A* v měřicím přístroji, který kontinuálně zobrazuje naměřené hodnoty na zobrazovači. [22]

Pořadí bajtu	Bajt
1	Header Byte (0xAB)
2	Header Byte (0xCD)
3	Header Byte (0xEF)
4	Proud (ef. hodnota) – bajt 0
5	Proud (ef. hodnota) – bajt 1
6	Proud (ef. hodnota) – bajt 2
7	Proud (ef. hodnota) – bajt 3
8	Napětí (ef. hodnota) – bajt 0
9	Napětí (ef. hodnota) – bajt 1
10	Činný výkon – bajt 0
11	Činný výkon – bajt 1
12	Činný výkon – bajt 2
13	Činný výkon – bajt 3
14	Jalový výkon – bajt 0
15	Jalový výkon – bajt 1
16	Jalový výkon – bajt 2
17	Jalový výkon – bajt 3
18	Frekvence – bajt 0
19	Frekvence – bajt 1
20	Kontrolní součet (Checksum)

Tab. 7: Rámec dat v režimu Single-Wire [22]

#### 3.1.6 Kalibrace

Jestliže IO neměří s přesností, která je garantovaná výrobcem, je nutné provést kalibraci. Kalibrace je také nutná při nastavování jiných než výchozích rozsahů nebo při jiné hodnotě bočníku nebo jiném než výchozím poměru napěťového děliče. Kalibrace má za úkol kompenzovat chybu ADC, tolerance součástek a celkově šum v systému. Obvod na čipu obsahuje kalibrační algoritmus, který umožňuje rychlé provedení jednoduché kalibrace systému. [22]

Vynikající analogové vlastnosti ADC umožňují jednobodovou kalibraci a jediný kalibrační příkaz pro dosažení přesných měření v režimu AC. V režimu DC je obvykle ještě vyžadována kalibrace offsetu. Kalibraci lze provést buď pomocí předdefinovaných příkazů autokalibrace, nebo přímo zápisem do kalibračních registrů. [22]

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Autokalibrace zisku funguje tak, že do příslušných registrů jsou zapsány očekávané hodnoty proudu, napětí a činného výkonu. Na měřicí svorky přivedeme právě takové přesné napětí, a právě takový přesný proud, které byly zapsány do registrů, a příkazem autokalibrace se kalibrační registry automaticky nastaví do požadovaných hodnot. Autokalibrace reaktivního zisku a autokalibrace frekvence se provádějí obdobně. [22]

V případě ruční kalibrace je nutno vypočítat nový kalibrační zisk. Ten se vypočítá podle vztahu:

$$GAIN_{NEW} = GAIN_{OLD} \cdot \frac{Expected}{Measured}, \quad (1)$$

kde  $GAIN_{NEW}$  bude nová hodnota kalibračního registru zisku,  $GAIN_{OLD}$  je současná hodnota zapsaná v kalibračním registru zisku,  $Expected$  je očekávaná hodnota po změření,  $Measured$  je skutečná změřená hodnota. Je tedy nutné vyčíst starou kalibrační hodnotu z registru. Dále je nutné na proudové vstupy (tedy v případě, že bude kalibrován proud) přivést co možná nejpřesnější známou hodnotu proudu a zadat příkaz pro změření (např. toho proudu). Tím získáme všechny potřebné proměnné do vztahu (1). [22]

Následuje příklad pro kalibraci DC proudu:

- 1) Vysláním rámce {0xA5, 0x08, 0x41, 0x00, 0x6C, 0x4E, 0x02, 0xAA} vyčteme hodnotu registru *DC Gain Current RMS* na adrese 0x006C.
- 2) Byla vyčtena hodnota 0x8131 – toto je  $GAIN_{OLD}$ .
- 3) Na proudové vstupy přivedeme co možná nejpřesnější hodnotu proudu 1 A. Očekáváme tedy, že obvodem *MCP39F511A* změříme dekadickou hodnotu 100 000 000, tj. 0x05F5E100. (Číslo 100 000 000 bylo zvoleno pro jednoduchý přepočítání na 1 A, a zároveň bude efektivně využit rozsah registru při 10A rozsahu měření) – toto je hodnota *Expected*.
- 4) Vysláním rámce {0xA5, 0x08, 0x41, 0x00, 0x0E, 0x4E, 0x04, 0x4E} přečteme změřenou hodnotu proudu.
- 5) Byla vyčtena hodnota 0x05c9ca44 – toto je hodnota *Measured*.
- 6) Dosadíme do vztahu (1)

$$GAIN_{NEW} = 0x8131 \cdot \frac{0x05F5E100}{0x05c9ca44} \quad (2)$$

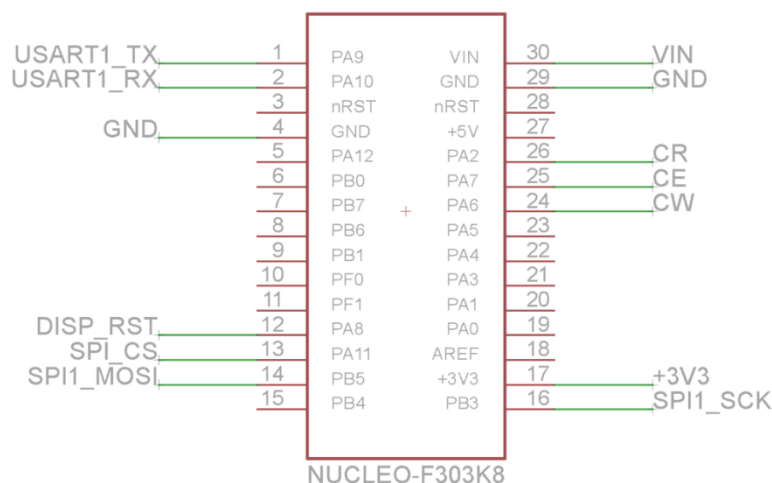
### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

$$GAIN_{NEW} = 0x8509 \quad (3)$$

- 7) Vysláním rámce {0xa5, 0x0a, 0x41, 0x00, 0x6c, 0x4d, 0x02, 0x09, 0x85, 0x39} zapíšeme do registru *DC Gain Current RMS* hodnotu  $GAIN_{NEW}$ , tj. 0x8509.
- 8) Tímto způsobem můžeme provést kalibraci ostatních parametrů (např. *DC Gain Voltage RMS*)
- 9) Příkazem *Save To Flash* - vysláním rámce {0xA5, 0x04, 0x53, 0xFC} uložíme všechny konfigurační a kalibrační registry do paměti Flash. Díky tomu zůstanou hodnoty konfiguračních a kalibračních registrů po odpojení napájení zachovány. Po opětovném naběhnutí napájení se hodnoty z Flash paměti automaticky přepíší do konfiguračních a kalibračních registrů, není třeba žádných příkazů.

#### 3.2 Práce s deskou Nucleo32-F303K8

Práce s vývojovou deskou *Nucleo32-F303K8* je velmi jednoduchá. Díky integrovanému debuggeru a programátoru se deska jednoduše připojí k PC pomocí USB kabelu. Samotný program pro mikrokontroler byl psán ve vývojovém prostředí *STM32CubeIDE*. Více o programovém kódu viz 3.7 *Naprogramování zařízení*. Zapojení pinů vývojové desky je na *Obr. 21.*, popis jednotlivých pinů je vypsán v *Tab. 8.*



Obr. 21: Zapojení pinů desky Nucleo32-F303K8



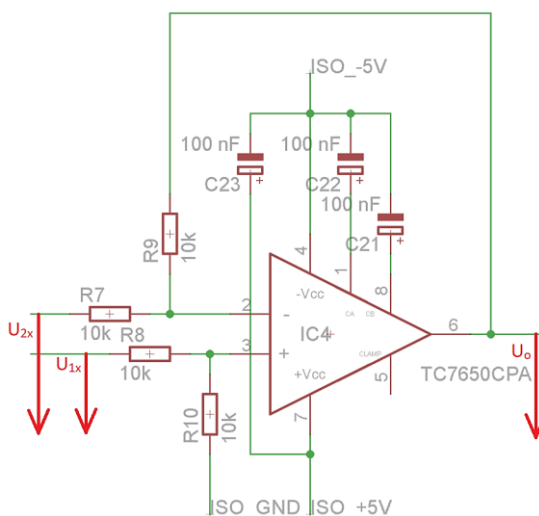
### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Číslo pinu	Label	Vstup / výstup	Popis
1	USART1_TX	OUT	Komunikace s měřicím obvodem po UART – vysílací pin
2	USART1_RX	IN	Komunikace s měřicím obvodem po UART – přijímací pin
4	GND	IN/OUT	Zemnicí pin
12	DISP_RST	OUT	Reset displeje (aktivní v log. 0)
13	SPI_CS	OUT	Chip select pro komunikaci s displejem (aktivní v log. 0)
14	SPI1_MOSI	OUT	Vysílací pin pro komunikaci s displejem
16	SPI1_SCK	OUT	Hodinový pin pro komunikaci s displejem
17	+3V3	OUT	Výstupní stabilizované napětí 3,3 V, které je přivedeno na DC/DC měnič
24	CW	OUT	Ovládací pin bílého podsvícení displeje (aktivní v log. 0)
25	CE	OUT	Ovládací pin zeleného podsvícení displeje (aktivní v log. 0)
26	CR	OUT	Ovládací pin červeného podsvícení displeje (aktivní v log. 0)
29	GND	IN/OUT	Zemnicí pin
30	VIN	IN	Vstupní napětí z baterie přivedeno na tento pin

Tab. 8: Popis pinů desky Nucleo32-F303K8

#### 3.3 Měření napětí a proudu

Jak už bylo zmíněno výše, pro měření proudu byl zvolen senzor s Halloovou sondou *ACHS-7121*. Měřicí obvod *MCP39F511A* ovšem očekává shunt rezistor, proto je třeba výstup ze senzoru náležitě upravit. Bylo již řečeno, že senzor má proudový rozsah od -10 V do +10 V. Výstupní napětí ze senzoru se pohybuje od 0 V do +5 V, kdy 0 A odpovídá 2,5 V. K tomuto účelu byl použit operační zesilovač *TC7650CPA* v zapojení, které je na *Obr. 22*.



Obr. 22: Zapojení operačního zesilovače

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Vzhledem k tomu, že platí podmínka

$$\frac{R_9}{R_7} = \frac{R_{10}}{R_8}, \quad (4)$$

bude se zapojení chovat jako rozdílový zesilovač. Výstupní napětí se určí podle vztahu

$$U_o = \frac{R_9}{R_7} \cdot (U_{2x} - U_{1x}). \quad (5)$$

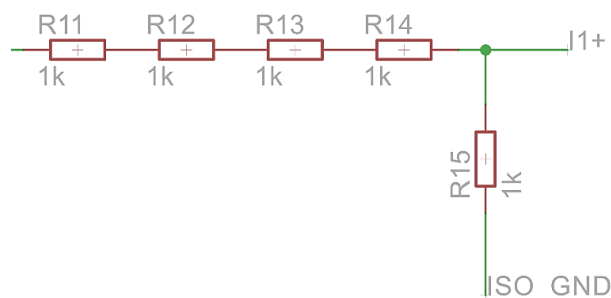
Protože ještě platí podmínka

$$R_9 = R_7, \quad (6)$$

bude výsledné výstupní napětí

$$U_o = U_{2x} - U_{1x}. \quad (7)$$

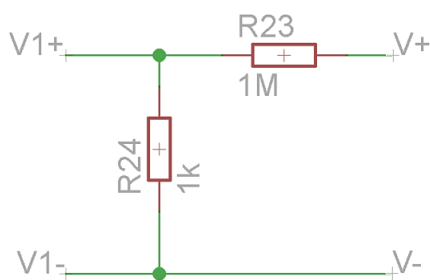
Napětí  $U_{2x}$  je připojeno na výstupní napětí z proudové sondy a napětí  $U_{1x}$  je připojeno na referenční napětí 2,5 V. Tím se rozsah napětí změní na -2,5 V až +2,5 V. Na vstupy měřicího obvodu *MCP39F511A* může být připojeno napětí v rozmezí od -600 mV do +600 mV, proto je nutné napětí ještě dále upravit. To bylo vyřešeno odporovým děličem 5:1 viz *Obr. 23*. Tímto se rozsah napětí změnil na -500 mV až +500 mV, kdy napětí +500 mV odpovídá proudu 10 A. Celé zapojení se tak pro měřicí obvod tváří jako shunt rezistor o hodnotě 50 mΩ.



Obr. 23: Odporový dělič pro přizpůsobení napětí proudového měřicího kanálu

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Měření napětí je provedeno stejně, jako v doporučeném zapojení výrobce obvodu *MCP39F511A*. Zde je zapojen odporový dělič v poměru 1000:1 viz *Obr. 24*. Pro tento poměr by byla potřeba odpor 999 k $\Omega$ , takže se tím zde vytváří chyba 0,1 %, což je ale zanedbatelné vzhledem k toleranci součástek 1 %.

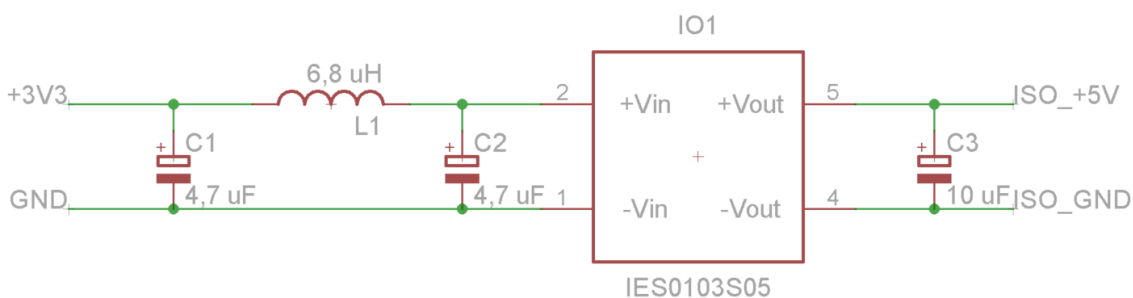


Obr. 24: Odporový dělič pro napěťový měřicí kanál

#### 3.4 Galvanické oddělení a zdroje napájení

Pro galvanické oddělení řídicí části a měřicí části slouží izolační DC/DC měnič a dvojice optočlenů.

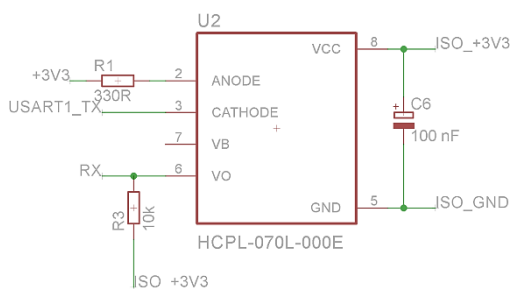
DC/DC měnič *IES0103S05* je zapojen podle *Obr. 25*. Schéma zapojení je shodné s doporučením výrobce, žádné úpravy nebyly třeba. Schéma zapojení obsahuje jen vstupní EMI filtr ve formě LC  $\Pi$ -článku a výstupní kondenzátor. Měnič zajišťuje izolační napětí 1,5 kV. [30]



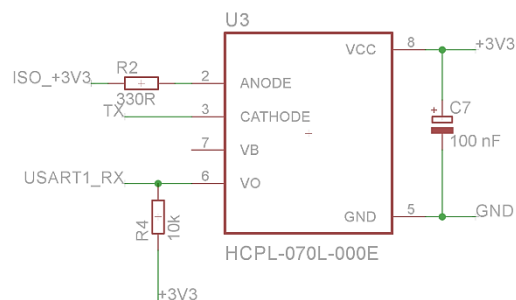
Obr. 25: Zapojení DC/DC měniče

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Zapojení optočlenů *HCPL-070L-000E* je na *Obr. 26* a *Obr. 27*. Optočlen zajišťuje izolační napětí 3,75 kV. Odpor R1, resp. R2 je omezujícím odporem pro LED ve vysílací části optočlenu. Odporů R3 a R4 vytváří pull-up. [29]

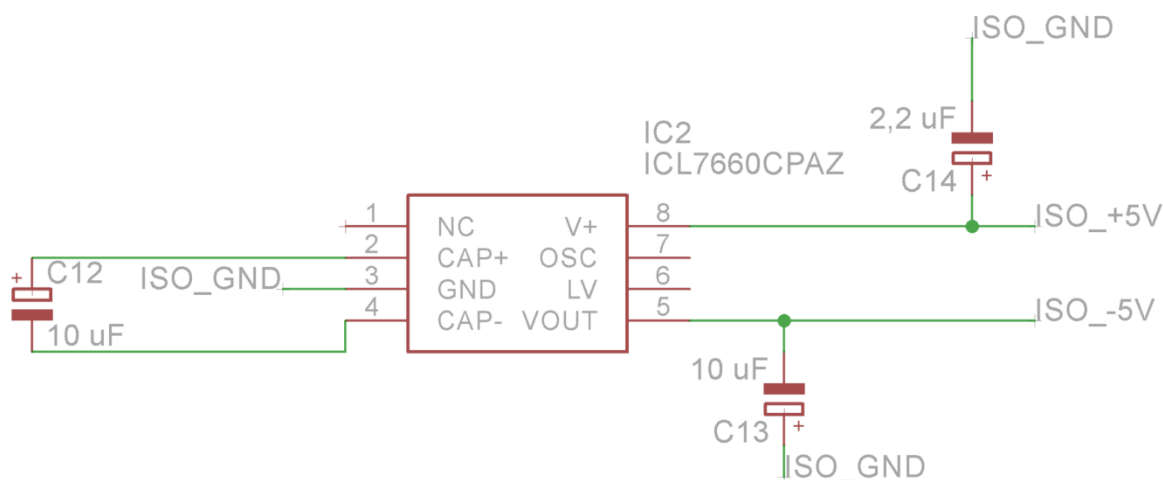


Obr. 26: Zapojení optočlenu pro komunikaci směrem z mikrokontroleru do měřicího obvodu



Obr. 27: Zapojení optočlenu pro komunikaci směrem z měřicího obvodu do mikrokontroleru

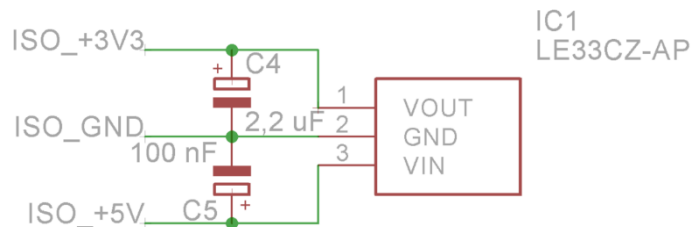
Pro nutnost symetrického napájení operačního zesilovače bylo potřebné vytvořit další napájecí napětí o hodnotě -5 V. K tomu byla použita nábojová pumpa *ICL7660CPAZ*. Obvod je zapojen podle *Obr. 28* v zapojení, kdy je vstupní napětí překlopeno na opačnou polaritu. Zapojení je shodné s doporučením výrobce. Výstupní proud může být až 20 mA. Vzhledem k předpokladu, že operační zesilovač nebude mít téměř žádný odběr, protože bude pouze v režimu napěťového výstupu bez zatížení (naprázdno), je tento proud dostačující. [32]



Obr. 28: Zapojení nábojové pumpy

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Pro vytvoření napájecího napětí 3,3 V pro měřicí obvod *MCP39F511A* byl použit jednoduchý třibodový LDO stabilizátor typu. Zapojení na *Obr. 29*:

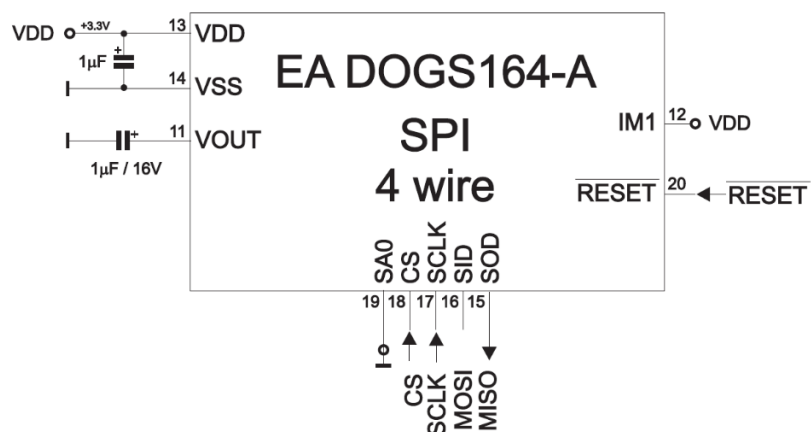


Obr. 29: Zapojení LDO stabilizátoru na napětí 3,3 V

## 3.5 Práce s displejem

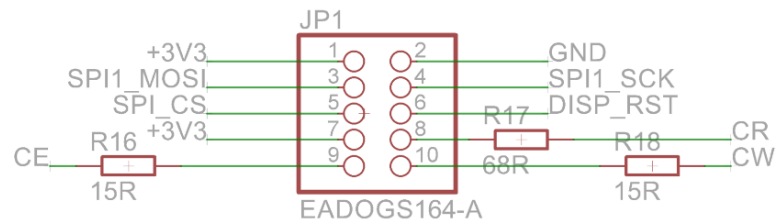
### 3.5.1 Zapojení displeje

Displej byl zapojen podle doporučení výrobce viz *Obr. 30*. Signál *SOD/MISO* nebyl zapojen. Dva kondenzátory  $1\mu\text{F}$ , které byly potřeba připojit podle schématu byly připájeny přímo na vývody displeje. Umístění displeje je v panelu krabičky, nikoliv na desce plošného spoje, proto byl k propojení s deskou plošných spojů použit plochý 10žilový kabel s prořezávaným konektorem. Zapojení konektoru je na *Obr. 31*. [28]

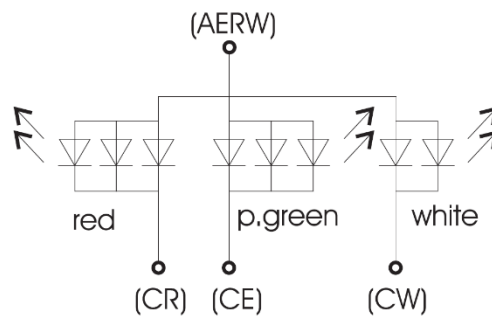


Obr. 30: Zapojení displeje EA DOGS164W-A pro komunikaci po SPI [28]

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU



Obr. 31: Zapojení konektoru pro připojení displeje EA DOGS164W-A



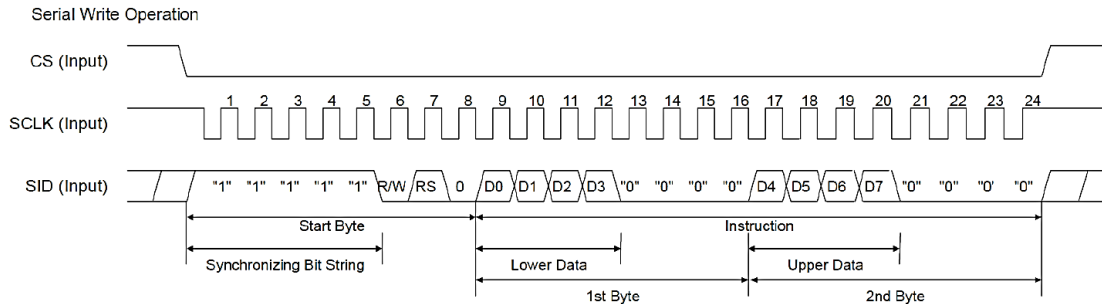
Obr. 32: Vnitřní zapojení modulu podsvícení EA LED40x33-ERW [28]

Signály *CR*, *CW* a *CE* se ovládá podsvícení displeje. Signál *AERW* je připojen přímo na napájení +3,3 V. Odporů *R16*, *R17* a *R18* jsou odpory omezující proud na 20 mA. Signály jsou připojeny na GPIO piny mikrokontroleru a je jimi podsvícení i napájeno, proto proud nesmí přesáhnou 25 mA, i když maximální proud osvětlení může být až 30 mA u bílého, resp. 40 mA u červeného a zeleného podsvícení. Osvětlení je aktivní v log. 0. Na *Obr. 32* je zobrazeno, jak je modul osvětlení *EA LED40x33-ERW* zapojen. [26, 28]

#### 3.5.2 Komunikace s displejem

Pro komunikaci s displejem byla použita sběrnice SPI. Časový průběh vysílací sekvence pro zápis dat do displeje je zobrazen na *Obr. 33*. Rámec se skládá vždy ze tří bajtů. Start bajt začíná pěti log. 1, následující bit R/W značí zápis nebo čtení, bit RS značí, jestli jsou vysílána data nebo příkaz a bajt je ukončen log. 0. V dalším bajtu už je vysílána první polovina samotných dat. Začíná se nižší polovinou bajtu od LSB, vyšlou se celkem čtyři bity dat, které jsou doplněny čtyřmi log. 0. Třetí vyslaný bajt je vyšší polovina datového bajtu a opět doplněná čtyřmi log. 0. Zpětná komunikace z displeje do mikrokontroleru nebyla implementována. [28]

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU



Obr. 33: Časový průběh vysílací sekvence pro zápis dat do displeje EA DOGS164W-A [28]

V mikrokontroleru zajišťuje vysílací algoritmus funkce `LCD_send()`:

```
void LCD_send(uint8_t val, bool rs)
{
    uint8_t startB = 0x1f, lowerB = 0, highB = 0;
    if(rs)
        startB |= 0x40;

    lowerB = (val & ~0xf0);
    highB = ((val & ~0xf) >> 4);

    GPIOWrite(GPIOA, 11, false);    // CS = low

    * ((__IO uint8_t *) & SPI1->DR) = (uint8_t) startB; // SPI1->DR = startB;
    while (SPI_IS_BUSY(SPI1)) // POZOR - BLOKUJICI FUNKCE
        ;

    * ((__IO uint8_t *) & SPI1->DR) = (uint8_t) lowerB; // SPI1->DR = lowerB;
    while (SPI_IS_BUSY(SPI1)) // POZOR - BLOKUJICI FUNKCE
        ;

    * ((__IO uint8_t *) & SPI1->DR) = (uint8_t) highB; // SPI1->DR = highB;
    while (SPI_IS_BUSY(SPI1)) // POZOR - BLOKUJICI FUNKCE
        ;

    GPIOWrite(GPIOA, 11, true);     // CS = high
}
```

#### 3.5.3 Inicializace displeje

Po připojení displeje, resp. na začátku programu je nejprve potřeba displej inicializovat nastavením některých registrů. Co bylo jakými příkazy nastaveno je okomentováno v kódu. Toto má na starosti inicializační funkce `LCD_init()`:

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

```
void LCD_init(void)
{
    LCD_reset();

    Nucleo_SetPinGPIO(LCD_GREEN, ioPortOutputPP); // CE
    GPIOWrite(GPIOA, 7, true); // false = sviti, true = nesviti
    Nucleo_SetPinGPIO(LCD_WHITE, ioPortOutputPP); // CW
    GPIOWrite(GPIOA, 6, true); // false = sviti, true = nesviti
    /*
    Nucleo_SetPinGPIO(LCD_RED, ioPortOutputPP); // CR
    GPIOWrite(GPIOA, 2, true); // false = sviti, true = nesviti
    // nelze pouzít - konflikt s USART2
    */

    uint8_t InitData[] = {
        0x3A, // 8 bit data length extension Bit RE=1; REV=0
        0x09, // 4 line display
        0x06, // bottom view
        0x1E, // BS1=1
        0x39, // 8 bit data length extension Bit RE=0; IS=1
        0x1B, // BS0=1 -> Bias=1/6
        0x6C, // Devider on and set value
        0x54, // Booster on and set contrast (DB1=C5, DB0=C4)
        0x7A, // Set contrast (DB3-DB0=C3-C0)
        0x38, // 8 bit data length extension Bit RE=0; IS=0
        0x0F // Display on, cursor on, blink on
    };

    uint8_t i;
    for(i=0; i<sizeof(InitData); i++)
    {
        LCD_send(InitData[i], 0);
    }

    LCD_clear_all();
}
```

Je možné si všimnout, že není využité červené podsvícení displeje. Je to způsobeno hardwarovou chybou při návrhu desky, kdy je pin v konfliktu s pinem USART2, kterým mikrokontroler komunikuje s PC. Pro případnou druhou verzi přístroje je toto třeba vyřešit připojením červeného podsvícení k jinému volnému pinu mikrokontroleru.

Další funkce týkající se LCD displeje jsou k dispozici v 7. Příloha – programový kód v jazyku C.

#### 3.6 Návrh schématu a desky plošných spojů

Schéma zapojení i návrh desky plošných spojů byly navrhovány v programu *EAGLE* od společnosti *Autodesk*. Celé schéma zapojení je součástí přílohy č. 3. Příloha – schéma zapojení navrženého měřidla. Důležité části schématu jsou popsány v příslušných kapitolách s dotčenými komponentami.



### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

---

Deska byla navrhována jako dvouvrstvá a byla vyrobena z materiálu *FR-4*. Plošný spoj byl frézován na fréze KEI FEL. Rozměry desky jsou 98 x 98 mm. Je na ní několik konstrukčních otvorů, kterými bude deska přišroubována do konstrukční krabičky. Na desce je patrná izolační bariéra, která odděluje měřicí obvod od řídicí části s mikrokontrolerem a komunikací s PC. Přes tuto bariéru jsou osazeny dva SMD optočleny typu *HCPL-070L-000E* a SMD izolační step-up DC/DC měnič.

Kromě těchto dvou SMD komponent byly osazovány pouze vývodové součástky. Obvod *MCP39F511A* je zapouzdřen do pouzdra *QFN-28*, což je velmi miniaturní pouzdro a velmi špatně se součástka pájí. Proto byl nechán zapájet na KEI FEL do konverzní destičky, která pouzdro převádí na *DIP-28* kompatibilní vývody.

Jak již bylo zmíněno výše, proudový senzor s Hallovou sondou *ACHS-7121* byl pořízen na již zapájené desce *POLOLU-4030*, čímž jednak odpadá nutnost pájet SMD obvod a jednak se tímto sendvičovým řešením otevírá možnost výměny senzoru za senzor s jinou hodnotou rozsahu měřeného proudu (např. na měření proudů  $\pm 30$  A pomocí desky *POLOLU-4032* s obvodem *ACHS-7123*).

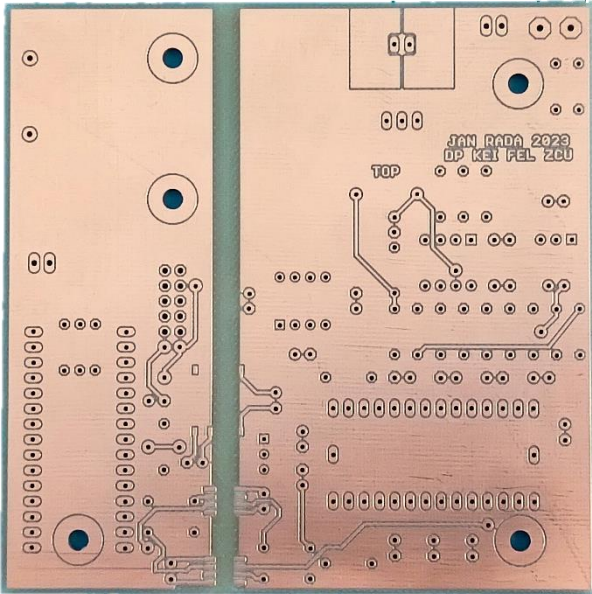
Deska je mj. osazena pouzdrem na 3 AAA monočlánky, které mají celkové napětí 4,5 V. To umožňuje přístroji fungovat nezávisle na připojení USB konektoru. Alternativou by mohlo být napájení z měřeného obvodu tak, jak je uvedeno v příkladu od výrobce v datasheetu viz 1. Příloha – doporučené zapojení obvodu *MCP39F511A* [22]. Toto řešení je ale spíše vhodnější pro aplikace, kdy se měření provádí pouze na síťovém napětí. Pro účely takového měřicího přístroje je vhodnější bateriového napájení. [22]

Napětí z baterie je přes panelový vypínač přivedeno na pin VIN desky *Nucleo32-F303K8*. Díky LDO typu *LD39050PU33R*, kterým je deska osazena, se napětí stabilizuje na 3,3 V, které je pak vyvedeno na pin +3V3. Deska *Nucleo32-F303K8* má současně vyřešeno přepínání napájení mezi baterií a USB blokovacími diodami: pokud je připojeno napájení přes USB, baterie je odpojena a deska je napájena z USB, pokud napájení přes USB připojeno není, je deska napájena z baterie (je-li k dispozici). Proud LDO může být až 500 mA, což je pro účely toho přístroje dostačující. [26, 35]

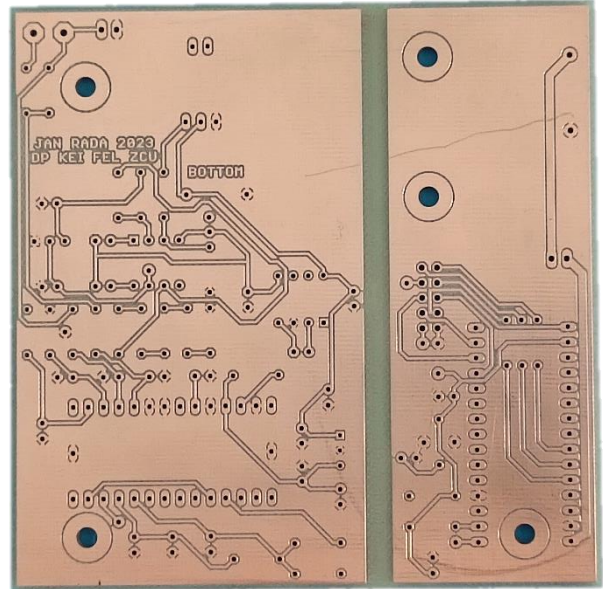
Displej je umístěn v konstrukčním otvoru krabičky, proto je připojen pomocí 10žilového plochého vodiče a rastrového prořezávaného konektoru. Na desce je dále umístěna svorkovnice pro napěťový vstup.

Fotografie vyfrézované desky jsou na *Obr. 34* a *Obr. 35*, fotografie osazené desky je na *Obr. 36*. Návrh desky je dostupný v 4. Příloha – návrh desky plošných spojů.

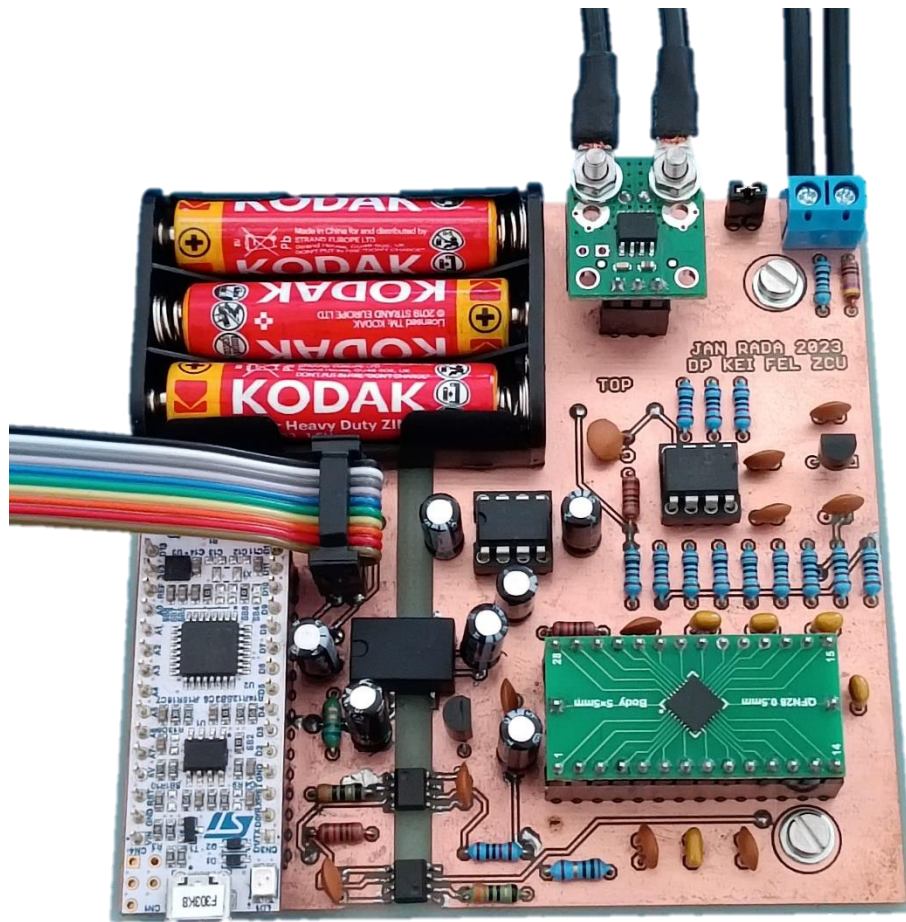
### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU



Obr. 34: Fotografie vyrobené, neosazené desky – strana TOP



Obr. 35: Fotografie vyrobené, neosazené desky – strana BOTTOM



Obr. 36: Fotografie osazené desky plošných spojů

### 3.7 Naprogramování zařízení

Celý programový kód v jazyku C je dostupný v 7. Příloha – programový kód v jazyku C. Program byl vytvářen v prostředí *STM32CubeIDE*. Program se skládá z několika souborů. Seznam zdrojových souborů je v tabulce *Tab. 9*.

Soubor	Obsah souboru
main.c	Hlavní soubor s hlavní smyčkou programu
my_nucleo.c	Funkce týkající se desky <i>Nucleo32-F303K8</i> , jako je práce s GPIO a s USART2 periferií, kterou je možné komunikovat s PC
PWR_meter.c	Funkce týkající se obvodu <i>MCP39F511A</i> – příprava a zpracování zpráv pro periferii USART1, která měřicím IO komunikuje
EADOGS_164.c	Funkce týkající se displeje <i>EADOGS164W-A</i> a periferie SPI1, která s displejem komunikuje

Tab. 9: Výčet zdrojových souborů pro Nucleo32-F303K8

### 3.8 Oživení přístroje

Během ožívání osazené desky se vyskytlo několik problémů. Hlavní problém, který zabral hodně času, než byl odhalen, byl nefunkční vnitřní oscilátor měřicího obvodu *MCP39F511A*. Vnitřní oscilátor se nedá nijak spustit nebo vypnout, je spouštěn automaticky, pokud není detekován vnější oscilátor, ten však v našem případě nebyl připojen. Na diskusním fóru na stránkách výrobce bylo zjištěno, že příčinou nefunkčního oscilátoru mohou být přívodní cesty k pinům *OSCI* a *OSCO*, do kterých se může indukovat rušení a obvod jej pak rozpozná jako připojený oscilátor a nespustí tak vnitřní oscilátor. Toto bylo vyřešeno připájením krystalového oscilátoru 4 MHz přímo na piny *OSCI* a *OSCO*. Obvod *MCP39F511A* pak již fungoval a odpovídal, jak má.

Další problém byl s komunikací přes optočleny *HCPL-070L-000E*. Velmi dlouhé časy náběžných a doběžných hran výstupního signálu způsobovali takovou deformaci, že komunikace nemohla být rozpoznána. Toto bylo vyřešeno postupným snižováním pull-up rezistorů nejprve z hodnoty 10 k $\Omega$  na 2,2 k $\Omega$  a poté ještě snížením na 1,1 k $\Omega$ . Při této hodnotě pull-up rezistoru již komunikace funguje bez problémů. Pomocí osciloskopu však ale bylo zjištěno, že průběh signálu je i tak velmi deformován.

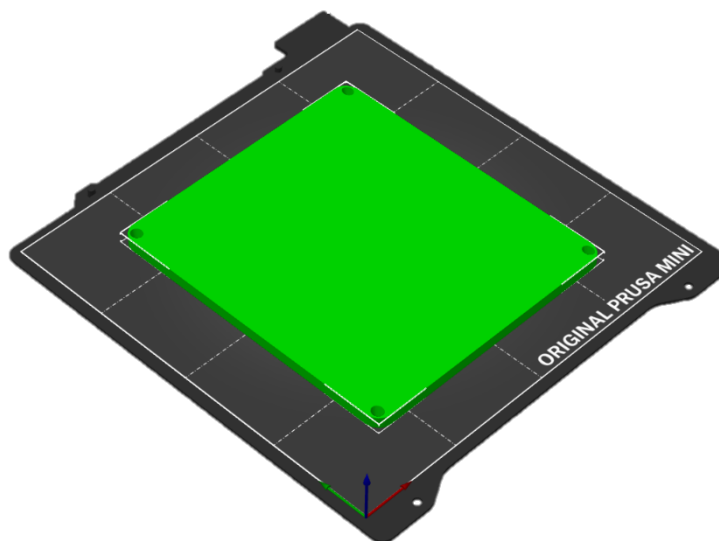
### 3.9 Konstrukční krabička

Aby bylo dosaženo kompaktních rozměrů, nebyla konstrukční krabička zařízení kupována, ale byla vymodelována a následně vytištěna na 3D tiskárně. 3D tisk je velmi zajímavý a rychle se rozvíjející obor, který mě během zpracovávání této diplomové práce oslovil a možná se i stane mým novým koníčkem.

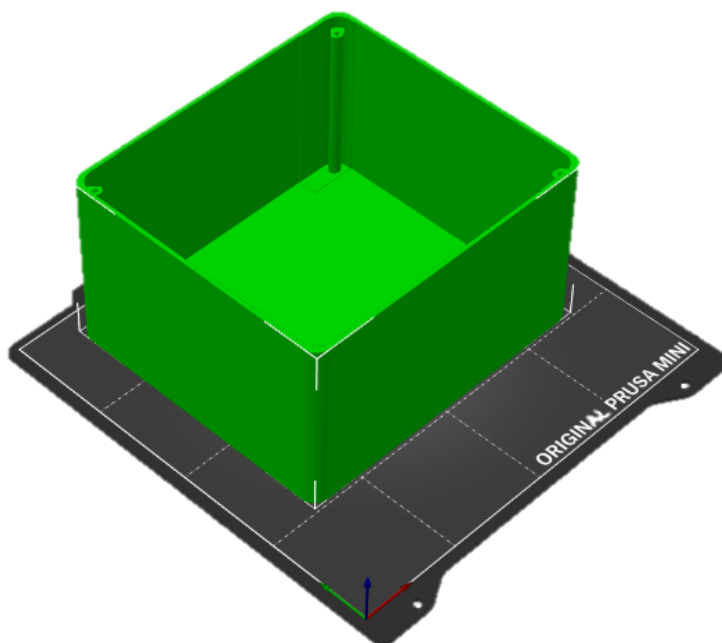
Model byl navržen v CAD softwaru *TinkerCAD* od společnosti *Autodesk*, následně byl model vyslicován softwarem *PrusaSlicer* a vytištěn pomocí 3D tiskárny *Prusa mini*. Krabička má vnější rozměry šířka x hloubka x výška 139 x 125 x 78 mm, což jsou rozměry pro potřebu tohoto zařízení ideální. Kompletní dokumentace krabičky je dostupná v 5. Příloha – výkres navržené konstrukční krabičky.

K tisku byl použit filament z materiálu PLA. Tento materiál je společně s materiálem ABS pro 3D tisk nejpoužívanější. Tisk z tohoto materiálu je velmi jednoduchý a je tak vhodný a doporučený pro začátečníky v tomto mladém oboru 3D tisku. PLA, neboli kyselina polyléčná, je biologicky plně odbouratelný materiál. Vyrábí se z kukuřičného nebo bramborového škrobu či z cukrové řepy, není tedy ropným produktem a je tak zároveň šetrný k životnímu prostředí. [36]

Na obrázcích *Obr. 37* a *Obr. 38* jsou zobrazeny modely krabičky a na obrázcích *Obr. 39* a *Obr. 40* jsou fotografie vytištěných výrobků.



Obr. 37: Model víčka krabičky v softwaru PrusaSlicer



Obr. 38: Model krabičky v softwaru PrusaSlicer



Obr. 39: Fotografie výtisku konstrukční krabičky



Obr. 40: Fotografie výtisku konstrukční krabičky – víčko

#### 3.10 Kompletace výrobku

Model a výtisk krabičky byl proveden bez montážních otvorů. Díry pro svorky a pojistku a okénka pro vypínač, USB konektor a displej byly až následně vyvrtány a vyřezány. Horní strana krabičky byla polepena odolným samolepicím papírem s potiskem viz 6. *Příloha – potisk krabičky*. Na horní stranu krabičky byly umístěny 4 zdířky pro banánky o průměru 4 mm s možností přišroubování vodiče. Zásuvka je odolná proudu až do 50 A. Tyto svorky jsou určeny pro připojení měřicího přístroje do měřeného obvodu: dvě jsou napěťové a dvě proudové. [37]

### 3. REALIZACE ELEKTRONICKÉHO WATTMETRU S VYUŽITÍM INTEGROVANÉHO OBVODU PRO MĚŘENÍ VÝKONU

Dále je zde vypínač napájení z baterie. Napájení z USB není možné vypínačem vypnout – jestliže je USB kabel přípojek k napájení, přístroj je zapnutý. USB konektor se nachází z boku přístroje. Konečně je na horní straně přístroje umístěný LCD displej pro zobrazení měřených veličin.

Kompletní finální výrobek je na obrázcích *Obr. 41*, *Obr. 42* a *Obr. 43*.



Obr. 41: Fotografie hotového výrobku – pohled shora



Obr. 42: Fotografie hotového výrobku – pohled z boku



Obr. 43: Fotografie hotového výrobku s připojeným USB kabelem – izometrický pohled

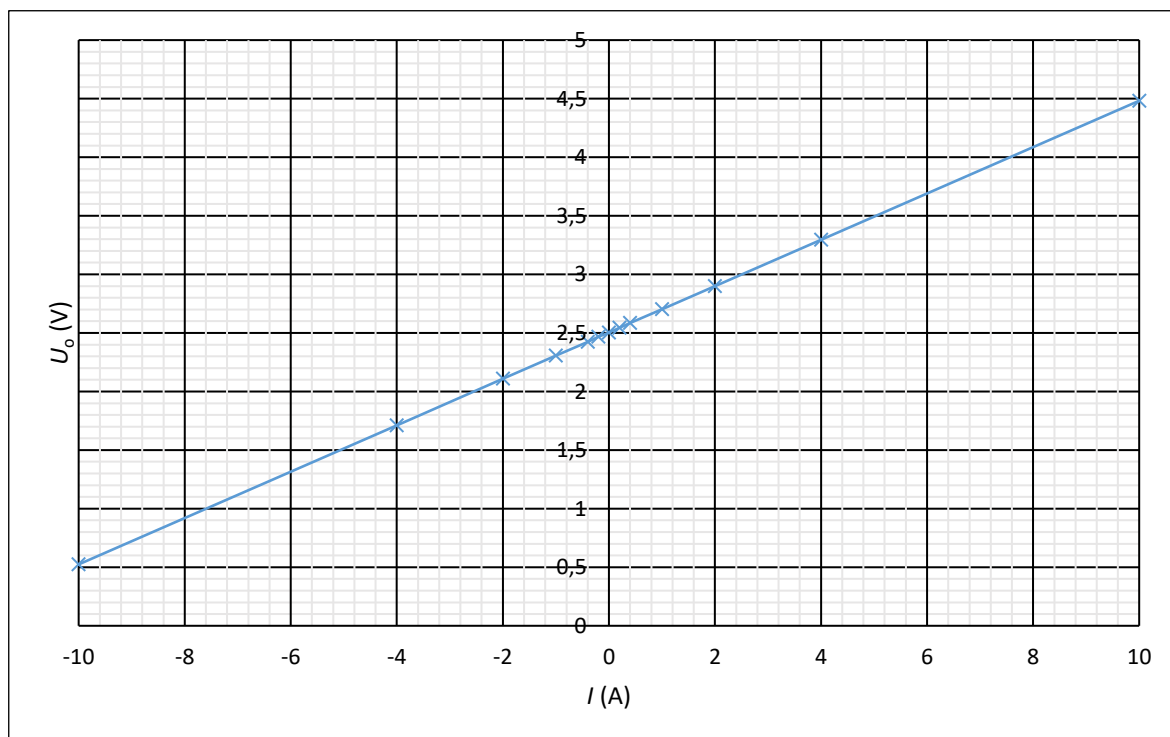
## 4 Ověření funkcí navrženého a realizovaného elektronického wattmetru

### 4.1 Proudový senzor s Hallovou sondou

Nejprve byla ověřena funkce proudového senzoru s Hallovou sondou *ACHS-7121*. Sondou byl protékán stejnosměrný proud přes celý měřitelný rozsah sondy a bylo zkoumáno výstupní napětí. Naměřené hodnoty jsou dostupné v *TAB* a výsledky byly vyneseny do grafu na *Obr. 44*, kde  $I$  je proud sondou a  $U_o$  je výstupní napětí. Bylo ověřeno, že sonda má opravdu citlivost 185 mV/A a funguje přesně podle předem očekávaných parametrů stanovené výrobcem. [27]

<b>I (A)</b>	-10	-4	-2	-1	-0,4	-0,2	0	0,2	0,4	1	2	4	10
<b>U<sub>o</sub> (V)</b>	0,524	1,711	2,108	2,306	2,423	2,466	2,504	2,543	2,586	2,701	2,899	3,296	4,482

Tab. 10: Naměřené hodnoty při ověřování funkce proudového senzoru ACHS-7121

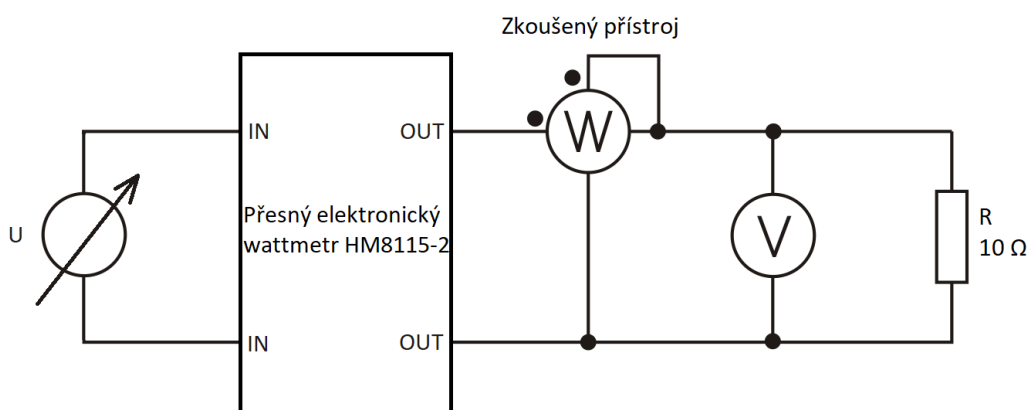


Obr. 44: Závislost výstupního napětí na proudu sondou ACHS-7121

## 4. OVĚŘENÍ FUNKCÍ NAVRŽENÉHO A REALIZOVANÉHO ELEKTRONICKÉHO WATTMETRU

### 4.2 Schéma zapojení měření

Všechna následující měření probíhala se zapojením viz *Obr. 45*. Jako přesný referenční měřič byl použit elektronický programovatelný wattmetr *HAMEG HM8115-2*. Přístroj měří napětí, proud, činný výkon, jalový výkon a zdánlivý výkon. Z přístroje byly odečítány hodnoty proudu a hodnoty výkonu, napětí bylo odečítáno na voltmetru *METEX-P10*. Jako zátěž byla použita odporová zátěž  $10\ \Omega$ . Pro stejnosměrné měření byl použit zdroj *Manson SPS9600*, pro střídavé měření autotransformátor *METREL HSN 0203*.



Obr. 45: Schéma zapojení ověření funkce vyrobeného wattmetru

### 4.3 Stejnosměrné měření na vyrobeném přístroji

Naměřené hodnoty z prvního stejnosměrného měření jsou dostupné v *Tab. 11*, graf závislosti  $I_o = f(I_n)$  na *Obr. 46* a graf závislosti  $\Delta I = f(I_o)$  na *Obr. 47*. Jako  $X_n$  je označovaná naměřená normálová hodnota veličiny (přesná),  $X_o$  je označovaná naměřená veličina zkoušeným přístrojem. Absolutní chyba měření byla počítána podle vztahu:

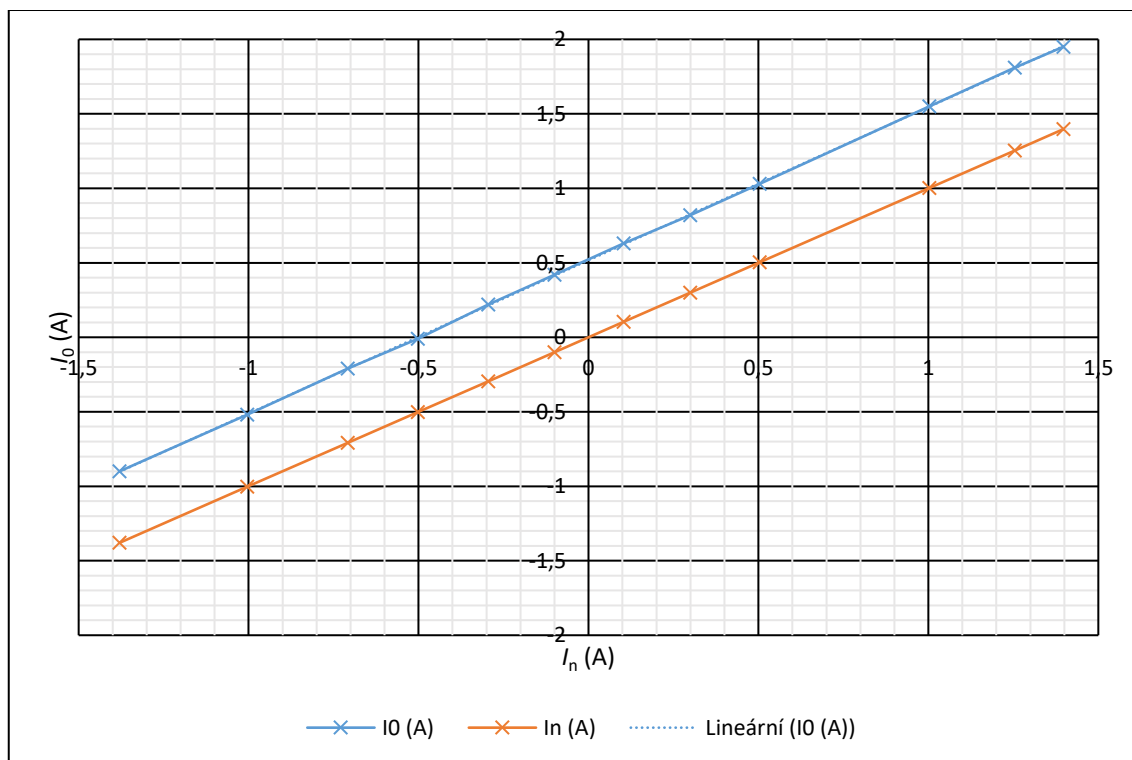
$$\Delta X = X_o - X_n \text{ (jedd.)} \quad (8)$$

$U_n$ (V)	-13,8	-10	-7	-5	-3	-1	1	3	5	10	12,5	13,9
$U_o$ (V)	-13	-9	-6	-4	-2	0	2	4	6	10	13	14
$\Delta U$ (V)	0,8	1	1	1	1	1	1	1	1	0	0,5	0,1
$I_n$ (A)	-1,38	-1,004	-0,708	-0,502	-0,295	-0,1	0,103	0,299	0,504	1,003	1,254	1,397
$I_o$ (A)	-0,9	-0,52	-0,21	-0,01	0,22	0,42	0,63	0,82	1,03	1,55	1,81	1,95
$\Delta I$ (A)	0,48	0,484	0,498	0,492	0,515	0,52	0,527	0,521	0,526	0,547	0,556	0,553
$P_n$ (W)	19,11	10,09	4,98	2,5	0,88	0,1	0,108	0,94	2,52	10,08	15,66	19,32
$P_o$ (W)	10	4	1	0	0	0	1	3	5	14	21	25
$\Delta P$ (W)	-9,11	-6,09	-3,98	-2,5	-0,88	-0,1	0,892	2,06	2,48	3,92	5,34	5,68

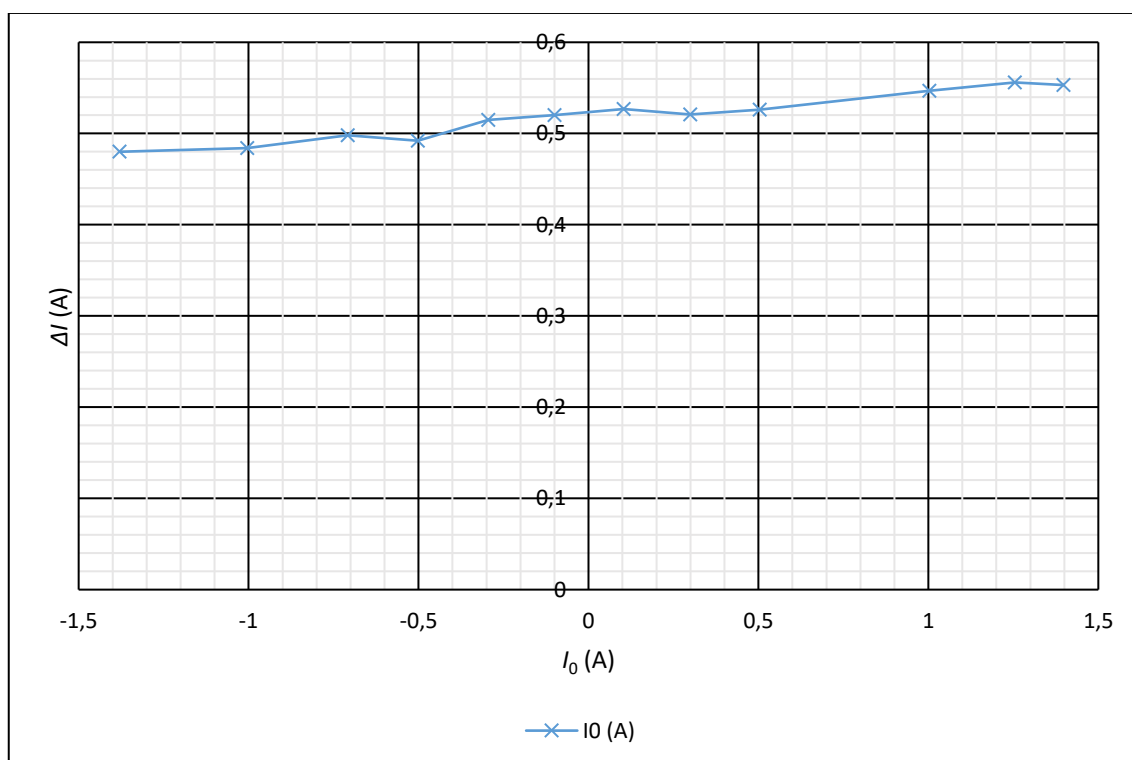
Tab. 11: Naměřené hodnoty s chybou měření proudu



#### 4. OVĚŘENÍ FUNKCÍ NAVRŽENÉHO A REALIZOVANÉHO ELEKTRONICKÉHO WATTMETRU



Obr. 46: Závislost naměřeného proudu zkoušeným měřidlem na proudu měřeným přesným měřidlem

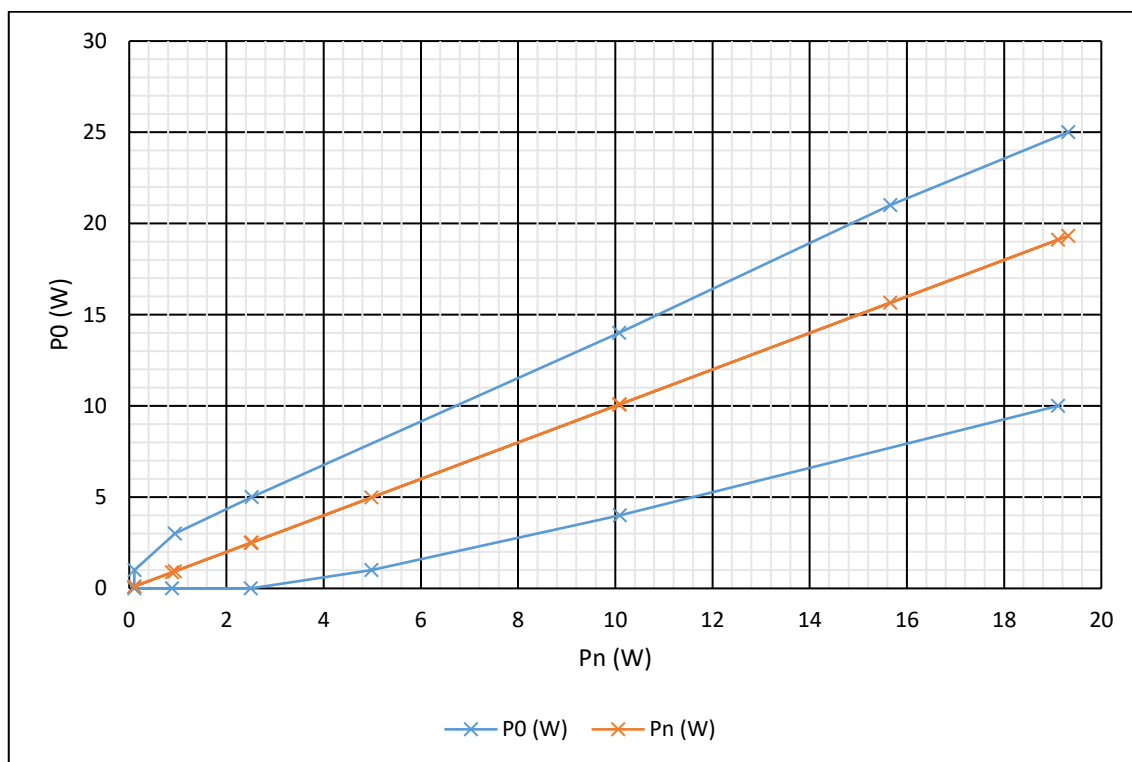


Obr. 47: Závislost absolutní chyby měření na měřeném proudu

#### 4. OVĚŘENÍ FUNKCÍ NAVRŽENÉHO A REALIZOVANÉHO ELEKTRONICKÉHO WATTMETRU

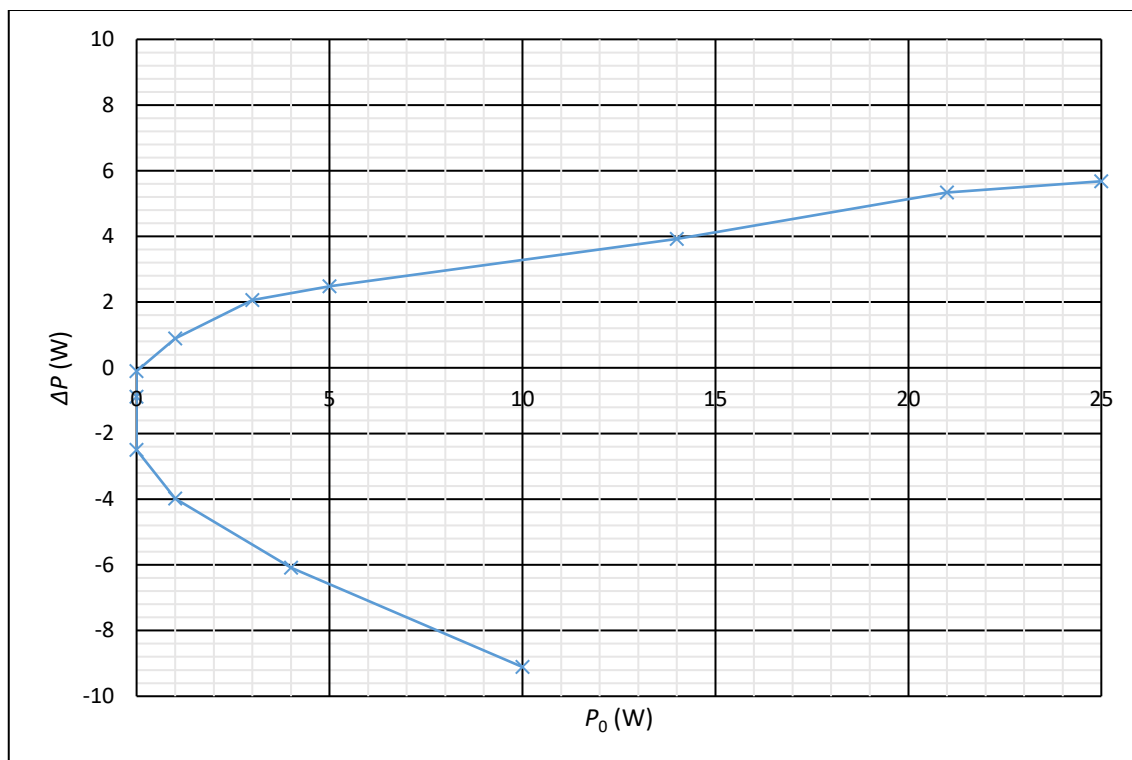
První měření, které proběhlo na hotovém výrobku neproběhlo s dobrým výsledkem. Důsledkem nedostatečného napájení proudové sondy, kdy byla napájena jen asi 4,8 V, nebyl střed výstupního napětí přesně v 2,5 V a tím vznikla chyba při měření proudu s offsetem asi 0,5 A. Touto chybou byl následně zatížen i výpočet výkonu viz *Obr. 48* se zobrazením závislosti  $P_0 = f(P_n)$  a *Obr. 49* se zobrazením závislosti  $\Delta P = f(P_0)$ .

Pro potlačení chyby bylo na proudový senzor přivedeno přesné napájení 5 V z externího stejnosměrného zdroje.



Obr. 48: Závislost naměřeného výkonu zkoušeným měřidlem na výkonu měřeným přesným měřidlem

#### 4. OVĚŘENÍ FUNKCÍ NAVRŽENÉHO A REALIZOVANÉHO ELEKTRONICKÉHO WATTMETRU



Obr. 49: Závislost absolutní chyby měření výkonu na měřeném výkonu

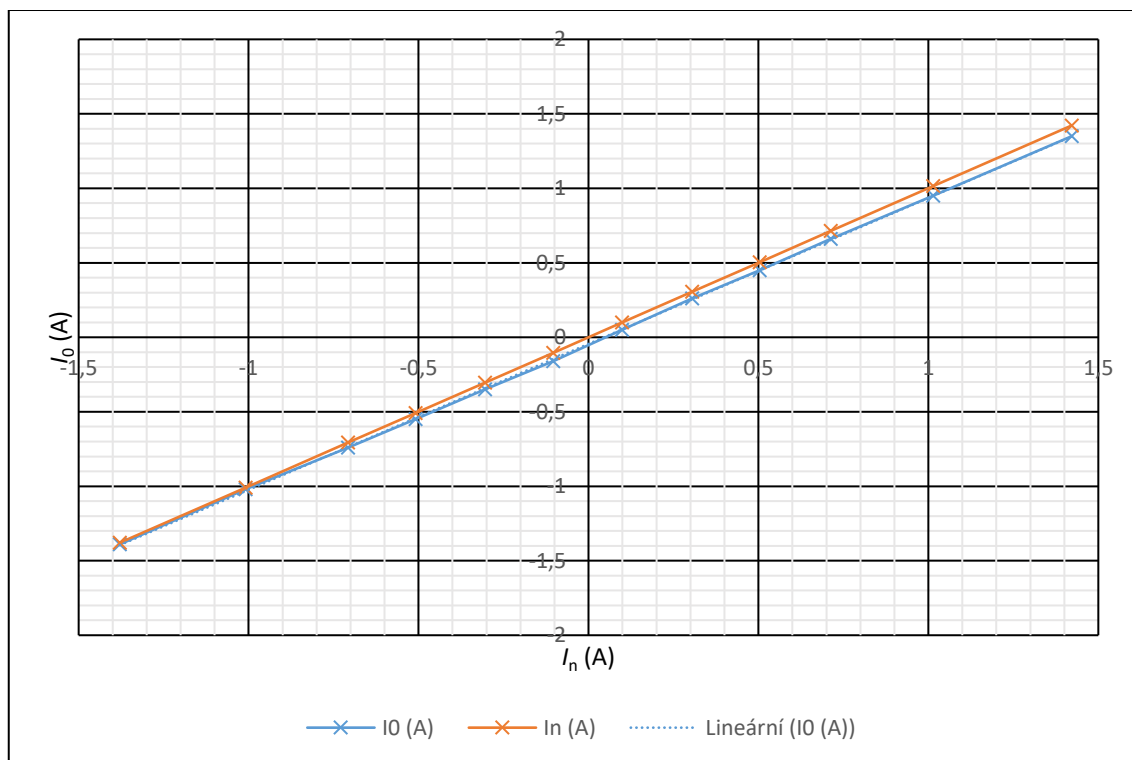
#### 4.4 Stejnoseměrné měření na vyrobeném přístroji s externím napájením proudového senzoru

Druhé měření bylo provedeno stejným způsobem, jako v předchozím případě. Měření na přístroji bylo po připojení externího napájení pro proudový senzor o poznání lepší. Přesto ještě nebylo dosaženo ideálních výsledků. Naměřené hodnoty jsou v *Tab. 12*.

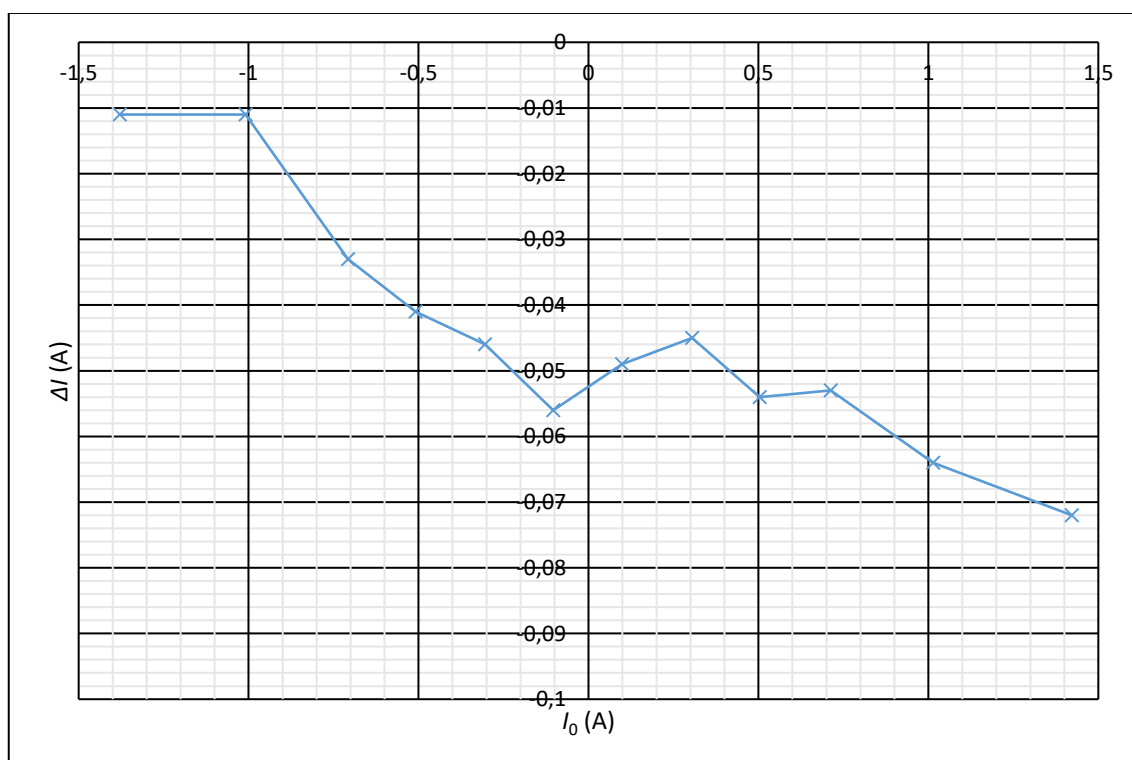
$U_n$ (V)	-13,7	-10	-7	-5	-3	-1	1	3	5	7	10	14
$U_o$ (V)	-13	-9	-6	-4	-2	0	2	4	6	8	11	14
$\Delta U$ (V)	0,7	1	1	1	1	1	1	1	1	1	1	0
$I_n$ (A)	-1,379	-1,009	-0,707	-0,509	-0,304	-0,104	0,099	0,305	0,504	0,713	1,014	1,422
$I_o$ (A)	-1,39	-1,02	-0,74	-0,55	-0,35	-0,16	0,05	0,26	0,45	0,66	0,95	1,35
$\Delta I$ (A)	-0,011	-0,011	-0,033	-0,041	-0,046	-0,056	-0,049	-0,045	-0,054	-0,053	-0,064	-0,072
$P_n$ (W)	18,93	10,18	5,01	-2,62	0,99	0,11	0,096	0,91	2,51	5,03	10,17	19,82
$P_o$ (W)	16	9	4	2	1	0	0	1	2	5	9	17
$\Delta P$ (W)	-2,93	-1,18	-1,01	4,62	0,01	-0,11	-0,096	0,09	-0,51	-0,03	-1,17	-2,82

Tab. 12: Tabulka naměřených hodnot po připojení externího napájení pro proudový senzor

#### 4. OVĚŘENÍ FUNKCÍ NAVRŽENÉHO A REALIZOVANÉHO ELEKTRONICKÉHO WATTMETRU



Obr. 50: Závislost naměřeného proudu zkoušeným měřidlem na proudu měřeným přesným měřidlem



Obr. 51: Závislost absolutní chyby měření na měřeném proudu

#### 4. OVĚŘENÍ FUNKCÍ NAVRŽENÉHO A REALIZOVANÉHO ELEKTRONICKÉHO WATTMETRU

Měření proudu bylo připojením vnějšího napájení na proudovou sondu výrazně přesnější, ovšem chyba měření výkonu je přesto velmi velká – řádově jednotky wattů. Je proto nutné přistoupit ke zpřesnění kalibrace. Postup kalibrace byl popsán v kapitole 3.1.6 *Kalibrace*.

##### 4.5 Stejnoseměrné měření po kalibraci

Kalibrace byla provedena na přesných hodnotách stejnosměrného napětí, proudu a výkonu. Napětí bylo nastaveno na 10 V, následně byl nastaven proud na 1 A, a nakonec výkon na 10 W. Postup této konkrétní kalibrace je popsán jako příklad v kapitole 3.1.6 *Kalibrace*.

Současně byl upraven výpis měřených hodnot napětí a výkonu na displej tak, aby zobrazoval desetinné místo a porovnávací měření tak mohlo být přesnější.

$U_n$ (V)	-13,7	-9	-6	-3	-1	1	3	6	9	13,7
$U_0$ (V)	-12,3	-7,9	-5	-2,2	-0,2	1,5	3,5	6,3	9,2	13,7
$\Delta U$ (V)	1,4	1,1	1	0,8	0,8	0,5	0,5	0,3	0,2	0
$I_n$ (A)	-1,394	0,915	-0,61	-0,301	-0,101	0,099	0,307	0,614	0,917	1,398
$I_0$ (A)	-1,43	-0,94	-0,63	-0,32	-0,11	0,09	0,3	0,61	0,92	1,4
$\Delta I$ (A)	-0,036	-1,855	-0,02	-0,019	-0,009	-0,009	-0,007	-0,004	0,003	0,002
$P_n$ (W)	19,66	8,57	3,86	0,96	0,108	0,102	0,97	3,8	8,45	19,62
$P_0$ (W)	18	7,6	3,3	0,7	0	0,1	1,1	3,9	8,6	19,5
$\Delta P$ (W)	-1,66	-0,97	-0,56	-0,26	-0,108	-0,002	0,13	0,1	0,15	-0,12

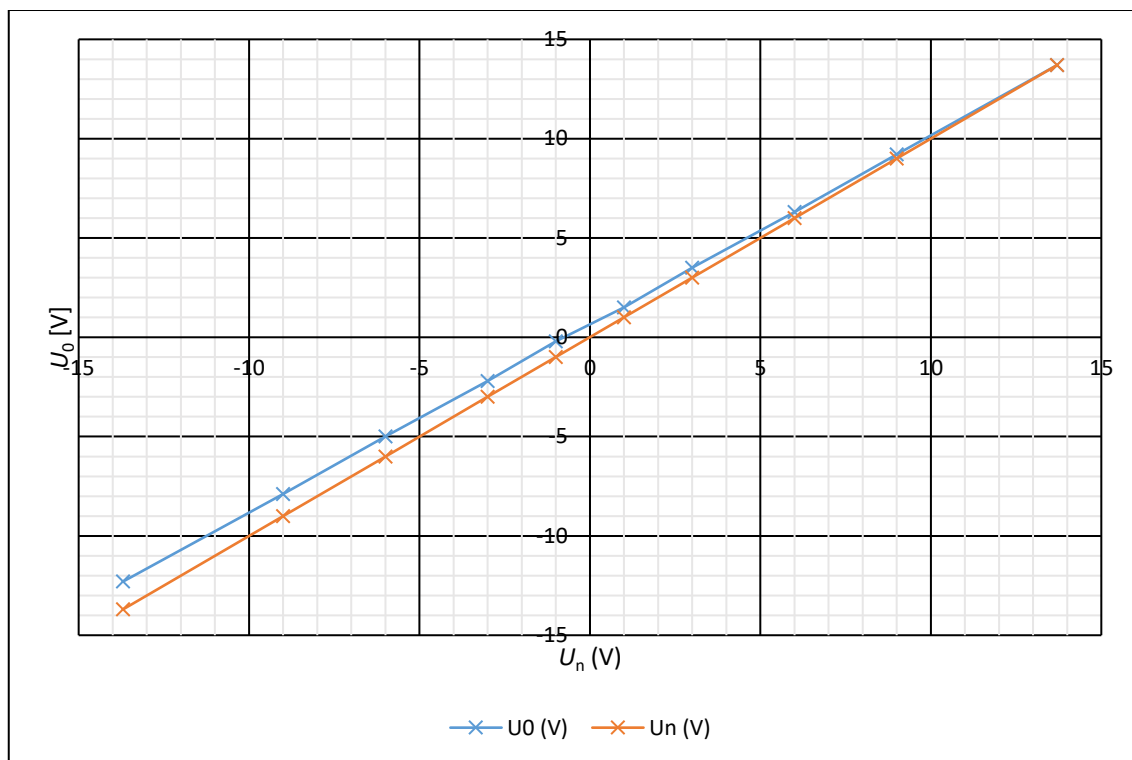
Tab. 13: Tabulka naměřených hodnot po kalibraci

Z naměřených hodnot je vidět, že proud je nyní měřen velmi přesně, tj. v řádu setin ampéru, což je na 10A rozsah relativně dobrý výsledek. Grafy z měření proudu jdou na *Obr. 54* a *Obr. 55*.

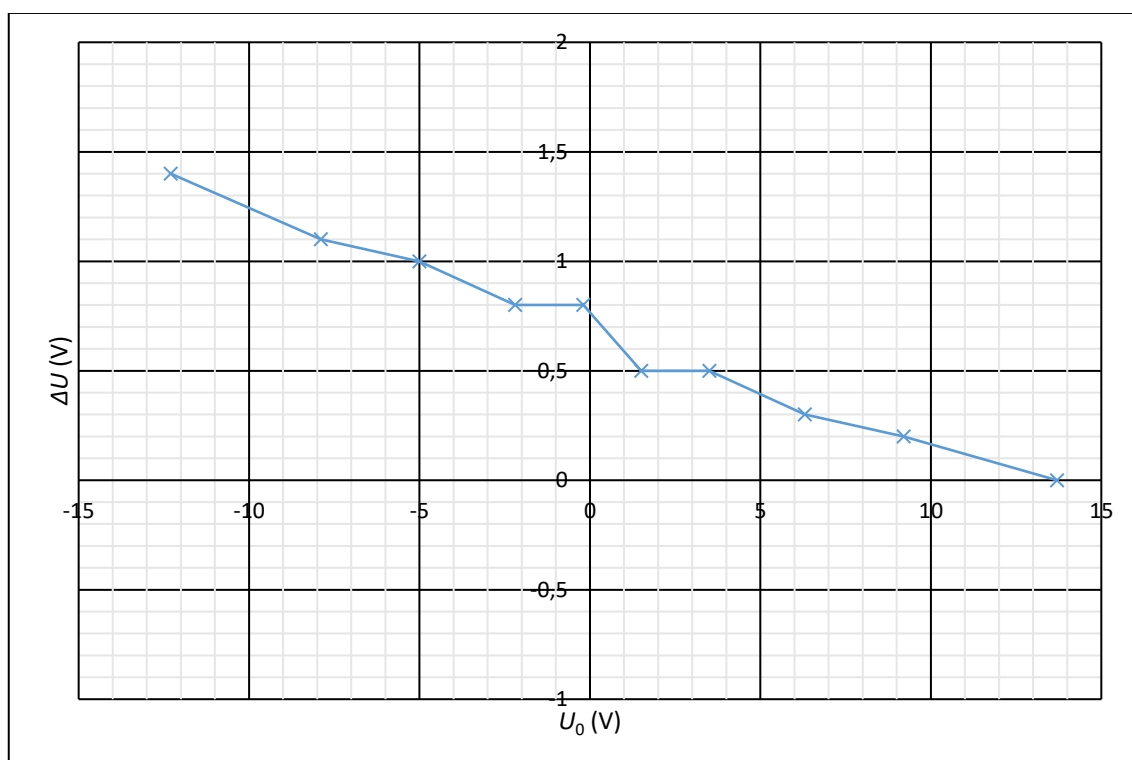
Z měření napětí je vidět, že je zde nežádoucí offset. Toto musí být ještě doladěno a DC offset donastaven v příslušném kalibračním registru. Grafy měření napětí viz *Obr. 52* a *Obr. 53*.

Důsledkem chyby měření napětí je nepřesnost přenášená do výpočtu výkonu. Z hodnot lze vyčíst, že při přesnějších naměřených hodnotách napětí (v tomto případě při kladných napětích) je i měření výkonu relativně přesné. Grafy z měření výkonu jsou na *Obr. 56* a *Obr. 57*.

#### 4. OVĚŘENÍ FUNKCÍ NAVRŽENÉHO A REALIZOVANÉHO ELEKTRONICKÉHO WATTMETRU

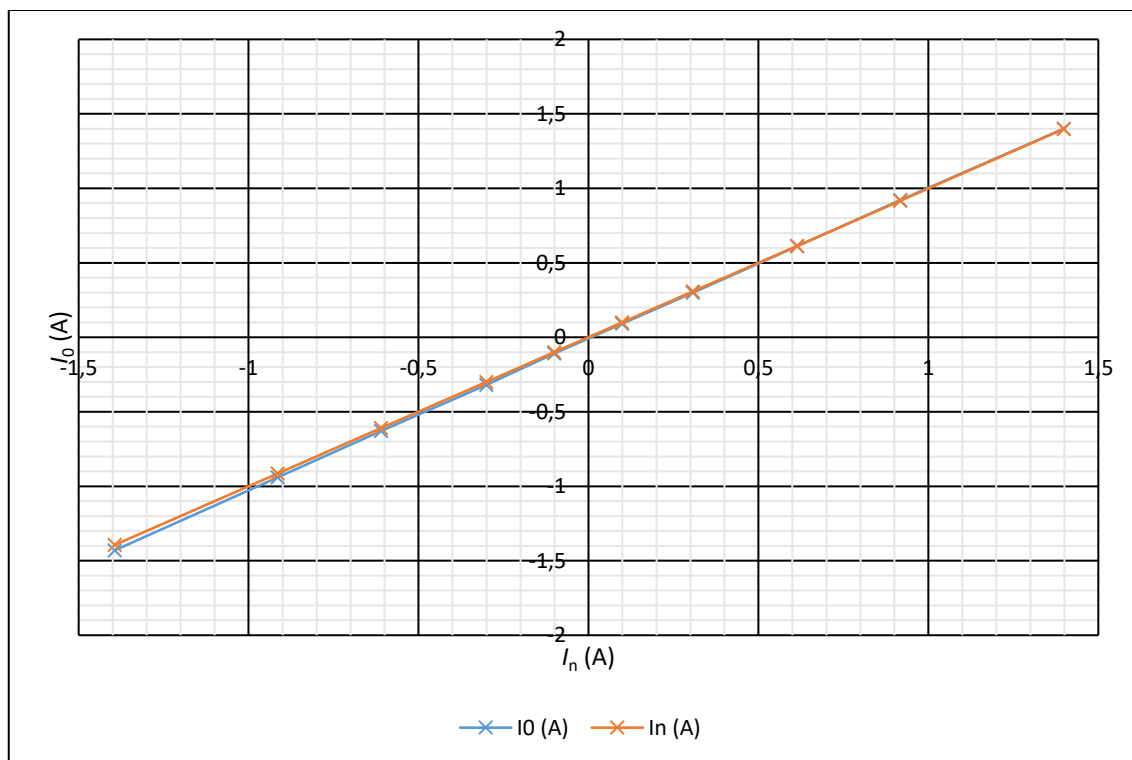


Obr. 52: Závislost naměřeného napětí zkoušeným měřidlem na napětí měřeným přesným měřidlem

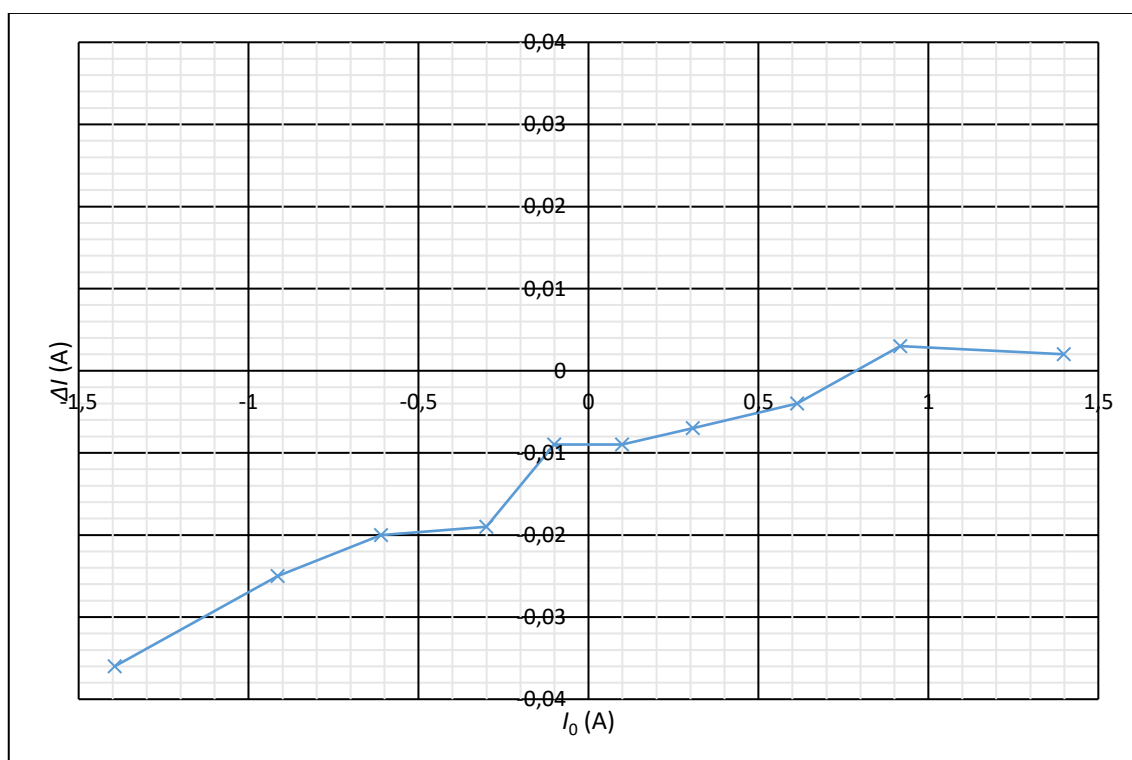


Obr. 53: Závislost absolutní chyby měření na měřeném napětí

#### 4. OVĚŘENÍ FUNKCÍ NAVRŽENÉHO A REALIZOVANÉHO ELEKTRONICKÉHO WATTMETRU

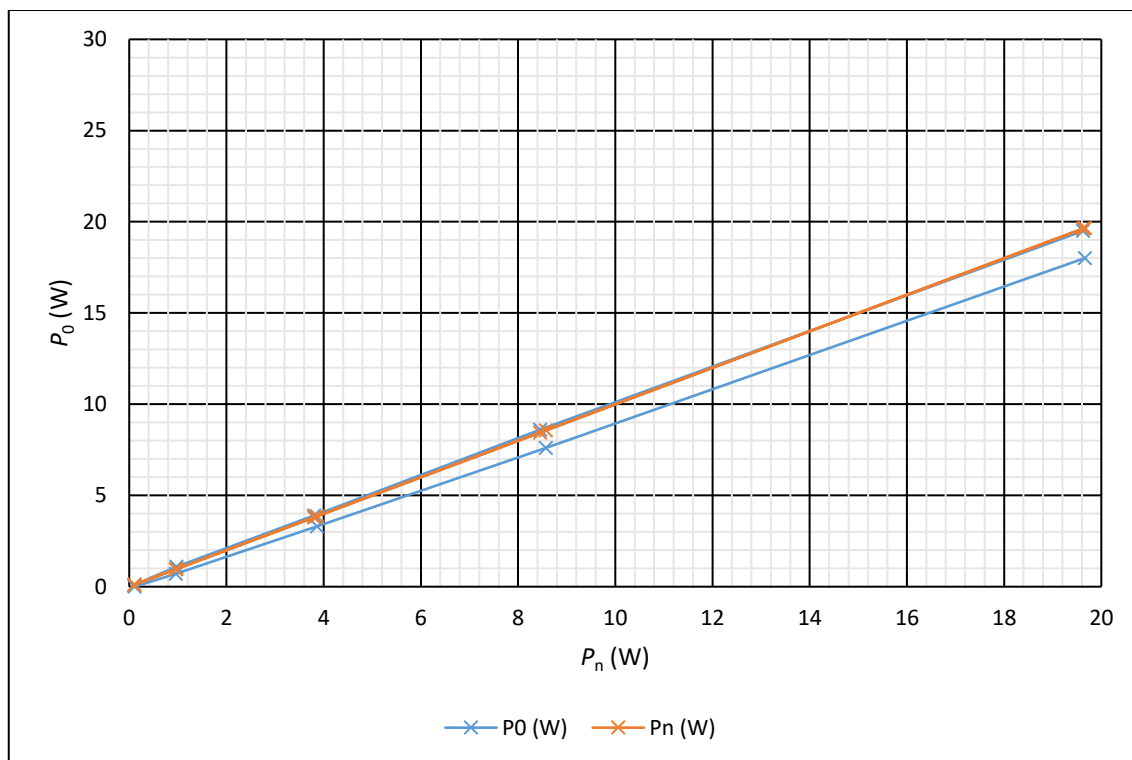


Obr. 54: Závislost naměřeného proudu zkoušeným měřidlem na proudu měřeným přesným měřidlem

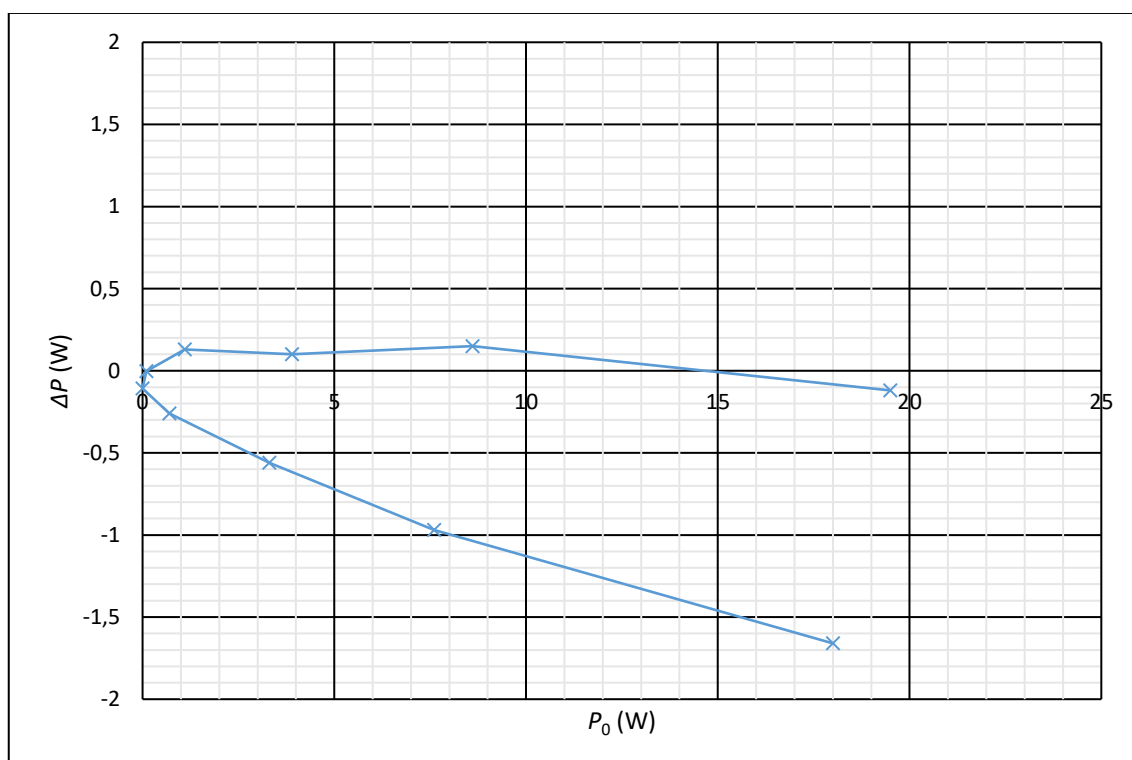


Obr. 55: Závislost absolutní chyby měření na měřeném proudu

#### 4. OVĚŘENÍ FUNKCÍ NAVRŽENÉHO A REALIZOVANÉHO ELEKTRONICKÉHO WATTMETRU



Obr. 56: Závislost naměřeného výkonu zkoušeným měřidlem na výkonu měřeným přesným měřidlem



Obr. 57: Závislost absolutní chyby měření výkonu na měřeném výkonu



#### 4.6 Střídavé měření na vyrobeném přístroji

Zapojení měřicího obvodu zůstává stejné, jen místo stejnosměrného zdroje byl zapojen autotransformátor *METREL HSN 0203*.

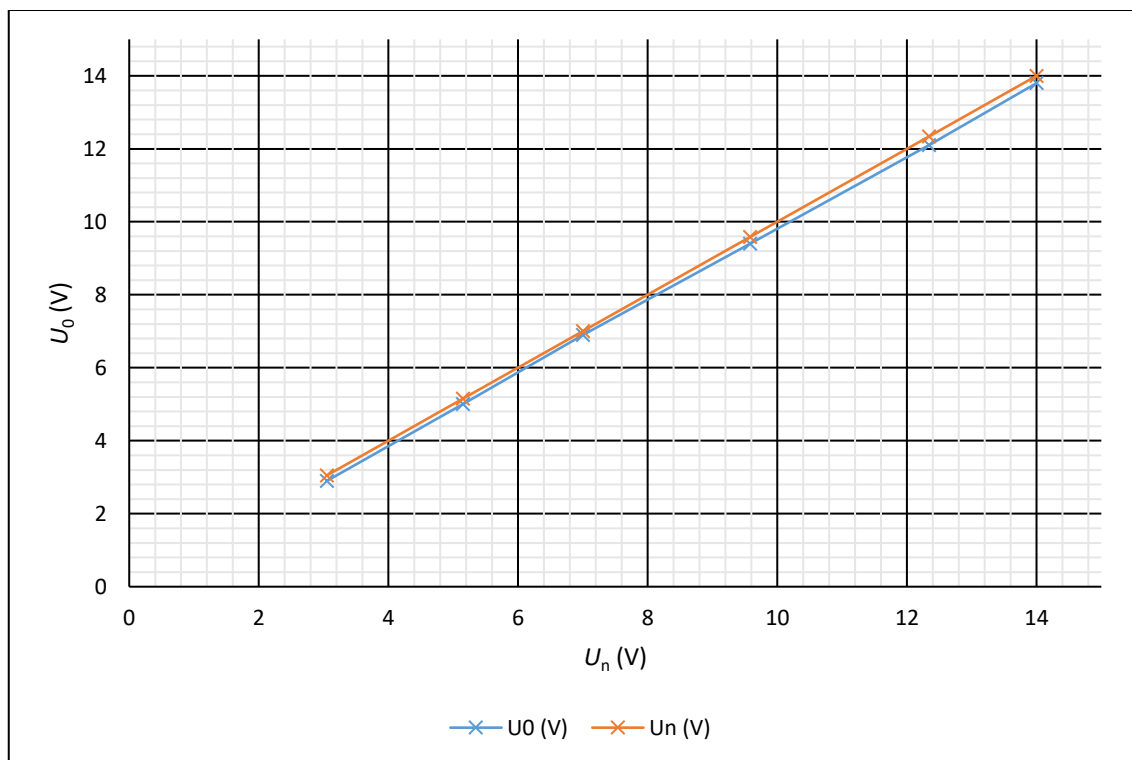
Naměřené hodnoty při měření střídavých veličin jsou v *Tab. 14*. Z hodnot je jasné, že činný výkon  $P_0$  je změřen špatně, a i zde bude potřeba provést kalibraci zisku. Zdánlivý výkon  $S_0$  se křivkou téměř shoduje s normálovým činným výkonem. Je to jednak proto, že měření probíhá na čistě odporové zátěži, a jednak proto, že tento výkon je vypočítáván prostým násobením změřeného napětí a proudu, které byly v tomto případě změřeny poměrně přesně.

$U_n$ (V)	3,05	5,15	7	9,58	12,34	14
$U_0$ (V)	2,9	5	6,9	9,4	12,1	13,8
$\Delta U$ (V)	-0,15	-0,15	-0,1	-0,18	-0,24	-0,2
$I_n$ (A)	0,312	0,527	0,721	0,98	1,263	1,438
$I_0$ (A)	0,32	0,55	0,75	1,02	1,32	1,49
$\Delta I$ (A)	0,008	0,023	0,029	0,04	0,057	0,052
$P_n$ (W)	1,01	2,85	5,31	9,72	16,18	20,84
$P_0$ (W)	0,6	1,7	3,2	5,8	9,8	12,5
$\Delta P$ (W)	-0,41	-1,15	-2,11	-3,92	-6,38	-8,34
$S_0$ (VA)	1	2,7	5,2	9,6	15,9	20,5

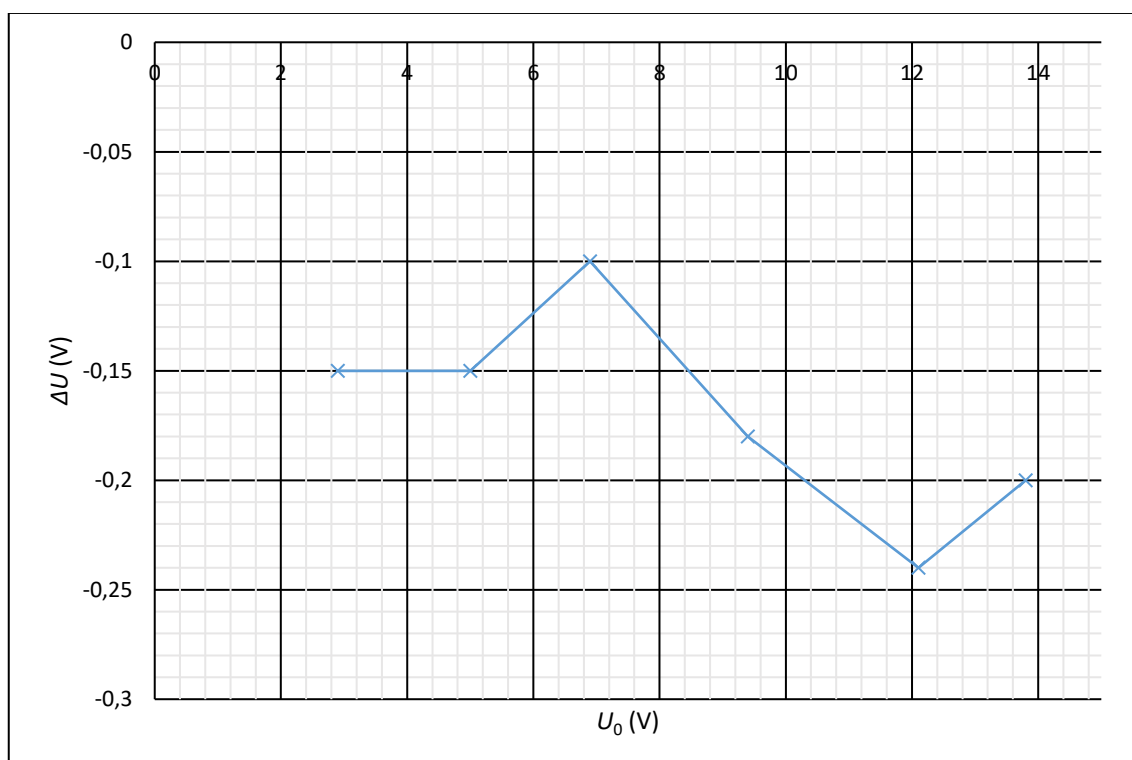
Tab. 14: Naměřené hodnoty při střídavém měření

Na *Obr. 58* a *Obr. 59* jsou grafy měření napětí, na *Obr. 60* a *Obr. 61* jsou grafy měření proudu a na *Obr. 62* a *Obr. 63* jsou grafy měření výkonů.

#### 4. OVĚŘENÍ FUNKCÍ NAVRŽENÉHO A REALIZOVANÉHO ELEKTRONICKÉHO WATTMETRU

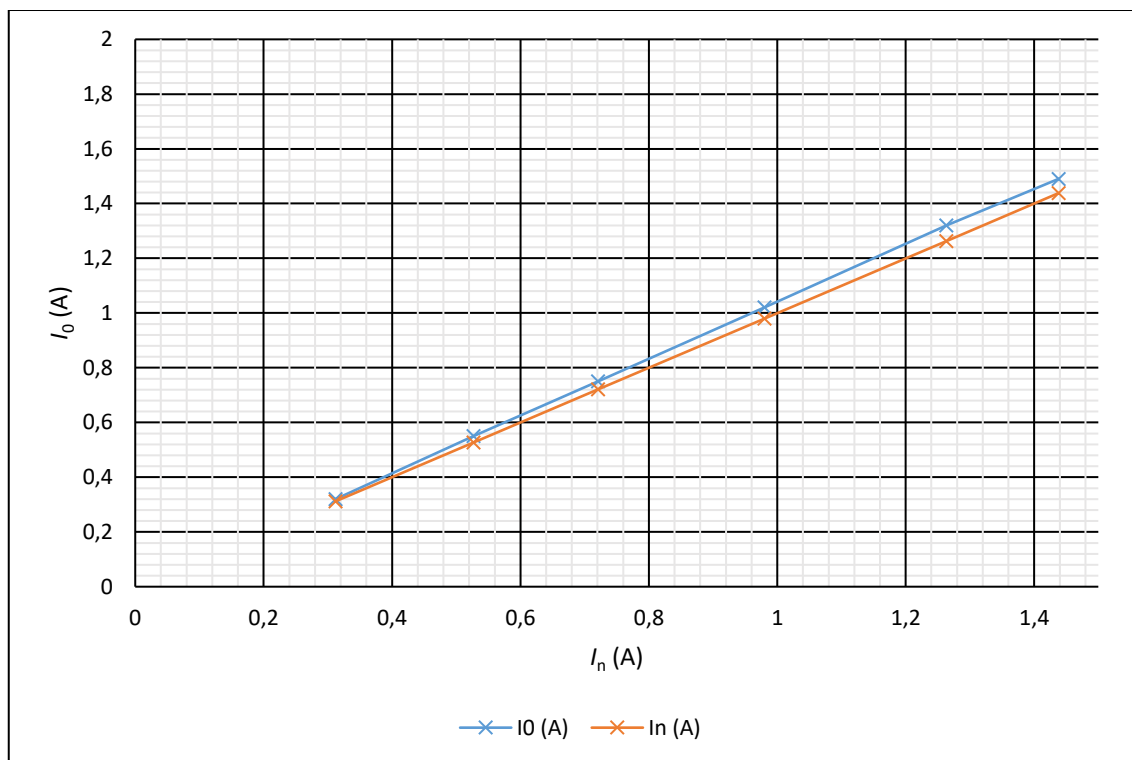


Obr. 58: Závislost naměřeného napětí zkoušeným měřidlem na napětí měřeným přesným měřidlem

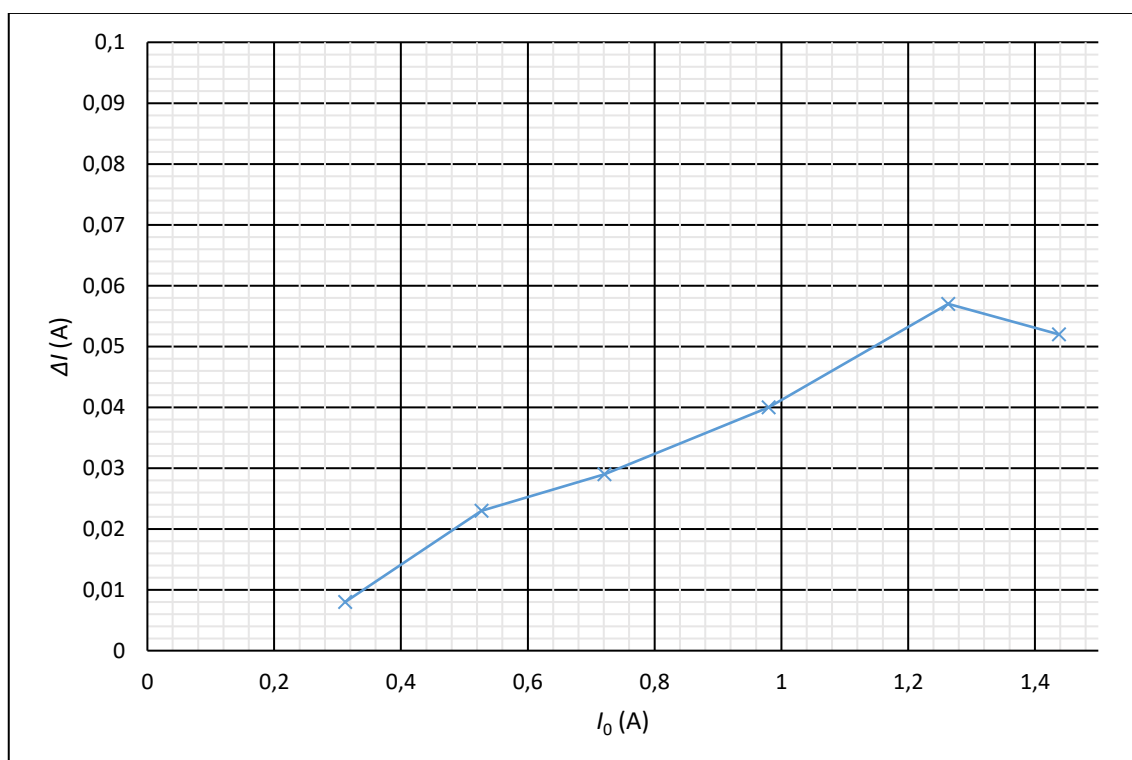


Obr. 59: Závislost absolutní chyby měření na měřeném napětí

#### 4. OVĚŘENÍ FUNKCÍ NAVRŽENÉHO A REALIZOVANÉHO ELEKTRONICKÉHO WATTMETRU

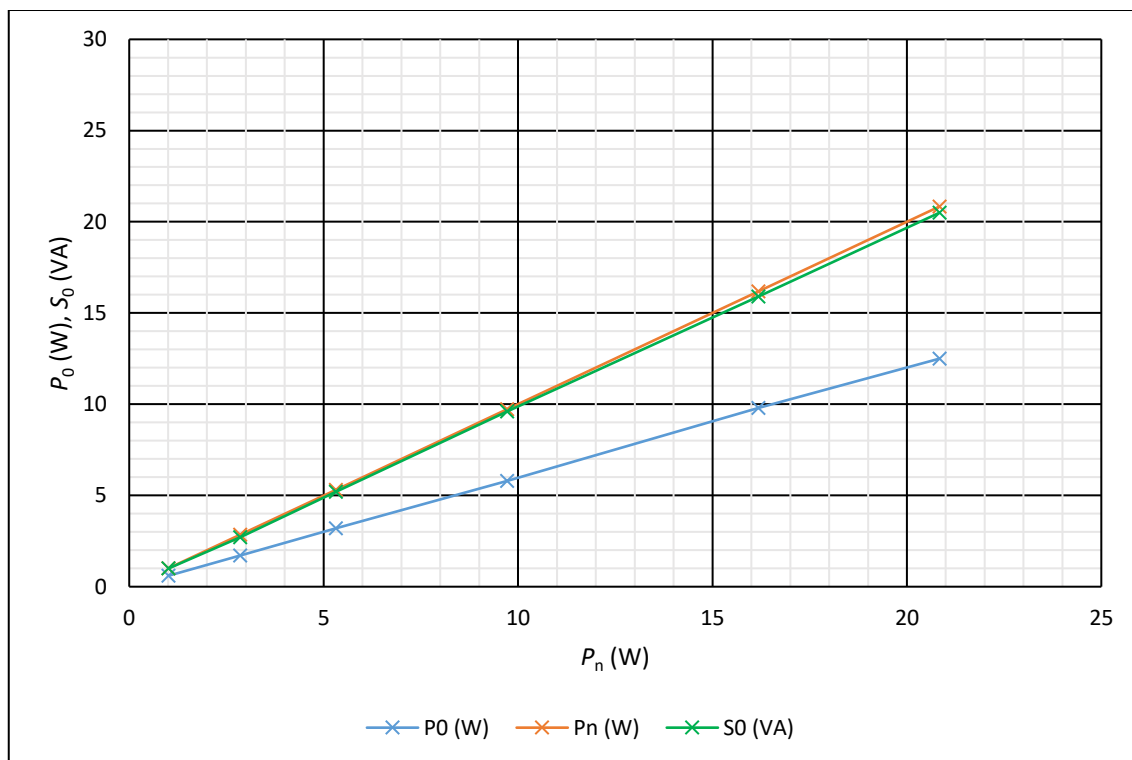


Obr. 60: Závislost naměřeného proudu zkoušeným měřidlem na proudu měřeným přesným měřidlem

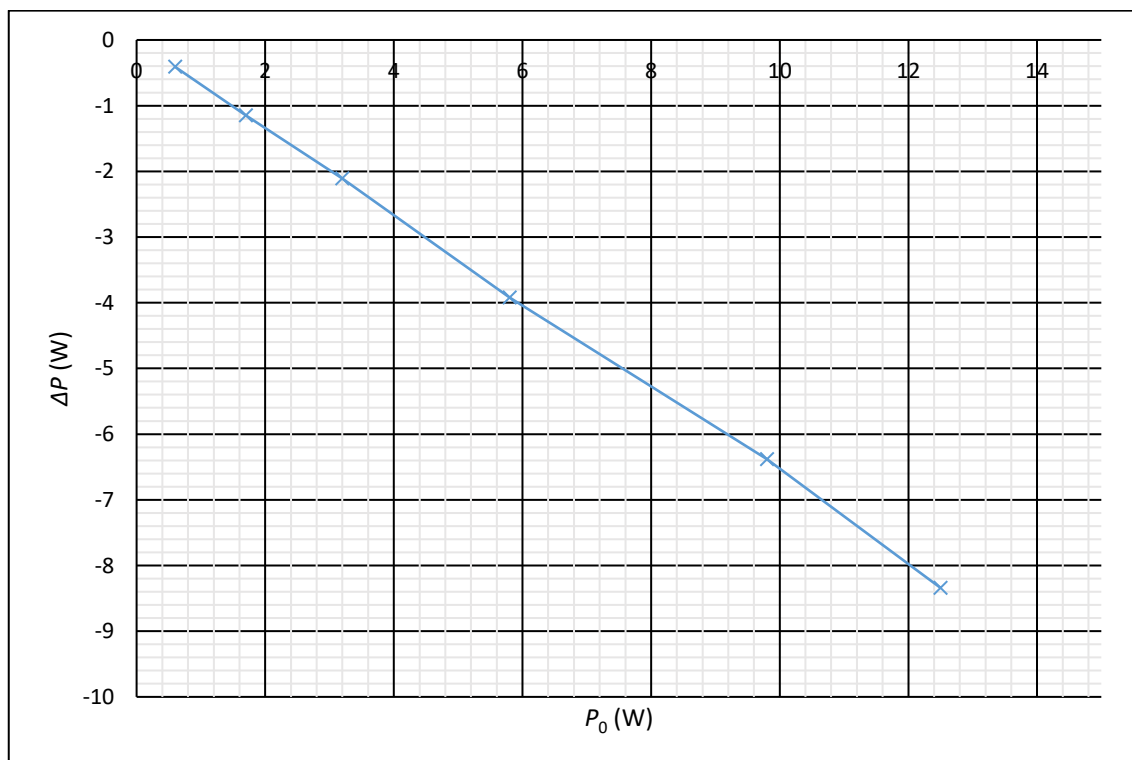


Obr. 61: Závislost absolutní chyby měření na měřeném proudu

#### 4. OVĚŘENÍ FUNKCÍ NAVRŽENÉHO A REALIZOVANÉHO ELEKTRONICKÉHO WATTMETRU



Obr. 62: Závislost naměřeného činného a zdánlivého výkonu zkoušeným měřidlem na činném výkonu měřeným přesným měřidlem



Obr. 63: Závislost absolutní chyby měření činného výkonu na měřeném činném výkonu

## 5 Závěr a diskuse výsledků

V první části práce byly porovnány a popsány různé integrované obvody pro měření výkonu. Byly zde stručně představeny jejich vlastnosti a funkce.

V druhé kapitole byly představeny všechny komponenty použité k návrhu vlastního elektronického wattmetru a k následné realizaci a sestavení měřicího přístroje.

Třetí část práce se již věnuje samotné práci zejména s obvodem *MCP39F511A* a také deskou *PWR Meter click*. Je zde detailně popsána kalibrace měření, vyčítání naměřených hodnot z registrů obvodu a další. Dále je tu také popsána práce se všemi ostatními komponentami, jejich případné nastavení a zapojení. V kapitole je dále popsán návrh schématu zapojení a návrh desky plošných spojů. Je zde také navržena konstrukce přístroje. Součástí je i popis problémů při ožívování desky.

Poslední, tj. čtvrtá kapitola, se věnuje měření a ověření parametrů sestaveného měřidla. Z naměřených průběhů je vidět jisté zlepšování, ovšem je zde potřeba provést další kalibrace zejména ve střídavé oblasti. Z důvodu zdržení při hledání problémů s ožívováním desky nezbyl už další čas na přesnou kalibraci všech potřebných parametrů. Určitě je ještě nutné provést kalibraci pro střídavé měření a doladění offsetů příslušných parametrů. Toto jsou však „jen“ softwarové úpravy, které nepotřebují žádný zásah do hardwaru. Naopak co by vyžadovalo zásah do hardwaru je vylepšení napájení pro proudovou sondu, která byla pro účely měření napájena z externího zdroje. Řešením by mohlo být použití DC/DC měniče sice ze stejné řady *IES01*, ale s vyšším výstupním napětím např. 10 V místo 5 V a pro účely napájení proudové sondy napětí snížit stabilizátorem na přesné napětí 5 V.

Po odstranění těchto potíží výše popsanými řešeními, bude zařízení schopné velmi přesného měření jak stejnosměrného, tak střídavého výkonu. Bylo by také vhodné doplnit měření v celém rozsahu přístroje. Dále by bylo dobré provést měření na zátěži, která není čistě odporová.

Měřič v tuto chvíli v pořádku vypisuje na displej přístroje naměřené hodnoty napětí  $U$ , proudu  $I$ , činného výkonu  $P$ , jalového výkonu  $Q$ , zdánlivého výkonu  $S$ , frekvenci  $f$  a také zda měření probíhá ve stejnosměrném nebo střídavém režimu. Všechny tyto hodnoty jsou současně při připojení měřiče USB kabelem k PC odesílány po sériovém portu do PC, kde mohou být zobrazeny na terminálu, případně se mohou dále zpracovat ve vhodné aplikaci.

## Literatura

- [1] Energy Metering ICs. *Analog Devices* [online]. [vid. 2023-03-21]. Dostupné z: <https://www.analog.com/en/product-category/energy-metering-ics.html>
- [2] Smart Metering ICs Support Accurate Power Monitoring. *Electronic Design* [online]. [vid. 2023-03-21]. Dostupné z: <https://www.electronicdesign.com/technologies/power-electronics-systems/article/21191899/smart-metering-ics-support-accurate-power-monitoring>
- [3] *Třídy přesnosti elektroměrů. Požadavky na elektroměr pro přesnost. Zařízení na měření elektřiny* [online]. 2019 [vid. 2023-03-22]. Dostupné z: <https://cs.puntomariner.com/accuracy-classes-of-electricity-meters/>
- [4] CS5463 Single Phase, Bi-directional Power/Energy IC. *Cirrus Logic* [online]. [vid. 2023-03-19]. Dostupné z: [https://statics.cirrus.com/pubs/proDatasheet/CS5463\\_F4.pdf](https://statics.cirrus.com/pubs/proDatasheet/CS5463_F4.pdf)
- [5] CS5463: Cirrus Logic. *Cirrus Logic* [online]. [vid. 2023-03-19]. Dostupné z: <https://www.cirrus.com/products/cs5463/>
- [6] 78M6613 Evaluation Board User Manual. *Maxim Integrated* [online]. 2011 [vid. 2023-03-20]. Dostupné z: <https://pdfserv.maximintegrated.com/en/an/UG5297.pdf>
- [7] 78M6613 Single-Phase AC Power Measurement IC. *Maxim Integrated* [online]. 2012 [vid. 2023-03-19]. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/78m6613.pdf>
- [8] 78M6613-IM/F Maxim Integrated. *Mouser Electronics* [online]. [vid. 2023-03-20]. Dostupné z: <https://cz.mouser.com/ProductDetail/Maxim-Integrated/78M6613-IM-F?qs=VX0QGrT4dTwXcLQYQNNYjA%3D%3D>
- [9] Power Circuits for Instrumentation. *Analog Devices* [online]. [vid. 2023-03-21]. Dostupné z: <https://www.analog.com/en/applications/markets/instrumentation-and-measurement-pavilion-home/electronic-test-and-measurement/power-circuits-for-instrumentation.html>
- [10] Power Management. *Analog Devices* [online]. [vid. 2023-03-21]. Dostupné z: <https://www.analog.com/en/product-category/power-management.html>

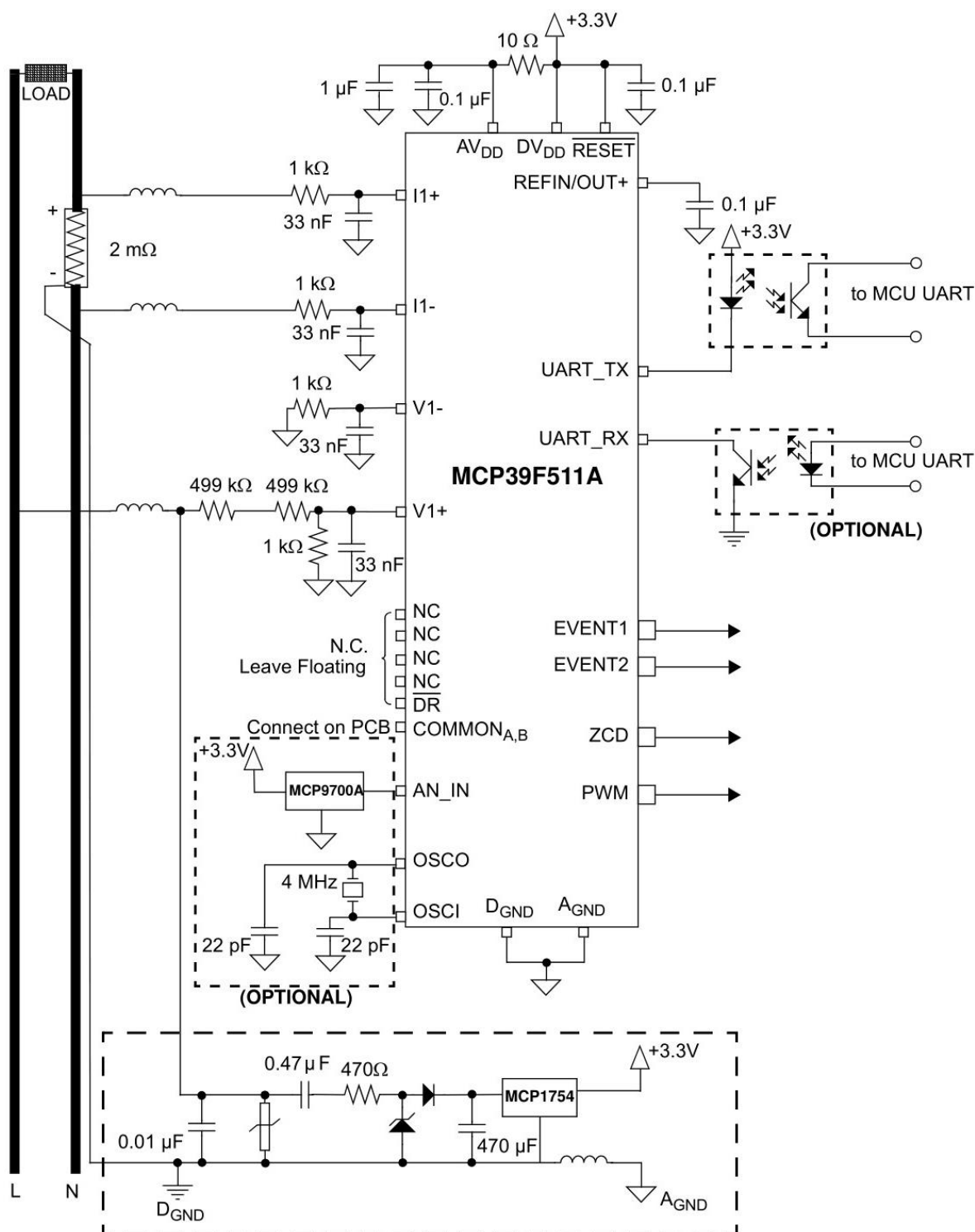
- [11] Digital Isolation in Smart Energy Metering Applications. *Analog Devices* [online]. [vid. 2023-03-21]. Dostupné z: <https://www.analog.com/en/technical-articles/digital-isolation-in-smart-energy-metering-applications.html>
- [12] Electric Meters. *Analog Devices* [online]. [vid. 2023-03-21]. Dostupné z: <https://www.analog.com/en/applications/markets/energy-pavilion-home/metering-and-energy-monitoring/electric-meters.html>
- [13] ADE9153A: Energy Metering IC with Autocalibration. *Analog Devices* [online]. [vid. 2023-03-21]. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/ade9153a.pdf>
- [14] ADE9153A Datasheet and Product Info. *Analog Devices* [online]. [vid. 2023-03-21]. Dostupné z: <https://www.analog.com/en/products/ade9153a.html#product-overview>
- [15] ADE9153A Energy Metering ICs with Autocalibration. *Mouser Electronics* [online]. [vid. 2023-03-22]. Dostupné z: <https://eu.mouser.com/new/analog-devices/adi-ade9153>
- [16] ADE7759 Datasheet and Product Info. *Analog Devices* [online]. [vid. 2023-03-22]. Dostupné z: <https://www.analog.com/en/products/ade7759.html#product-samplebuy>
- [17] ADE7758 Datasheet. *Analog Devices* [online]. [vid. 2023-05-13]. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADE7758.pdf>
- [18] ADE7758 Datasheet and Product Info. *Analog Devices* [online]. [vid. 2023-05-13]. Dostupné z: <https://www.analog.com/en/products/ade7758.html>
- [19] MCP3905. *Microchip Technology* [online]. [vid. 2023-05-13]. Dostupné z: <https://www.microchip.com/en-us/product/MCP3905>
- [20] MCP3905. *Microchip* [online]. [vid. 2023-05-13]. Dostupné z: <https://ww1.microchip.com/downloads/en/devicedoc/21948b.pdf>
- [21] SHEPARD, Paul. Single IC Provides Real-Time Measurements of Both AC and DC Power - New Industry Products. *EE Power* [online]. 2018 [vid. 2023-05-13]. Dostupné z: <https://eepower.com/new-industry-products/single-ic-provides-real-time-measurements-of-both-ac-and-dc-power/#>

- [22] MCP39F511A. *Microchip* [online]. [vid. 2023-05-13]. Dostupné z: <https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/MCP39F511A-Data-Sheet-20006044A.pdf>
- [23] Time-saving embedded tools. *MIKROE* [online]. [vid. 2023-05-11]. Dostupné z: <https://www.mikroe.com/>
- [24] PWR Meter click. *Mikroelektronika* [online]. [vid. 2023-05-13]. Dostupné z: <https://www.mikroe.com/pwr-meter-click>
- [25] PWR Meter click schematic v100. *MikroElektronika* [online]. [vid. 2023-05-13]. Dostupné z: <https://download.mikroe.com/documents/add-on-boards/click/pwr-meter/pwr-meter-click-schematic-v100.pdf>
- [26] NUCLEO-F303K8. *Mbed* [online]. [vid. 2023-05-11]. Dostupné z: <https://os.mbed.com/platforms/ST-Nucleo-F303K8/>
- [27] ACHS-7121 Current Sensor Carrier -10A to +10A Pololu 4030. *MakerSupplies Singapore* [online]. [vid. 2023-05-13]. Dostupné z: <https://www.makersupplies.sg/products/achs-7121-current-sensor-carrier-10a-to-10a>
- [28] EA DOGS164-A. *DISPLAY VISIONS GmbH* [online]. [vid. 2023-05-14]. Dostupné z: <https://www.tme.eu/Document/6b80336f65557e5fbbdec8c38f4b332a/EADOGS164X-A.pdf>
- [29] HCPL-270L/070L/273L/073L Low Input Current, High Gain, LVTTTL/LVCMOS Compatible Optocouplers. *Avago Technologies* [online]. [vid. 2023-05-14]. Dostupné z: <https://www.farnell.com/datasheets/1266251.pdf>
- [30] IES01 Series. *XP Power* [online]. [vid. 2023-05-14]. Dostupné z: [https://cz.mouser.com/datasheet/2/942/XP\\_power\\_12\\_7\\_2022\\_SF\\_IES01-3077096.pdf](https://cz.mouser.com/datasheet/2/942/XP_power_12_7_2022_SF_IES01-3077096.pdf)
- [31] Very low-dropout voltage regulator with inhibit function. *STMicroelectronics* [online]. 2017 [vid. 2023-05-14]. Dostupné z: <https://cz.mouser.com/datasheet/2/389/lexx-1849820.pdf>
- [32] ICL7660S, ICL7660A Super Voltage Converters. *Renesas* [online]. 2013 [vid. 2023-05-14]. Dostupné z: [https://cz.mouser.com/datasheet/2/698/REN\\_icl7660s\\_a\\_DST\\_20121024-1997724.pdf](https://cz.mouser.com/datasheet/2/698/REN_icl7660s_a_DST_20121024-1997724.pdf)



- [33] TC7650 Chopper Stabilized Operational Amplifier. *Microchip* [online]. 2012 [vid. 2023-05-15]. Dostupné z: <https://cz.mouser.com/datasheet/2/268/21463C-2449324.pdf>
- [34] Převodník úrovní USB na TTL. (CH340) . *Santy.cz* [online]. [vid. 2023-05-16]. Dostupné z: <https://www.santy.cz/moduly-c22/sa0323-i346/>
- [35] LD39050. *STMicroelectronics* [online]. 2019 [vid. 2023-05-19]. Dostupné z: <https://cz.mouser.com/datasheet/2/389/ld39050-1849494.pdf>
- [36] PLA filamenty. *MaterialPRO3D* [online]. [vid. 2023-05-16]. Dostupné z: <https://www.materialpro3d.cz/materialovy-slovník/pla/>
- [37] BS-50A-B - Zásuvka; banánek 4mm; 50A; 60VDC; černá; s přišroubováním. *TME Czech Republic s.r.o.* [online]. [vid. 2023-05-20]. Dostupné z: <https://www.tme.eu/cz/details/bs-50a-b/zdirky-pro-bananky-4mm/>

## 1 Příloha – doporučené zapojení obvodu MCP39F511A [22]



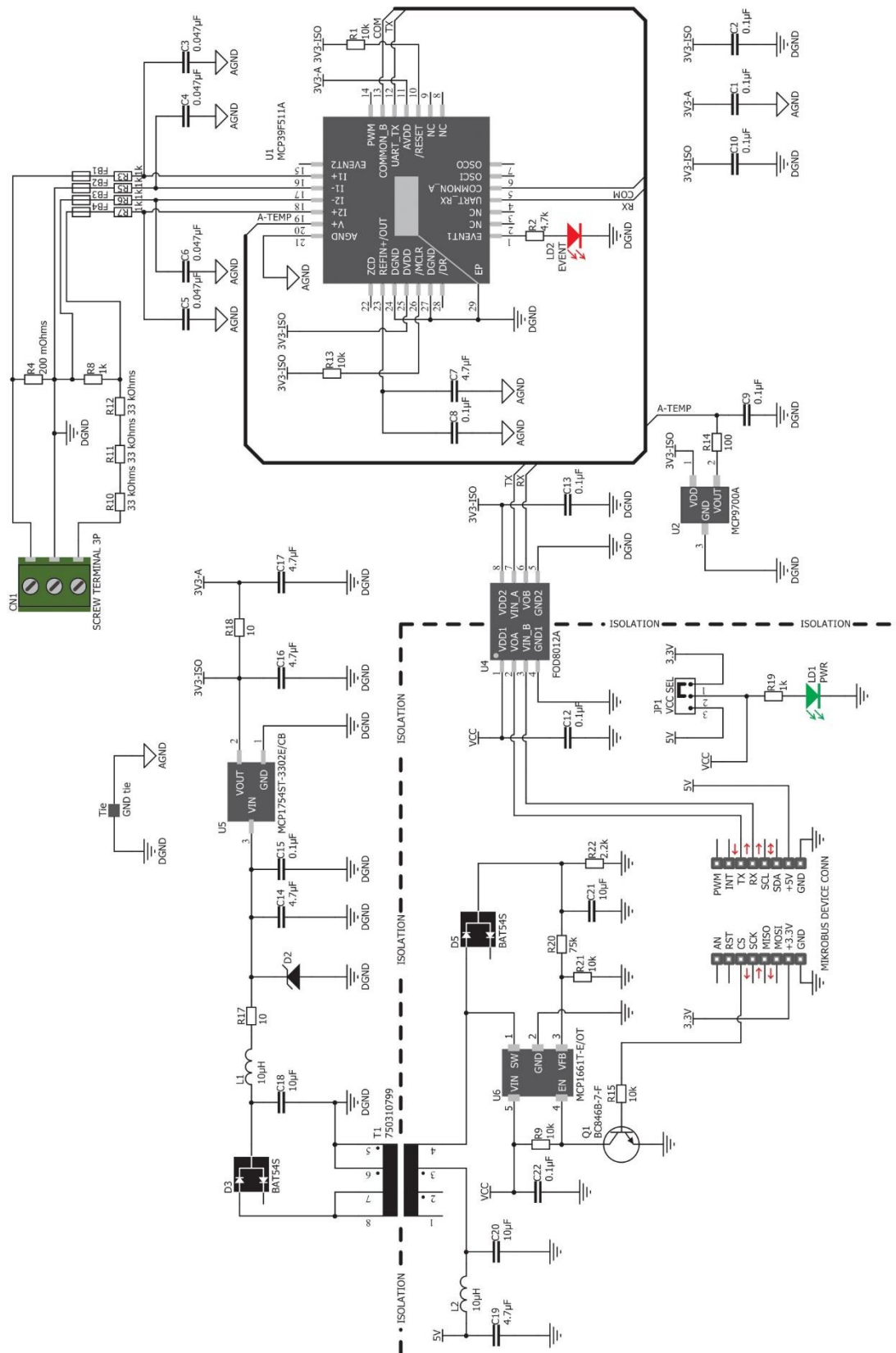
**Note 1:** The MCP39F511A demonstration board uses a switching power supply, however a low-cost capacitive-based supply, as shown here, is sufficient for many applications.

**2:** The external sensing components shown here, a 2 mΩ shunt, two 499 kΩ and 1 kΩ resistors for the 1000:1 voltage divider, are specifically chosen to match the default values for the calibration registers defined in [Section 6.0 “Register Descriptions”](#). By choosing low-tolerance components of these values (for instance 1% tolerance), measurement accuracy in the 2-3% range can be achieved with zero calibration (AC only, offset calibration may be needed in DC mode). See [Section 9.0 “MCP39F511A Calibration”](#) for more information.

## 2 Příloha – schéma zapojení desky PWR Meter click [25]



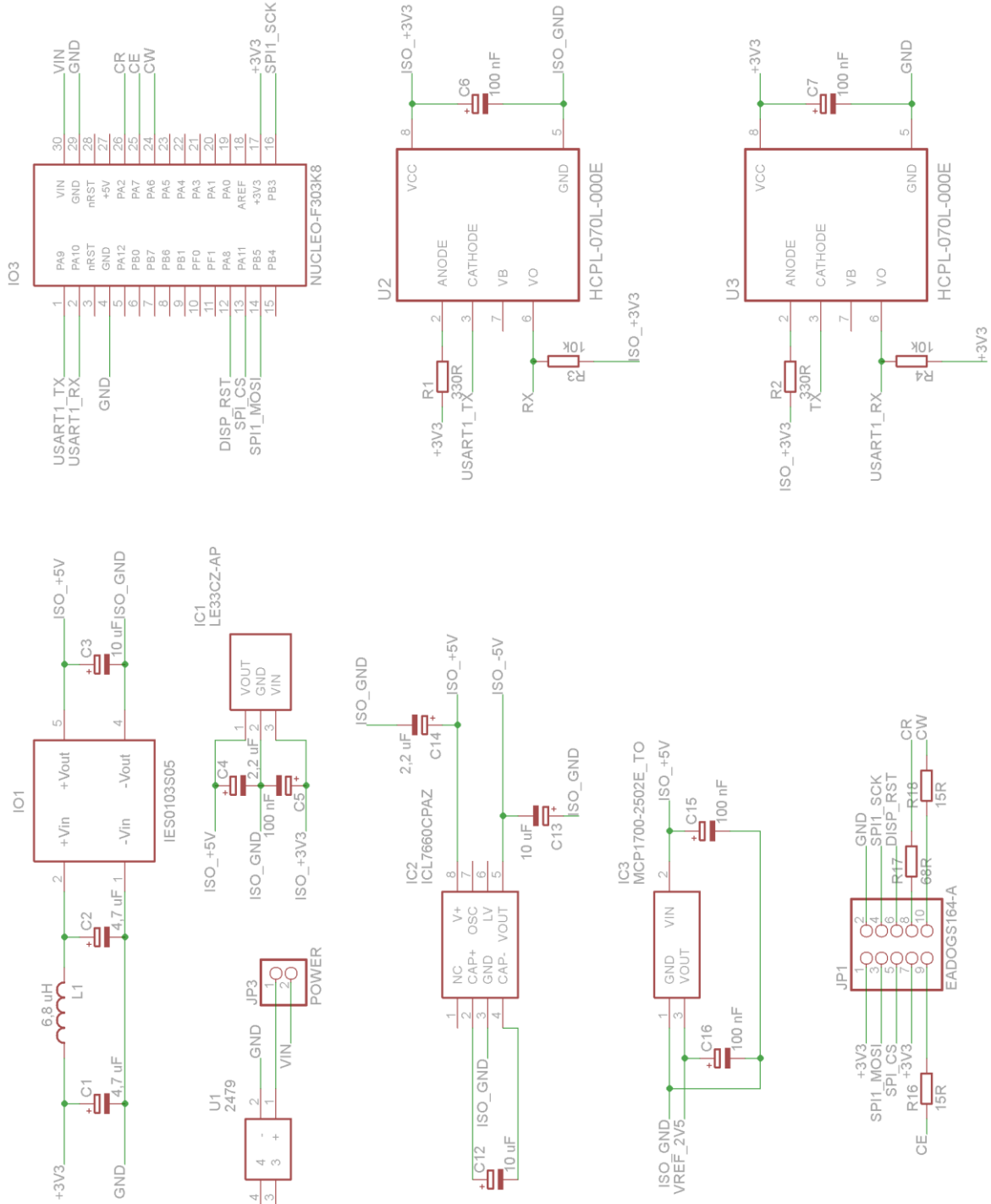
PWR Meter click schematic v100



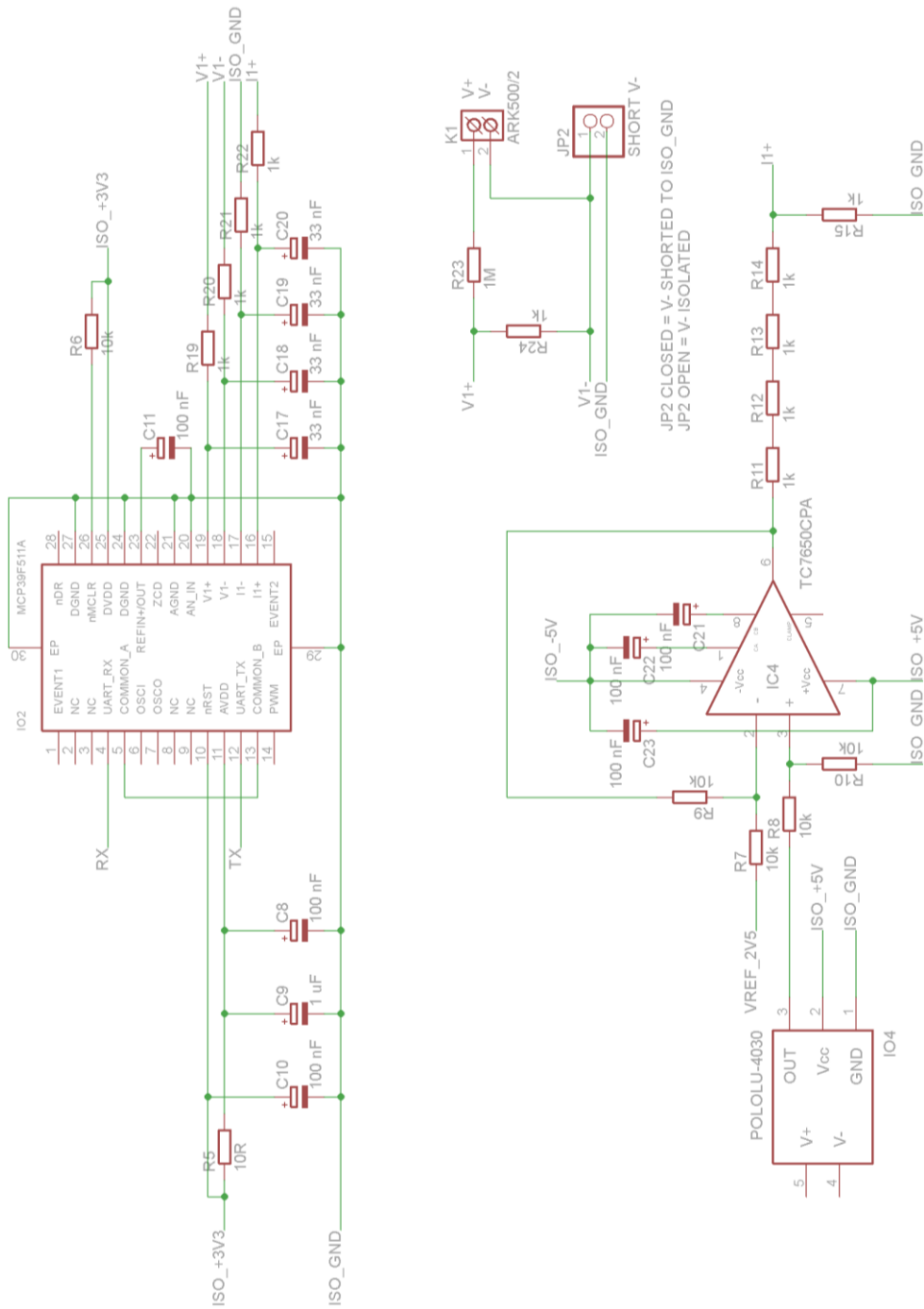
www.mikroe.com

MikroElektronika assumes no responsibility or liability for any errors or inaccuracies that may appear in the present document. Specification and information contained in the present schematic are subject to change at any time without notice. Copyright©2018 MikroElektronika. All rights reserved.

### 3 Příloha – schéma zapojení navrženého měřidla

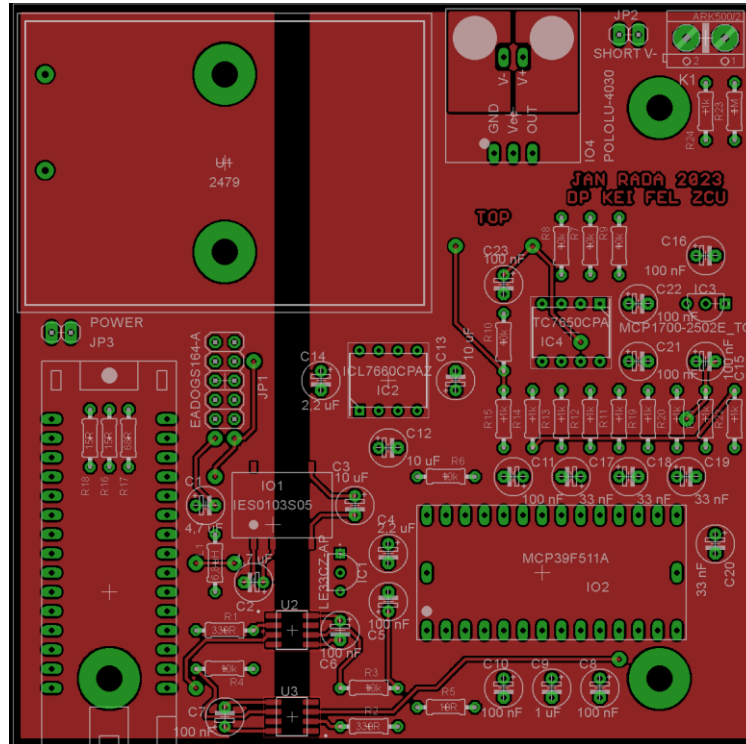


### 3. PŘÍLOHA – SCHÉMA ZAPOJENÍ NAVRŽENÉHO MĚŘIDLA

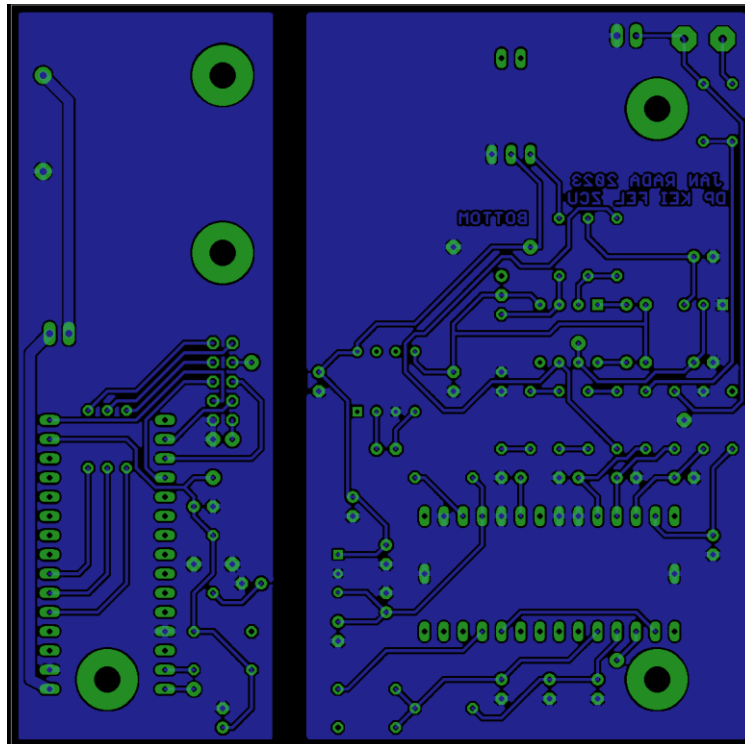


## 4 Příloha – návrh desky plošných spojů

Strana TOP, měřítko 1:1

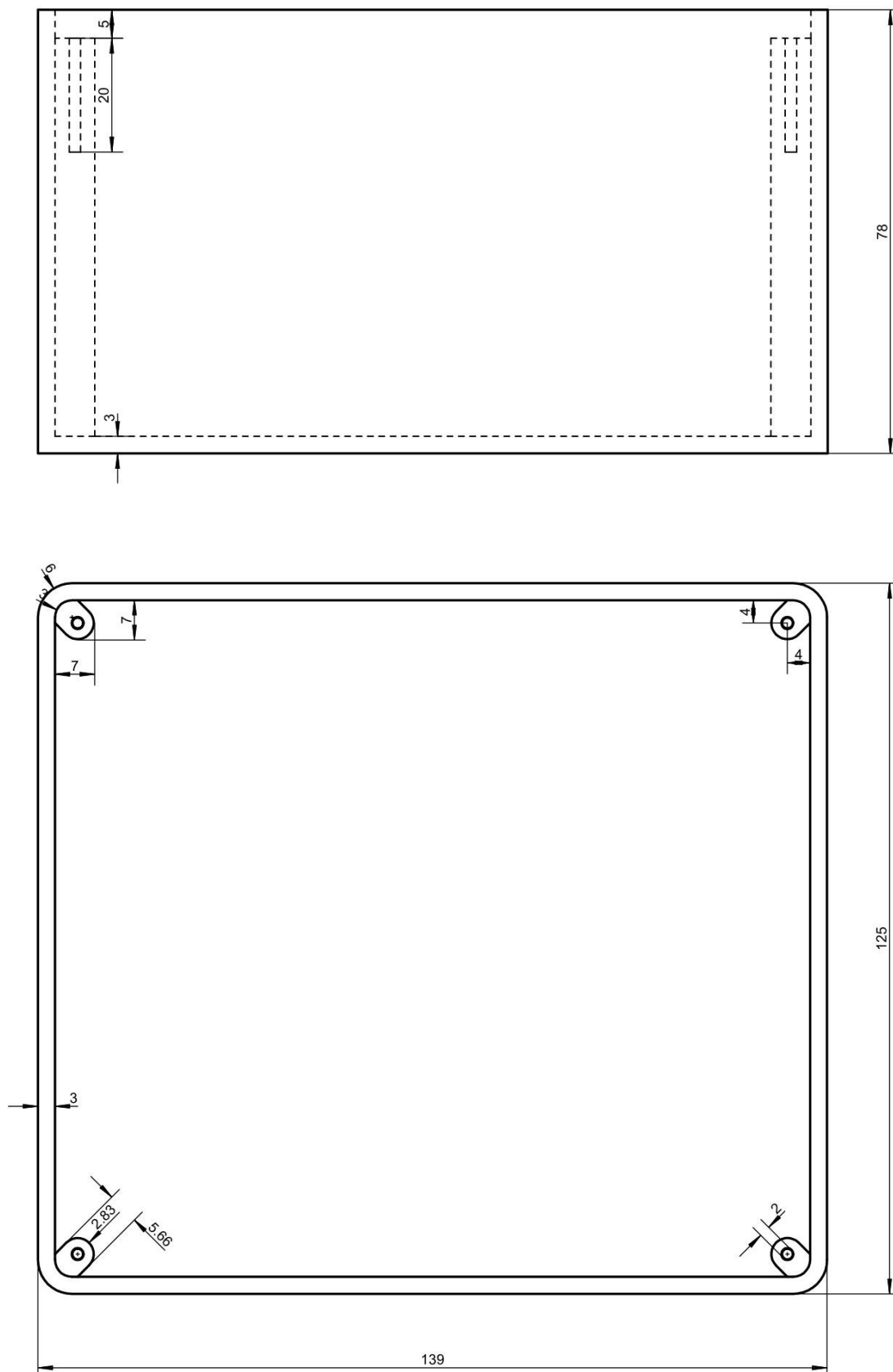


Strana BOTTOM, měřítko 1:1

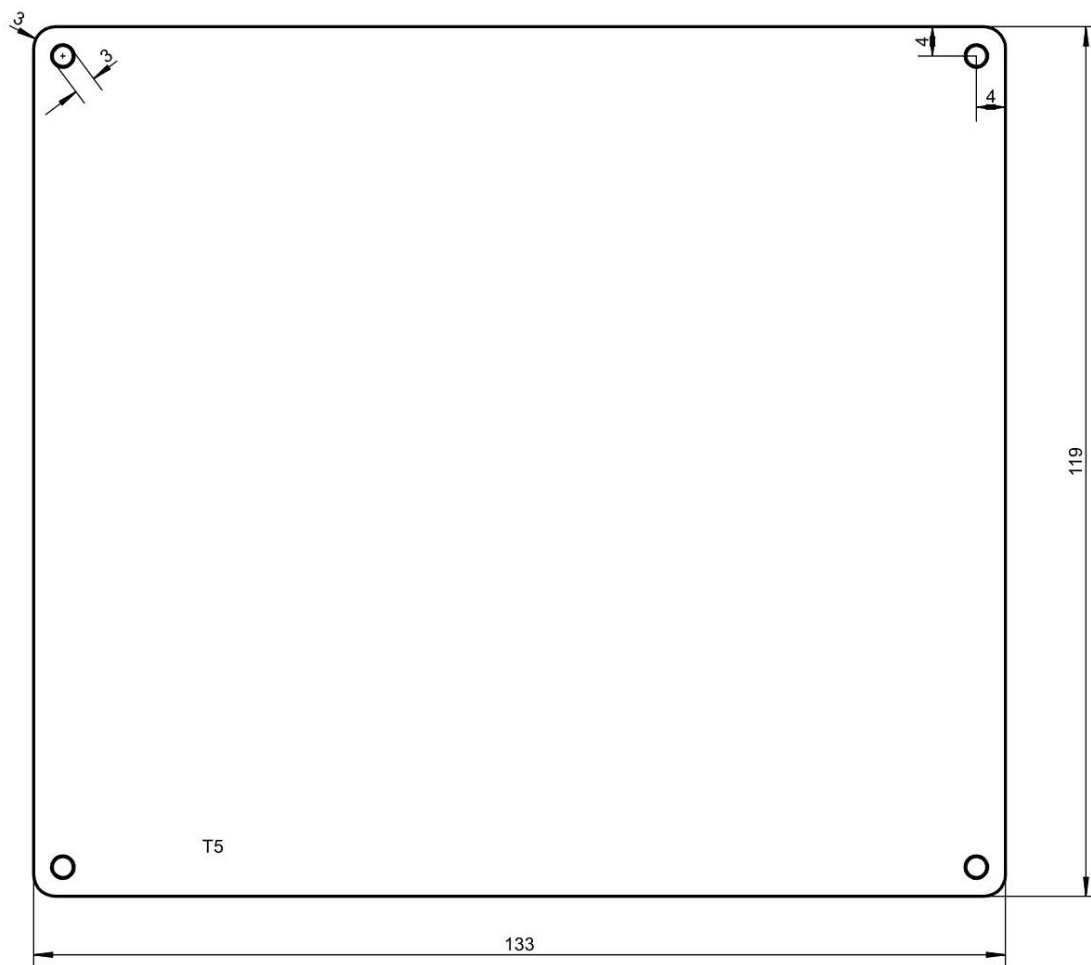


## 5 Příloha – výkres navržené konstrukční krabičky

Výkres navržené krabičky v měřítku 1:1



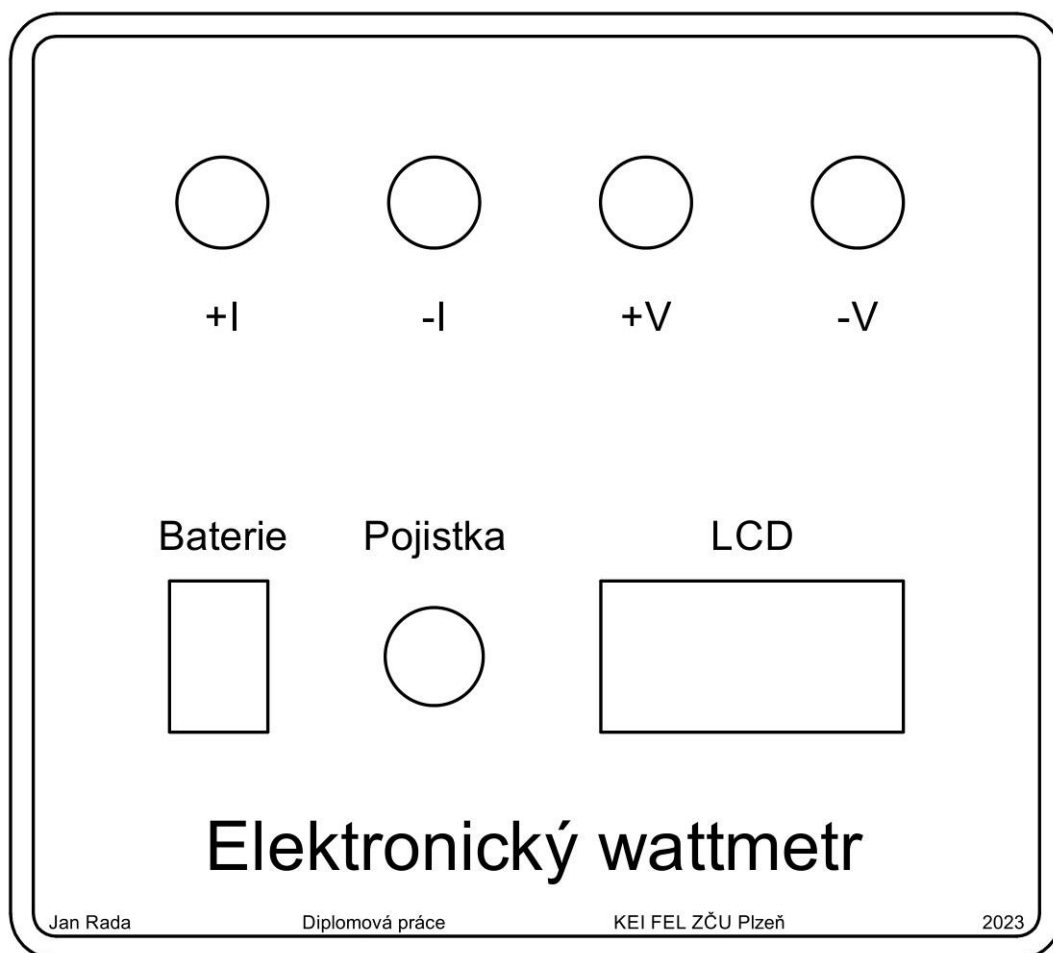
Výkres víčka k navržené krabičce v měřítku 1:1





## 6 Příloha – potisk krabičky

Potisk krabičky v měřítku 1:1



## 7 Příloha – programový kód v jazyku C

main.c

```

1 /*****
2 *
3 * Zapadoceska univerzita v Plzni
4 *
5 * Fakulta elektrotechnicka
6 *
7 * Katedra elektroniky a informačních technologií
8 *
9 * Diplomova prace
10 *
11 * Elektronicky wattmetr
12 *
13 * Jan Rada
14 *
15 * 2023
16 *
17 *****/
18
19 #include "my_nucleo.h"
20 #include "EADOGS_164.h"
21 #include "PWR_meter.h"
22
23 #define PWRMETER_DCMODE_MASK 0x8000
24 #define PWRMETER_DCCURR_SIGN_MASK 0x4000
25 #define PWRMETER_DCVOLT_SIGN_MASK 0x2000
26 #define PWRMETER_PR_SIGN_MASK 0x0020
27 #define PWRMETER_PA_SIGN_MASK 0x0010
28
29 #define CALIB_OFF 1 // pokud probiha kalibrace = 0, pokud probiha
    bezny provoz = 1
30
31 volatile uint32_t _ticks = 0; // pocitadlo stovek mikrosekund
32
33 void SysTick_Handler(void) // obsluha preruseni od SysTicku
34 {
35     _ticks++; // priznak dopocitani není nutne nulovat
36 }
37
38 int main(void)
39 {
40     // PLLInit(); // takt procesoru 64 MHz
41
42     uint32_t k;
43     for(k = 0; k < 10000; k++)
44         ;
45
46     uint32_t tm = 0;
47     uint32_t tm2 = 0;
48     // bool ledOn = false;
49     char znak = 'A';
50     char x = '*';
51     uint8_t recChckSum = 0; // prijaty check sum
52     uint8_t chckSum = 0; // vypocet kontrolního součtu (check sum)
53     uint8_t messlen = 1; // delka zpravy
54     uint8_t index = 0; // indexování Bytu v prijate zprave
55     uint8_t recbuf[100]; // prijimaci buffer
56     char logbuff[100]; // buffer pro vypis na displej a na obrazovku

```

## 7. PŘÍLOHA – PROGRAMOVÝ KÓD V JAZYKU C

```
main.c

57 bool      messComplete = false; // priznka, ze byl prijaty cely ramec
58 uint8_t   messnum = 0; // poradí zpravy
59
60 uint16_t  voltage_rms; // surova data napeti
61 uint32_t  current_rms; // surova data proudu
62 uint32_t  active_power; // surova data P
63 uint32_t  reactive_power; // surova data Q
64 uint32_t  apparent_power; // surova data S
65 uint16_t  frequency; // surova data f
66 uint16_t  status_byte; // obsahuje udaje o znamenkach namerenych hodnot
67
68 float     meas_data[ 6 ]; // prepocitana namerena data pro tisk
69
70 /* ----- zpravy pro PWR meter ----- */
71 uint8_t   PWR_R_U[] = {0xa5, 0x08, 0x41, 0x00, 0x06, 0x4e, 0x02, 0x44}; // precti napeti U
72 uint8_t   PWR_R_I[] = {0xa5, 0x08, 0x41, 0x00, 0x0e, 0x4e, 0x04, 0x4e}; // precti proud I
73 uint8_t   PWR_R_P[] = {0xa5, 0x08, 0x41, 0x00, 0x12, 0x4e, 0x04, 0x52}; // precti cinny
    vykon P
74 uint8_t   PWR_R_Q[] = {0xa5, 0x08, 0x41, 0x00, 0x16, 0x4e, 0x04, 0x56}; // precti jalovy
    vykon Q
75 uint8_t   PWR_R_S[] = {0xa5, 0x08, 0x41, 0x00, 0x1A, 0x4e, 0x04, 0x5A}; // precti zdanlivy
    vykon S
76 uint8_t   PWR_R_f[] = {0xa5, 0x08, 0x41, 0x00, 0x08, 0x4e, 0x02, 0x46}; // precti
    frekvenci f
77 uint8_t   PWR_R_StatusR[] = {0xa5, 0x08, 0x41, 0x00, 0x02, 0x4e, 0x02, 0x40}; // precti
    Status register
78
79 #if CALIB_OFF == 0
80 uint8_t   PWR_R_Range[] = {0xA5, 0x08, 0x41, 0x00, 0x9C, 0x4E, 0x04, 0xDC}; // precti
    Range register
81
82 uint8_t   PWR_W_RangeR[] = {0xa5, 0x0c, 0x41, 0x00, 0x9c, 0x4d, 0x04, 0x12, 0x02, 0x17,
    0x1b, 0x25}; // zapis do Range reg
83 uint8_t   PWR_R_CalibU[] = {0xA5, 0x08, 0x41, 0x00, 0x6E, 0x4E, 0x02, 0xAC}; // precti U
    calib reg
84 // uint8_t   PWR_W_CalibU[] = {0xa5, 0x0a, 0x41, 0x00, 0x6E, 0x4D, 0x02, 0x5e, 0xb2,
    0xbd}; // zapis do U calib reg 0xb25e
85 uint8_t   PWR_R_CalibI[] = {0xA5, 0x08, 0x41, 0x00, 0x6C, 0x4E, 0x02, 0xAA}; // precti I
    calib reg
86 // uint8_t   PWR_W_CalibI[] = {0xa5, 0x0a, 0x41, 0x00, 0x6C, 0x4D, 0x02, 0x31, 0x81,
    0x5d}; // zapis do I calib reg 0x8131
87 uint8_t   PWR_R_CalibP[] = {0xA5, 0x08, 0x41, 0x00, 0x70, 0x4E, 0x02, 0xAE}; // precti P
    calib reg
88 // uint8_t   PWR_W_CalibP[] = {0xa5, 0x0a, 0x41, 0x00, 0x70, 0x4D, 0x02, 0xD3, 0xBE,
    0x40}; // zapis do P calib reg 0xBED3
89
90 uint8_t   PWR_W_CalibU[] = {0xa5, 0x0a, 0x41, 0x00, 0x6E, 0x4D, 0x02, 0xdf, 0xae, 0x3a}; //
    zapis do U calib reg 0xaedf
91 uint8_t   PWR_W_CalibI[] = {0xa5, 0x0a, 0x41, 0x00, 0x6C, 0x4D, 0x02, 0x09, 0x85, 0x39}; //
    zapis do I calib reg 0x8509
92 uint8_t   PWR_W_CalibP[] = {0xa5, 0x0a, 0x41, 0x00, 0x70, 0x4D, 0x02, 0x16, 0xdb, 0xa0}; //
    zapis do P calib reg 0xdb16
93
94
95 /*
96 uint8_t   PWR_R_OffsetI[] = {0xA5, 0x08, 0x41, 0x00, 0x72, 0x4E, 0x02, 0xB0}; // precti I DC
    offset reg
```

## 7. PŘÍLOHA – PROGRAMOVÝ KÓD V JAZYKU C

```
main.c

97 // uint8_t PWR_W_OffsetI[] = {0xa5, 0x0a, 0x41, 0x00, 0x72, 0x4D, 0x02, 0x37, 0xbe,
   // 0xa6}; // zapis do I DC offset reg 0xbe37
98 uint8_t PWR_W_OffsetI[] = {0xa5, 0x0a, 0x41, 0x00, 0x72, 0x4D, 0x02, 0x00, 0x00, 0xb1}; //
   // zapis do I DC offset reg 0xbe37
99
100 */
101
102
103 uint8_t PWR_W_TargetI[] = {0xA5, 0x0C, 0x41, 0x00, 0xA0, 0x4D, 0x04, 0x40, 0x1A, 0xC8,
   // 0x05, 0x0A}; // musim 0,970 A, 0x5C81A40, cil kalibrace I = 1.014A ... 0x060B3DC0 (dek.
   // 101 400 000), potom delim 100 000 000
104 uint8_t PWR_W_TargetU[] = {0xA5, 0x0A, 0x41, 0x00, 0xA4, 0x4D, 0x02, 0x03, 0xE8,
   // 0xCE}; // musim 10 V ,stejne, cil kalibrace U = 10.0V ... 0x03E8 (dek. 1000), potom delim
   // 100
105 uint8_t PWR_W_TargetP[] = {0xA5, 0x0C, 0x41, 0x00, 0xA6, 0x4D, 0x04, 0xB7, 0x03, 0x00,
   // 0x00, 0xA3}; // musim 9,51 0x3B7, cil kalibrace P = 10.17W ..., 0x3F9 (dek. 1017)potom
   // delim 100
106 // uint8_t PWR_W_TargetQ[] = {0xA5, 0x0C, 0x41, 0x00, 0xAA, 0x4D, 0XX, 0xXX, 0xXX, 0xXX,
   // 0xXX, 0xXX}; // cil kalibrace Q = 0var
107 uint8_t PWR_W_AutoCalDC[] = {0xA5, 0x04, 0x5A, 0x03}; // Auto-calibrate gain command
108
109 uint8_t PWR_R_TargetI[] = {0xA5, 0x08, 0x41, 0x00, 0xA0, 0x4E, 0x04, 0xE0}; // precti cil
   // I cal
110 uint8_t PWR_R_TargetU[] = {0xA5, 0x08, 0x41, 0x00, 0xA4, 0x4E, 0x02, 0xE2}; // precti cil
   // U cal
111 uint8_t PWR_R_TargetP[] = {0xA5, 0x08, 0x41, 0x00, 0xA6, 0x4E, 0x04, 0xE6}; // precti cil
   // P cal
112
113
114 uint8_t PWR_SaveToFlash[] = {0xa5, 0x04, 0x53, 0xfc}; // save to Flash
115
116 uint8_t PWR_W_config[] = // zapis VOLT_GAIN_1 | CURR_GAIN_1 | BR_9600 do
   // PWR_SYS_CONFIG_REG
117 {
118     0xa5, // Header Byte
119     0x0c, // pocet Byte ve zprave
120     0x41, // set address pointer
121     0x00, // address high
122     0x94, // address low
123     0x4d, // write N Bytes
124     0x04, // budu posilat 4 Byty
125     0x00, // nic
126     0x80, // BR = 9600 Bd
127     0x00, // VOLT_GAIN_1 | CURR_GAIN_1
128     0x00, // VOLT_GAIN_1 | CURR_GAIN_1
129     0x57 // Check sum
130 };
131
132 uint8_t PWR_R_Page1[] = {0xa5, 0x05, 0x42, 0x01, 0xed}; // precti Page 1
133
134 uint8_t PWR_R_config[] = // precti cfg register
135 {
136     0xa5, // Header Byte
137     0x08, // pocet Byte ve zprave
138     0x41, // set address pointer
139     0x00, // address high
140     0x94, // address low
```

```
main.c

141         0x4e, // read N Bytes
142         0x04, // budu cist 4 Byty
143         0xd4 // Check sum
144     };
145 #endif
146 /* ----- konec zprav pro PWR meter ----- */
147
148 SystemCoreClockUpdate(); // aktualizuje globalni promennou SystemCoreClock
149
150 SysTick_Config(SystemCoreClock / 10000); // nastav 100 us (64M / 8 / 8000)
151
152 // Nucleo_SetPinGPIO(LED_BOARD, ioPortOutputPP); // LEDka nejde pouzit, protoze je na
    stejnem pinu jako SPI1_SCK
153
154 UsartPCInit(115200);
155 UsartPWRmInit(9600);
156
157 printf("\r\n\r\nStart APP - build at " __DATE__ " " __TIME__ "\r\n\r\n");
158
159 printf("    Jan Rada\r\n");
160 printf("Diplomova prace\r\n");
161 printf("    2023\r\n");
162 printf(" KEI FEL ZCU\r\n\r\n");
163
164 SPI1Init();
165
166 LCD_init();
167
168 GPIOWrite(LCD_GREEN, false); // false = sviti, true = nesviti
169
170 LCD_string("    Jan Rada    ");
171 LCD_string("Diplomova prace ");
172 LCD_string("    2023    ");
173 LCD_string(" KEI FEL ZCU ");
174
175 while (1)
176 {
177
178 #if CALIB_OFF == 0
179     if(IsUsartPCRecv())
180     {
181         znak = UsartPCRecv();
182
183         switch(znak)
184         {
185             case '1':
186                 UsartPWRmString(PWR_R_U);
187                 LCD_char('1');
188                 break;
189             case '2':
190                 UsartPWRmString(PWR_R_I);
191                 LCD_char('2');
192                 break;
193             case '3':
194                 UsartPWRmString(PWR_R_P);
195                 LCD_char('3');
196                 break;
```

```
main.c

197     case '4':
198         UsartPWRmString(PWR_R_config);
199         LCD_char('4');
200         break;
201     case '5':
202         UsartPWRmString(PWR_W_config);
203         LCD_char('5');
204         break;
205     case '6':
206         UsartPWRmString(PWR_SaveToFlash);
207         LCD_char('6');
208         break;
209     case '7':
210         UsartPWRmString(PWR_R_StatusR);
211         LCD_char('7');
212         break;
213     case '8':
214         UsartPWRmString(PWR_W_RangeR);
215         LCD_char('8');
216         break;
217     case '9':
218         UsartPWRmString(PWR_R_CalibI);
219         LCD_char('9');
220         break;
221     case '0':
222         UsartPWRmString(PWR_W_CalibI);
223         LCD_char('0');
224         break;
225     case '*':
226         UsartPWRmString(PWR_R_Range);
227         LCD_char('*');
228         break;
229     case '/':
230         UsartPWRmString(PWR_R_CalibU);
231         LCD_char('/');
232         break;
233     case '+':
234         UsartPWRmString(PWR_W_CalibU);
235         LCD_char('+');
236         break;
237     case '-':
238         UsartPWRmString(PWR_R_CalibP);
239         LCD_char('-');
240         break;
241     case '.':
242         UsartPWRmString(PWR_W_CalibP);
243         LCD_char('.');
244         break;
245     case 'a':
246         UsartPWRmString(PWR_W_TargetI);
247         LCD_char('a');
248         break;
249     case 'b':
250         UsartPWRmString(PWR_W_TargetU);
251         LCD_char('b');
252         break;
253     case 'c':
```

```
main.c

254         UsartPWRmString(PWR_W_TargetP);
255         LCD_char('c');
256         break;
257     case 'A':
258         UsartPWRmString(PWR_R_TargetI);
259         LCD_char('A');
260         break;
261     case 'B':
262         UsartPWRmString(PWR_R_TargetU);
263         LCD_char('B');
264         break;
265     case 'C':
266         UsartPWRmString(PWR_R_TargetP);
267         LCD_char('C');
268         break;
269     case 'd':
270         UsartPWRmString(PWR_W_AutoCalDC);
271         LCD_char('d');
272         break;
273
274     }
275
276 }
277 #endif
278
279 if(IsUsartPWRmRecv()) // prislo neco z PWR meteru ?
280 {
281     recbuf[index] = UsartPWRmRecv(); // ukladej prijate Byty do bufferu
282
283     if(index == 0) // if((index == 0) && (recbuf[index] != 0xA5))
284     {
285         // TODO kontrola prvnioho Bytu
286 /*
287         switch(recbuf[index])
288         {
289         case 0x06:
290             // ACK
291             UsartPCString("ACK ");
292             break;
293         case 0xA5:
294             // Header Byte
295             UsartPCString("Data ");
296             break;
297         case 0x51:
298             // CSFAIL
299             UsartPCString("CSFAIL ");
300             break;
301         case 0x15:
302             // NAK
303             UsartPCString("NAK ");
304             break;
305         default:
306             // jiny error
307             UsartPCString("err ");
308             break;
309         }
310 */
```

```

main.c

311     }
312
313     if(index == 1)                // druhy Byte znamena delku zpravy
314         messlen = recbuf[index] - 1;    // delka zpravy minus 1
315
316     if(index == messlen)          // tzn. zprava je u konci a tento Byte je checksum
317     {
318         recChckSum = recbuf[index];
319
320         if(chckSum == recChckSum)    // kontrola check sumu
321         {
322             // TODO kontrola check sumu
323         }
324
325         recbuf[++index] = 0;          // pro ukonzeni retezce kvuli vypisu na disp -
vylepsit
326         index = 0;                  // vynuluj index, aby pristi zprava zacinala opet
na zacatku pole
327         chckSum = 0;                // vynuluj checksum, aby se mohl pocitat odznovu v
dalsim ramci
328         messComplete = true;        // nahod priznak ze byl prijaty cely ramec
329         // LCD_string((char*) recbuf);
330     }
331     else
332     {
333         chckSum += recbuf[index];    // pricti Byte ke kontrolnimu souctu
334
335         index++;
336     }
337
338 }
339
340 #if CALIB_OFF == 1
341
342     if((messComplete || (messnum == 0)) && (_ticks >= tm)) // pokud uz je cas a ( prisel cely
ramec nebo musim pozadat o prvni
343     {
344         messComplete = false;
345
346         switch(messnum)
347         {
348             case 0:
349                 tm2 = _ticks + 10000;    // dalsi mereni za 1 sec.
350                 UsartPWRmString(PWR_R_U); // zeptej se na napeti
351                 messnum++;
352                 break;
353             case 1:
354                 voltage_rms = ((recbuf[3] << 8) | recbuf[2]); // odbaveni predchozi prijate
zpravy o napeti (u16)
355                 UsartPWRmString(PWR_R_I); // zeptej se na proud
356                 messnum++;
357                 break;
358             case 2:
359                 current_rms = ((recbuf[5] << 24) | (recbuf[4] << 16) | (recbuf[3] << 8) | (recbuf
[2])); // odbaveni predchozi prijate zpravy o proudu (u32)
360                 UsartPWRmString(PWR_R_P); // zeptej se na P
361                 messnum++;

```



```

main.c

362         break;
363     case 3:
364         active_power = ((recbuf[5] << 24) | (recbuf[4] << 16) | (recbuf[3] << 8) | (recbuf
[2])); // odbavení předchozí přijaté zpravy o činném výkonu (u32)
365         UsartPWRmString(PWR_R_Q); // zeptej se na Q
366         messnum++;
367         break;
368     case 4:
369         reactive_power = ((recbuf[5] << 24) | (recbuf[4] << 16) | (recbuf[3] << 8) |
(recbuf[2])); // odbavení předchozí přijaté zpravy a o jalovém výkonu (u32)
370         UsartPWRmString(PWR_R_S); // zeptej se na S
371         messnum++;
372         break;
373     case 5:
374         apparent_power = ((recbuf[5] << 24) | (recbuf[4] << 16) | (recbuf[3] << 8) |
(recbuf[2])); // odbavení předchozí přijaté zpravy o zdanlivém výkonu (u32)
375         UsartPWRmString(PWR_R_f); // zeptej se na f
376         messnum++;
377         break;
378     case 6:
379         frequency = ((recbuf[3] << 8) | recbuf[2]); // odbavení předchozí přijaté zpravy o
frekvenci (u16)
380         UsartPWRmString(PWR_R_StatusR); // zeptej se na polaritu měřených hodnot (uloženo
ve Status registeru)
381         messnum++;
382         break;
383     case 7:
384         status_byte = ((recbuf[3] << 8) | recbuf[2]); // odbavení předchozí přijaté zpravy
o Status registeru (u16)
385         meas_data[ 0 ] = ( float ) voltage_rms / 10;
386         meas_data[ 1 ] = ( float ) current_rms / 100000000;
387         meas_data[ 2 ] = ( float ) active_power / 100;
388         meas_data[ 3 ] = ( float ) reactive_power / 100000;
389         meas_data[ 4 ] = ( float ) apparent_power / 1000000;
390         meas_data[ 5 ] = ( float ) frequency / 1000;
391
392         messnum = 0;
393         tm = tm2; // prepis time-stamp
394
395         LCD_clear_all();
396
397         sprintf(logbuff, "U:");
398         LCD_string(logbuff);
399         printf("%s", logbuff);
400         if(((status_byte & PWRMETER_DCMODE_MASK) != 0) && ((status_byte &
PWRMETER_DCVOLT_SIGN_MASK) == 0 ))
401             {
402                 sprintf(logbuff, "-");
403             }
404         else
405             {
406                 sprintf(logbuff, " ");
407             }
408         LCD_string(logbuff);
409         printf("%s", logbuff);
410         sprintf(logbuff, "%2.1fv ", meas_data[0] );
411         LCD_string(logbuff);

```

```
main.c

412     sprintf(logbuff, "%fV   ", meas_data[0] );
413     printf("%s", logbuff);
414
415     sprintf(logbuff, "I:");
416     LCD_string(logbuff);
417     printf("%s", logbuff);
418     if(((status_byte & PWRMETER_DCMODE_MASK) != 0) && ((status_byte &
PWRMETER_DCCURR_SIGN_MASK) == 0 ))
419     {
420         sprintf(logbuff, "-");
421     }
422     else
423     {
424         sprintf(logbuff, " ");
425     }
426     LCD_string(logbuff);
427     printf("%s", logbuff);
428     sprintf(logbuff, "%1.2fA   ", meas_data[1] );
429     LCD_string(logbuff);
430     sprintf(logbuff, "%fA", meas_data[1] );
431     printf("%s\r\n", logbuff);
432
433     sprintf(logbuff, "P:");
434     LCD_string(logbuff);
435     printf("%s", logbuff);
436
437     if((status_byte & PWRMETER_PA_SIGN_MASK) == 0)
438     {
439         sprintf(logbuff, "-");
440     }
441     else
442     {
443         sprintf(logbuff, " ");
444     }
445     LCD_string(logbuff);
446     printf("%s", logbuff);
447     sprintf(logbuff, "%2.1fW ", meas_data[2]);
448     LCD_string(logbuff);
449     sprintf(logbuff, "%fW   ", meas_data[2]);
450     printf("%s", logbuff);
451
452     sprintf(logbuff, "Q:");
453     LCD_string(logbuff);
454     printf("%s", logbuff);
455     if((status_byte & PWRMETER_PR_SIGN_MASK ) == 0 )
456     {
457         sprintf(logbuff, "-");
458     }
459     else
460     {
461         sprintf(logbuff, " ");
462     }
463     LCD_string(logbuff);
464     printf("%s", logbuff);
465     sprintf(logbuff, "%2.1fvar   ", meas_data[3] );
466     LCD_string(logbuff);
467     sprintf(logbuff, "%fvar", meas_data[3] );
```

```
main.c

468     printf("%s\r\n", logbuff);
469
470     sprintf(logbuff, "S: %2.1fVA", meas_data[4]);
471     LCD_string(logbuff);
472     sprintf(logbuff, "S: %fVA    ", meas_data[4]);
473     printf("%s", logbuff);
474     sprintf(logbuff, "f:%2.2fHz    ", meas_data[5]);
475     LCD_string(logbuff);
476     sprintf(logbuff, "f: %fHz", meas_data[5]);
477     printf("%s\r\n", logbuff);
478
479     if((status_byte & PWRMETER_DCMODE_MASK) != 0)
480     {
481         sprintf(logbuff, "    DC mode    ");
482     }
483     else
484     {
485         sprintf(logbuff, "    AC mode    ");
486     }
487     LCD_string(logbuff);
488     printf("%s\r\n-----\r\n", logbuff);
489
490     break;
491 }
492 }
493 #endif
494
495 }
496 return 0;
497 }
498
499
```

```
my_nucleo.h

1 /*
2  * my_nucleo.h
3  *
4  *   Author: Bc. Jan Rada
5  */
6
7 #ifndef MY_NUCLEO_H_
8 #define MY_NUCLEO_H_
9
10 #include "stm32f3xx.h"
11 #include <stdbool.h>
12 #include <stdio.h>
13
14 typedef enum {
15   ioPortOutputPP, // vystup typu Push-Pull
16   ioPortOutputOC, // vystup typu Open Collector
17   ioPortAnalog, // analogovy vstup - pro prevodnik
18   ioPortInputFloat, // vstup bez Pull-Up/Down
19   ioPortInputPU, // vstup s Pull-Up
20   ioPortInputPD, // vstup s Pull-Down
21   ioPortAlternatePP, // alternativni vystup - Push-Pull
22   ioPortAlternateOC // alternativni vystup - Open Collector
23 } eIoPortModes;
24
25 bool Nucleo_SetPinGPIO(GPIO_TypeDef *gpio, uint32_t bitnum, eIoPortModes mode);
26 bool Nucleo_SetPinAFGPIO(GPIO_TypeDef *gpio, uint32_t bitnum, uint32_t afValue);
27 void GPIOToggle(GPIO_TypeDef *gpio, uint32_t bitnum);
28 bool GPIORead(GPIO_TypeDef *gpio, uint32_t bitnum);
29 void GPIOWrite(GPIO_TypeDef *gpio, uint32_t bitnum, bool state);
30
31 // fce z DP_01
32 bool PLLInit(void); // nastaveni taktu procesoru na 64 MHz
33
34 // fce z DP_02
35 int __io_putchar(int ch); // pro komunikaci po seriaku s PC
36 int __io_getchar(void); // pro komunikaci po seriaku s PC
37 int UsartPCSend(char c); // vyslani znaku po USART2
38 void UsartPCString(char *txt); // vyslani retezce po USART2
39 int UsartPCRecv(void); // prijem po USART2
40 bool IsUsartPCRecv(void); // priznak, ze je neco v bufferu
41 void UsartPCInit(int baudSpeed); // nastaveni komunikace po USART2 pomoci printf po VCP
42
43 #define LED_BOARD GPIOB,3
44
45 #endif /* NUCLEO_CORE_H_ */
46
```

my\_nucleo.c

```
1 /*
2  * my_nucleo.c
3  *
4  *     Author: Bc. Jan Rada
5  */
6
7 #include "my_nucleo.h"
8
9 bool Nucleo_SetPinGPIO(GPIO_TypeDef *gpio, uint32_t bitnum, eIoPortModes mode)
10 {
11     uint32_t enr_mask = 0;    // hodnota do xxENR registru
12     uint32_t rstr_mask = 0;  // hodnota do xxRSTR registru
13
14     // detekce který GPIO
15     switch((uint32_t)gpio)
16     {
17     case (uint32_t)GPIOA:
18         enr_mask = RCC_AHBENR_GPIOAEN;
19         rstr_mask = RCC_AHBRSTR_GPIOARST;
20         break;
21     case (uint32_t)GPIOB:
22         enr_mask = RCC_AHBENR_GPIOBEN;
23         rstr_mask = RCC_AHBRSTR_GPIOBRSR;
24         break;
25     case (uint32_t)GPIOC:
26         enr_mask = RCC_AHBENR_GPIOCEN;
27         rstr_mask = RCC_AHBRSTR_GPIOCRST;
28         break;
29     case (uint32_t)GPIOD:
30         enr_mask = RCC_AHBENR_GPIODEN;
31         rstr_mask = RCC_AHBRSTR_GPIODRST;
32         break;
33     case (uint32_t)GPIOF:
34         enr_mask = RCC_AHBENR_GPIOFEN;
35         rstr_mask = RCC_AHBRSTR_GPIOFRST;
36         break;
37     }
38
39     if((enr_mask == 0) || (rstr_mask == 0)) // nevybran GPIO
40         return false;
41
42     // inicializace vybraného
43     if(!(RCC->AHBENR & enr_mask)) // nejsou povoleny hodiny ?
44     {
45         RCC->AHBENR |= enr_mask;    // povolit hodiny periferie
46         RCC->AHBRSTR |= rstr_mask;  // proved reset periferie
47         RCC->AHBRSTR &= ~rstr_mask; // a konec resetu
48     }
49
50     // nastaveni konfiguracnich bitu do defalutniho stavu (nemusi byt od drive)
51     gpio->MODER &= ~(0x03 << (2 * bitnum)); // vynuluji prislusne 2 bity v registru
52     gpio->PUPDR &= ~(0x03 << (2 * bitnum)); // vynuluji prislusne 2 bity v registru
53     gpio->OSPEEDR &= ~(0x03 << (2 * bitnum)); // vynuluji prislusne 2 bity v registru
54
55     // nastaveni registru podle typu vystupu/vstupu
56     switch(mode)
57     {
```

```

my_nucleo.c

58 case ioPortOutputOC:
59 case ioPortOutputPP:
60   gpio->MODER |= (0x01 << (2 * bitnum));           // 01 = output
61   gpio->PUPDR &= ~(0x03 << (2 * bitnum));         // 00 = bez PU/PD
62   gpio->OSPEEDR |= 0x03 << (2 * bitnum);         // 11 = high speed
63   if(mode == ioPortOutputOC)                    // Open Collector ?
64     gpio->OTYPER |= 0x01 << bitnum;              // 1 = Open Collector (Open Drain)
65   else
66     gpio->OTYPER &= ~(0x01 << bitnum);          // 0 = Push-Pull
67   break;
68 case ioPortInputPU:                             // MODER bity 00 = input
69   gpio->PUPDR &= ~(0x03 << (2 * bitnum));       // vycistit bity
70   gpio->PUPDR |= 0x01 << (2 * bitnum);          // 01 = Pull-Up
71   break;
72 case ioPortInputPD:
73   gpio->PUPDR &= ~(0x03 << (2 * bitnum));       // vycistit bity
74   gpio->PUPDR |= 0x02 << (2 * bitnum);          // 10 = Pull-Down
75   break;
76 case ioPortInputFloat:                          // 00 = input
77   gpio->PUPDR &= ~(0x03 << (2 * bitnum));       // 00 = bez Pull-Up/Down
78   break;
79 case ioPortAlternatePP:
80 case ioPortAlternateOC:
81   gpio->MODER |= 0x00000002 << (2 * bitnum);    // set bits
82
83   if(mode == ioPortAlternateOC)
84     gpio->OTYPER |= 0x00000001 << bitnum;       // 1 = Open Collector, jeden bit na GPIO
85   else
86     gpio->OTYPER &= ~(0x00000001 << bitnum);    // 0 = Push-Pull, jeden bit na GPIO
87
88   gpio->PUPDR &= ~(0x00000003 << (2 * bitnum)); // 00 = bez Pull-Up / Pull-Down
89   gpio->OSPEEDR |= 0x00000003 << (2 * bitnum); // 11 = high speed
90   // nutnost nastavit AFR registry !!!
91   break;
92 case ioPortAnalog:                              // 11 = analog
93   gpio->MODER |= 0x03 << (2 * bitnum);          // nastav bity
94   gpio->PUPDR &= ~(0x03 << (2 * bitnum));       // 00 = analog
95   break;
96 default:                                        // neznamy rezim ?
97   return false; // priznak chyby
98 }
99 return true; // priznak OK
100}
101
102 bool Nucleo_SetPinAFGPIO(GPIO_TypeDef *gpio, uint32_t bitnum, uint32_t afValue)
103{
104   gpio->AFR[(bitnum < 8) ? 0 : 1] &= ~(0x0f << (4 * (bitnum & 0x07))); // vynuluj AF bity
105   gpio->AFR[(bitnum < 8) ? 0 : 1] |= ((afValue & 0x0f) << (4 * (bitnum & 0x07))); // nastav
   AF bity
106   return true;
107}
108
109 void GPIOToggle(GPIO_TypeDef *gpio, uint32_t bitnum)
110{
111   gpio->ODR ^= 1 << bitnum; // vystupni registr - 1 bit = 1 vystup, operace XOR
112}
113

```

```

my_nucleo.c

114 bool GPIORead(GPIO_TypeDef *gpio, uint32_t bitnum)
115 {
116     return ((gpio->IDR & (1 << bitnum)) != 0);    // porovnani = vynucena logicka hodnota
117 }
118
119 void GPIOwrite(GPIO_TypeDef *gpio, uint32_t bitnum, bool state)
120 {
121     if(state)
122         gpio->BSRR = (1 << bitnum);    // nastav do log. 1
123     else
124         gpio->BSRR = (1 << (16 + bitnum));    // nastav do log. 0
125 }
126
127 bool PLLInit(void)
128 {
129     uint32_t t;
130
131     if(!(RCC->CR & RCC_CR_HSION))            // HSI not running ?
132     {
133         RCC->CR |= RCC_CR_HSION;            // enable
134
135         t = 100;
136         while(!(RCC->CR & RCC_CR_HSION) && t) // cekej na ON
137             t--;
138         if(!t)
139             return false;                    // chybovy priznak
140     }
141
142     if(RCC->CR & RCC_CR_PLLON)                // bezi PLL ?
143     {
144         RCC->CR &= ~RCC_CR_PLLON;            // zakazani PLL
145     }
146
147     RCC->CFGR &= ~RCC_CFGR_SW;                // nastav HSI jako zdroj HCLK
148
149     RCC->CFGR = 0;                            // RESET vseho
150
151     RCC->CFGR |= RCC_CFGR_PPRE1_DIV2;         // PRE1 = 100 = /2 pro APB1 (32 MHz)
152     RCC->CFGR |= RCC_CFGR_PLLSRC_HSI_DIV2;    // HSI / 2 jako zdroj pro PLL
153     RCC->CFGR |= RCC_CFGR_PLLMUL16;          // PLLMUL = 1111 = nastav nasobeni x16
154
155     RCC->CR |= RCC_CR_PLLON;                  // spust PLL
156
157     t = 100;
158     while(!(RCC->CR & RCC_CR_PLLRDY) && t)    // cekej na ON
159         t--;
160
161     if (!t)
162         return false;                        // chybovy priznak
163
164     // Nastavení waitState... pokud dám rychlejší frekvenci než zvládá flashka vydávat
165     FLASH->ACR &= ~FLASH_ACR_LATENCY;        // vymazani LATENCY
166     FLASH->ACR |= FLASH_ACR_LATENCY_1;       // 2 cekaci cykly (48 < HCLK < 72 MHz)
167
168     RCC->CFGR |= RCC_CFGR_SW_PLL;            // nastav PLL jako zdroj HCLK
169
170     t = 100;

```

```
my_nucleo.c

171 while(!((RCC->CFGR & RCC_CFGR_SWS) == RCC_CFGR_SWS_PLL) && t) // cekej na overeni SWS
172     t--;
173
174 if (!t)
175     return false;
176
177 return true;
178 }
179
180 // fce z DP_02
181
182 int __io_putchar(int ch) { return UsartPCSend(ch); }
183 int __io_getchar(void) { return UsartPCRecv(); }
184
185 int UsartPCSend(char c)
186 {
187     while(!(USART2->ISR & USART_ISR_TXE)) // cekej dokud neni volny TDR
188         ;
189     USART2->TDR = c; // zapis do Transmit Data Registru k odeslani
190     return c;
191 }
192
193 void UsartPCString(char *txt)
194 {
195     while(*txt)
196     {
197         UsartPCSend(*txt);
198         txt++;
199     }
200 }
201
202 int UsartPCRecv(void)
203 {
204     while(!(USART2->ISR & USART_ISR_RXNE)) // cekej dokud neco neprijde do RDR
205         ;
206     return USART2->RDR; // vyciti a vrat jako hodnotu
207 }
208
209 bool IsUsartPCRecv(void) // priznak, ze je neco v bufferu
210 {
211     return (USART2->ISR & USART_ISR_RXNE) != 0; // podminka vynuti true / false
212 }
213
214 void UsartPCInit(int baudSpeed)
215 {
216     Nucleo_SetPinGPIO(GPIOA, 2, ioPortAlternatePP);
217     Nucleo_SetPinAFGPIO(GPIOA, 2, 7); // AF7 je USART2
218     Nucleo_SetPinGPIO(GPIOA, 15, ioPortAlternatePP);
219     Nucleo_SetPinAFGPIO(GPIOA, 15, 7); // AF7 je USART2
220
221
222     if(!(RCC->APB1ENR & RCC_APB1ENR_USART2EN)) // neni povolen USART2
223     {
224         RCC->APB1ENR |= RCC_APB1ENR_USART2EN;
225         RCC->APB1RSTR |= RCC_APB1RSTR_USART2RST;
226         RCC->APB1RSTR &= ~RCC_APB1RSTR_USART2RST;
227     }

```



```
my_nucleo.c

228
229 USART2->CR1 = USART_CR1_RE | USART_CR1_TE; // staci povoleni prijmu a vysilani
230 USART2->CR2 = 0; // nic specialniho
231 USART2->CR3 = 0; // nic specialniho
232
233 // POZOR USART2 je pripojen na APB1 sbernici => pro tak procesoru 64MHz je 32MHz
234
235 { // vypocet BRR podle pozadovaneho Baud-rate a podle clocku
236 uint8_t sampling = (USART2->CR1 & USART_CR1_OVER8) ? 8 : 16;
237 uint32_t apb1, mantisa, tmp, fraction;
238 uint32_t bitval = 0;
239 uint32_t divider = 1;
240
241 bitval = (RCC->CFGR & (0x07 << 8)) >> 8; // PPRE1 [10:8] to lower 3 bits
242 if (bitval & 0x4) // 1xx
243     divider = 1 << ((bitval & 0x03) + 1); // 0 = /2, 1 = /4
244 else
245     divider = 1; // 0xx = not divided
246
247 SystemCoreClockUpdate(); // pro jistotu si nastav SystemCoreClock
248
249 apb1 = SystemCoreClock / divider;
250 mantisa = apb1 * 16 / (sampling * baudSpeed); // part of 16th
251 tmp = mantisa / 16;
252
253 fraction = mantisa - (tmp * 16); // remain after 16 division
254 USART2->BRR = (tmp << 4) | (fraction & 0x0f);
255 }
256
257 /* baudSpeed == 115200, fCLK == 64 MHz
258 USART2->BRR = (17 << 4) // USARTDIV = 17.3611111
259 | 6 // mantisa = 13 (posunuta o 4 bity vlevo)
260 ; // fraction = 6 (zaokrouhleno na 6/16)
261 */
262 /* baudSpeed == 38400, fCLK == 8MHz
263 USART2->BRR = (13 << 4); // USARTDIV = 13.02083333
264 // mantisa = 13 (posunuta o 4 bity vlevo)
265 // fraction = 0 (zaokrouhleno na 0/16)
266 */
267
268 /* baudSpeed == 460800, fCLK == 64 MHz
269 USART2->BRR = (4 << 4) // USARTDIV = 4.340277778 pro 64MHz a 460800Bd, 8.680555556 pro
64MHz a 230400Bd, 13.02083333 pro 8MHz a 38400Bd
270 | 5; // USARTDIV = fCLK / (baud x 8 x (2 - OVER8))
271 // mantisa = 4 (posunuta o 4 bity vlevo)
272 // fraction = 5 (zaokrouhleno na 5/16)
273 */
274
275 USART2->CR1 |= USART_CR1_UE; // povolen blok USARTu
276
277 // zrusit bufferovani vystupu i vstupu
278 setvbuf(stdout, NULL, _IONBF, 0);
279 setvbuf(stdin, NULL, _IONBF, 0);
280 }
281
```

PWR\_meter.h

```
1/*
2 * PWR_meter.h
3 *
4 *     Author: Bc. Jan Rada
5 */
6
7#ifndef PWR_METER_H_
8#define PWR_METER_H_
9
10#include "my_nucleo.h"
11
12// fce z DP_03
13int UsartPWRmSend(char c);           // vyslani znaku po USART1
14void UsartPWRmString(uint8_t *msg); // vyslani retezce po USART1
15int UsartPWRmRecv(void);            // prijem po USART1
16bool IsUsartPWRmRecv(void);         // priznak, ze je neco v bufferu
17void UsartPWRmInit(int baudSpeed);  // nastaveni komunikace po USART1 s PWR meterem
18
19#endif /* PWR_METER_H_ */
20
```

```
PWR_meter.c

1 /*
2  * PWR_meter.c
3  *
4  *   Author: Bc. Jan Rada
5  */
6
7 // fce z DP_03
8
9 #include "my_nucleo.h"
10
11 int UsartPWRmSend(char c)
12 {
13     while(!(USART1->ISR & USART_ISR_TXE)) // cekej dokud neni volny TDR
14         ;
15     * ((__IO uint8_t *) & USART1->TDR) = (uint8_t) c; // zapis do Transmit Data Registru k
        odeslani
16     return c;
17 }
18
19 void UsartPWRmString(uint8_t *msg)
20 {
21     uint8_t i;
22     uint8_t len = msg[1]; // delka zpravy - druhy Byte znamena delku zpravy
23
24     for(i = 0; i < len; i++)
25     {
26         UsartPWRmSend(msg[i]);
27     }
28
29     /*
30     while(*txt)
31     {
32         UsartPWRmSend(*txt);
33         txt++;
34     }
35     */
36 }
37
38 int UsartPWRmRecv(void)
39 {
40     while(!(USART1->ISR & USART_ISR_RXNE)) // cekej dokud neco neprijde do RDR
41         ;
42     return USART1->RDR; // vyciti a vrat jako hodnotu
43 }
44
45 bool IsUsartPWRmRecv(void) // priznak, ze je neco v bufferu
46 {
47     return (USART1->ISR & USART_ISR_RXNE) != 0; // podminka vynuti true / false
48 }
49
50 void UsartPWRmInit(int baudSpeed)
51 {
52     Nucleo_SetPinGPIO(GPIOA, 9, ioPortAlternatePP);
53     Nucleo_SetPinAFGPIO(GPIOA, 9, 7); // AF7 je USART1
54     Nucleo_SetPinGPIO(GPIOA, 10, ioPortAlternatePP);
55     Nucleo_SetPinAFGPIO(GPIOA, 10, 7); // AF7 je USART1
56 }
```

PWR\_meter.c

```

57
58 if(!(RCC->APB2ENR & RCC_APB2ENR_USART1EN)) // neni povolen USART1
59 {
60     RCC->APB2ENR |= RCC_APB2ENR_USART1EN;
61     RCC->APB2RSTR |= RCC_APB2RSTR_USART1RST;
62     RCC->APB2RSTR &= ~RCC_APB2RSTR_USART1RST;
63 }
64
65 USART1->CR1 = USART_CR1_RE | USART_CR1_TE; // staci povoleni prijmu a vysilani
66 USART1->CR2 = 0; // nic specialniho
67 USART1->CR3 = 0; // nic specialniho
68
69 // POZOR USART1 je pripojen na APB2 sbernici => pro takt procesoru 64MHz je 64MHz
70
71 { // vypocet BRR podle pozadovaneho Baud-rate a podle clocku
72     uint8_t sampling = (USART1->CR1 & USART_CR1_OVER8) ? 8 : 16;
73     uint32_t apb2, tmp, fraction;
74     uint64_t mantisa;
75     uint32_t bitval = 0;
76     uint32_t divider = 1;
77
78     bitval = (RCC->CFGR & (0x07 << 11)) >> 11; // PPRE2 [13:11] to lower 3 bits
79     if (bitval & 0x4) // 1xx
80         divider = 1 << ((bitval & 0x03) + 1); // 0 = /2, 1 = /4
81     else
82         divider = 1; // 0xx = not divided
83
84     SystemCoreClockUpdate(); // pro jistotu si nastav SystemCoreClock
85
86     apb2 = SystemCoreClock / divider;
87     mantisa = apb2 * 16 / (sampling * baudSpeed); // part of 16th
88     tmp = mantisa / 16;
89
90     fraction = mantisa - (tmp * 16); // remain after 16 division
91     USART1->BRR = (tmp << 4) | (fraction & 0x0f);
92 }
93
94 /* baudSpeed == 115200, fCLK == 32 MHz
95 USART1->BRR = (17 << 4) // USARTDIV = 17.3611111
96     | 6 // mantisa = 13 (posunuta o 4 bity vlevo)
97     ; // fraction = 6 (zaokrouhлено na 6/16)
98 */
99 /* baudSpeed == 38400, fCLK == 4MHz
100 USART1->BRR = (13 << 4); // USARTDIV = 13.02083333
101 // mantisa = 13 (posunuta o 4 bity vlevo)
102 // fraction = 0 (zaokrouhлено na 0/16)
103 */
104
105 /* baudSpeed == 460800, fCLK == 32 MHz
106 USART1->BRR = (4 << 4) // USARTDIV = 4.340277778 pro 64MHz a 460800Bd, 8.680555556 pro
64MHz a 230400Bd, 13.02083333 pro 8MHz a 38400Bd
107     | 5; // USARTDIV = fCLK / (baud x 8 x (2 - OVER8))
108 // mantisa = 4 (posunuta o 4 bity vlevo)
109 // fraction = 5 (zaokrouhлено na 5/16)
110 */
111
112 USART1->CR1 |= USART_CR1_UE; // povolen blok USARTu

```

PWR\_meter.c

```
113
114 // zrusit bufferovani vystupu i vstupu
115 setvbuf(stdout, NULL, _IONBF, 0);
116 setvbuf(stdin, NULL, _IONBF, 0);
117 }
118
```

EADOGS\_164.h

```
1 /*
2  * EADOGS_164.h
3  *
4  *     Author: Bc. Jan Rada
5  */
6
7 #ifndef EADOGS_164_H_
8 #define EADOGS_164_H_
9
10 #include "my_nucleo.h"
11
12 #define SPI_IS_BUSY(SPIx) (((SPIx)->SR & (SPI_SR_TXE | SPI_SR_RXNE)) == 0 || ((SPIx)->SR &
    SPI_SR_BSY))
13
14 #define LCD_WHITE    GPIOA,6
15 #define LCD_GREEN    GPIOA,7
16 #define LCD_RED      GPIOA,2
17
18 void SPI1Init(void);
19 void LCD_send(uint8_t val, bool rs);
20 void LCD_char(char znak);
21 void LCD_string(char *txt);
22 bool LCD_reset(void);
23 void LCD_init(void);
24
25
26 #endif /* EADOGS_164_H_ */
27
```

```

EADOGS_164.c

1/*
2 * EADOGS_164.c
3 *
4 *     Author: Bc. Jan Rada
5 */
6
7#include "EADOGS_164.h"
8
9void SPI1Init(void)
10{
11    Nucleo_SetPinGPIO(GPIOA, 8, ioPortOutputPP);           // reset displeje
12    GPIOWrite(GPIOA, 8, true);
13
14    Nucleo_SetPinGPIO(GPIOA, 11, ioPortOutputPP);         // SPI1_CS
15    GPIOWrite(GPIOA, 11, true);
16
17    Nucleo_SetPinGPIO(GPIOB, 5, ioPortAlternatePP);       // SPI1_MOSI
18    Nucleo_SetPinAFGPIO(GPIOB, 5, 5);
19
20    Nucleo_SetPinGPIO(GPIOB, 3, ioPortAlternatePP);       // SPI1_SCK
21    Nucleo_SetPinAFGPIO(GPIOB, 3, 5);
22
23    if(!(RCC->APB2ENR & RCC_APB2ENR_SPI1EN))
24    {
25        RCC->APB2ENR |= RCC_APB2ENR_SPI1EN;
26        RCC->APB2RSTR |= RCC_APB2RSTR_SPI1RST;
27        RCC->APB2RSTR &= ~RCC_APB2RSTR_SPI1RST;
28    }
29
30    SPI1->CR1 = 0 | SPI_CR1_MSTR
31              | SPI_CR1_CPOL           // CPOL = 1
32              | SPI_CR1_CPHA           // CPHA = 1
33              | SPI_CR1_LSBFIRST       // LSB first
34              | SPI_CR1_BR_2 | SPI_CR1_BR_0 // BR = 101 = fclk/64
35    //    | SPI_CR1_BR_1           // BR = 111 = fclk/256 -> pokud pouziji takt procesoru
64 MHz
36              | SPI_CR1_SSI | SPI_CR1_SSM; // CS rizeny softwarove
37
38    SPI1->CR2 = SPI_CR2_DS_0 | SPI_CR2_DS_1 | SPI_CR2_DS_2; // zadne preruseni ani
DMA, DS = 0111 = 8b data (default, neni treba nastavovat)
39
40    SPI1->CR1 |= SPI_CR1_SPE; // enable
41}
42
43#define SPI_IS_BUSY(SPIx) (((SPIx)->SR & (SPI_SR_TXE | SPI_SR_RXNE)) == 0 || ((SPIx)->SR &
SPI_SR_BSY))
44
45void LCD_send(uint8_t val, bool rs)
46{
47    uint8_t startB = 0x1f, lowerB = 0, highB = 0;
48
49    if(rs)
50        startB |= 0x40;
51
52    lowerB = (val & ~0xf0);
53    highB = ((val & ~0xf) >> 4);
54

```

EADOGS\_164.c

```
55   GPIOWrite(GPIOA, 11, false);
56
57   * ((__IO uint8_t *) & SPI1->DR) = (uint8_t) startB; // SPI1->DR = startB;
58   while (SPI_IS_BUSY(SPI1)) // POZOR - BLOKUJICI FUNKCE
59       ;
60
61   * ((__IO uint8_t *) & SPI1->DR) = (uint8_t) lowerB; // SPI1->DR = lowerB;
62   while (SPI_IS_BUSY(SPI1)) // POZOR - BLOKUJICI FUNKCE
63       ;
64
65   * ((__IO uint8_t *) & SPI1->DR) = (uint8_t) highB; // SPI1->DR = highB;
66   while (SPI_IS_BUSY(SPI1)) // POZOR - BLOKUJICI FUNKCE
67       ;
68
69   GPIOWrite(GPIOA, 11, true);
70 }
71
72 void LCD_char(char znak)
73 {
74     LCD_send(znak, 1);
75 }
76
77 void LCD_string(char *txt)
78 {
79     while(*txt)
80     {
81         LCD_char(*txt);
82         txt++;
83     }
84 }
85
86 bool LCD_reset(void)
87 {
88     uint16_t w, x = 0;
89
90     GPIOWrite(GPIOA, 8, false);
91
92     for (w = 0; w < 1000; w++)
93         x++; // Dummy increment prevents optimisation
94
95     GPIOWrite(GPIOA, 8, true);
96
97     for (w = 0; w < 1000; w++)
98         x++;
99
100    return (x > 0); // Trick to keep variable unoptimised ...
101 }
102
103 void LCD_clear_all(void)
104 {
105     LCD_send(0x01, 0);
106 }
107
108 void LCD_init(void)
109 {
110     LCD_reset();
111 }
```



EADOGS\_164.c

```
112 Nucleo_SetPinGPIO(LCD_GREEN, ioPortOutputPP); // CE
113 GPIOWrite(GPIOA, 7, true); // false = sviti, true = nesviti
114 Nucleo_SetPinGPIO(LCD_WHITE, ioPortOutputPP); // CW
115 GPIOWrite(GPIOA, 6, true); // false = sviti, true = nesviti
116 /*
117 Nucleo_SetPinGPIO(LCD_RED, ioPortOutputPP); // CR
118 GPIOWrite(GPIOA, 2, true); // false = sviti, true = nesviti
119 // nelze pouzít - konflikt s USART2
120 */
121
122 uint8_t InitData[] = {
123     0x3A, // 8 bit data length extension Bit RE=1; REV=0
124     0x09, // 4 line display
125     0x06, // bottom view
126     0x1E, // BS1=1
127     0x39, // 8 bit data length extension Bit RE=0; IS=1
128     0x1B, // BS0=1 -> Bias=1/6
129     0x6C, // Divider on and set value
130     0x54, // Booster on and set contrast (DB1=C5, DB0=C4)
131     0x7A, // Set contrast (DB3-DB0=C3-C0)
132     0x38, // 8 bit data length extension Bit RE=0; IS=0
133     0x0F // Display on, cursor on, blink on
134 };
135
136 uint8_t i;
137 for(i=0; i<sizeof(InitData); i++)
138 {
139     LCD_send(InitData[i], 0);
140 }
141
142 LCD_clear_all();
143
144 }
145
146
147
```