



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## REALIZACE CHYTRÉHO ELEKTROMĚRU

REALIZATION OF A SMART METER

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Lukáš Cu

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Lieskovan

BRNO 2022

# Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

**Student:** Bc. Lukáš Cu

**ID:** 197767

**Ročník:** 2

**Akademický rok:** 2021/22

**NÁZEV TÉMATU:**

## Realizace chytrého elektroměru

### POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je realizace chytrého elektroměru. K této realizaci bude využita platforma Raspberry Pi. V teoretické části student popíše komunikační protokol DLMS/COSEM a jednodeskové počítače Raspberry Pi. V praktické části student realizuje měření aktuální spotřeby, celkové spotřeby a frekvence v síti. Součástí realizace bude přepínání vysokého a nízkého tarifu. Všechny měřené hodnoty budou odesílány pomocí DLMS/COSEM patřičnými OBIS kódy.

### DOPORUČENÁ LITERATURA:

- [1] Gurux DLMS Server: <https://www.gurux.fi/Gurux.DLMS.Server>
- [2] JIRKA, Bc Matěj. Framework DLMS/COSEM pro sběr dat v AMM systémech.

**Termín zadání:** 7.2.2022

**Termín odevzdání:** 24.5.2022

**Vedoucí práce:** Ing. Tomáš Lieskovan

**doc. Ing. Jan Hajný, Ph.D.**  
předseda rady studijního programu

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Semestrální práce se zabývá problematikou chytrého měření, konkrétně měřením elektřiny. Čtenáři přibližuje základní pojmy s tím spojené a popisuje technologii použitou k realizaci takového elektroměru.

Praktickým výstupem této práce je reálný chytrý elektroměr vyrobený z jednodeskové platformy Raspberry Pi 3 B+, platformy Arduino, a dvou externích modulů pro měření stavu sítě. Těmito moduly jsou proudový senzor SCT-013-000 a senzor napětí ZMPT101B. Programové řešení pro ovládání jednotlivých komponentů a pro práci s těmito daty je uskutečněno pomocí kombinace vlastního kódu a externí knihovny společnosti Gurux.

## **KLÍČOVÁ SLOVA**

Arduino, DLMS/COSEM elektřina, chytrý elektroměr, proud, Python, Raspberry Pi

## **ABSTRACT**

This thesis deals with the issue of smart metering, specifically electricity metering. It introduces the reader to the basic concepts associated with it and describes the technology used to implement such an electricity meter.

The practical output of this work is a real smart electricity meter made of a single-board platform Raspberry Pi 3 B, the Arduino platform and two external modules for measuring the state of electrical network. These modules are current sensor SCT-013-000 and a voltage sensor ZMPT101B. The software solution for controlling individual components and for working with the data is implemented using a combination of code created specifically for this purpose and of code from an external library belonging to Gurux company.

## **KEYWORDS**

Raspberry Pi, current, electricity, python, smart electricity meter, DLMS/COSEM, Arduino

CU, Lukáš. *Realizace chytrého elektroměru*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2021, 46 s. Diplomová práce. Vedoucí práce: Ing. Tomáš Lieskovan

## Prohlášení autora o původnosti díla

**Jméno a příjmení autora:** Bc. Lukáš Cu  
**VUT ID autora:** 197767  
**Typ práce:** Diplomová práce  
**Akademický rok:** 2021/22  
**Téma závěrečné práce:** Realizace chytrého elektroměru

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora\*

---

\*Autor podepisuje pouze v tištěné verzi.

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu Diplomové práce panu Ing. Tomáší Lieskovanovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci a Ing. Richardu Vaňkovi za velice cenné vstupy a rady k problematice, která mi byla naprosto neznámá.

# Obsah

<b>Úvod</b>	<b>10</b>
<b>1 Teoretická část</b>	<b>12</b>
1.1 Chytrý elektroměr . . . . .	12
1.2 Raspberry Pi . . . . .	12
1.3 Arduino . . . . .	14
1.4 Měření proudu . . . . .	15
1.5 Měření napětí a frekvence sítě . . . . .	18
1.6 DLMS/COSEM . . . . .	19
1.6.1 Služby DLMS . . . . .	20
<b>2 Praktická část</b>	<b>23</b>
2.1 Zapojení . . . . .	23
2.1.1 Proudový senzor . . . . .	23
2.1.2 Senzor napětí . . . . .	23
2.2 Programové řešení . . . . .	25
2.2.1 Použité knihovny . . . . .	25
2.2.2 Funkce pro měření proudu a spotřeby . . . . .	25
2.2.3 Funkce pro měření napětí a frekvence . . . . .	26
2.2.4 Implementace protokolu DLMS/COSEM . . . . .	27
2.2.5 Měření spotřeby na základě tarifů . . . . .	28
2.2.6 Ukládání dat . . . . .	29
2.2.7 Zobrazování dat . . . . .	29
2.3 Výsledky . . . . .	31
<b>Závěr</b>	<b>42</b>
<b>Literatura</b>	<b>44</b>
<b>Seznam symbolů a zkratk</b>	<b>46</b>

# Seznam obrázků

1.1	Raspberry Pi model 3 B+ . . . . .	13
1.2	Klon platformy Arduino UNO . . . . .	15
1.3	Proudový senzor SCT-013-000 . . . . .	16
1.4	Schéma Ampérova pravidla pravé ruky . . . . .	17
1.5	Klon platformy Arduino UNO s přidaným shieldem pro šroubovací konektory . . . . .	22
2.1	Schéma zapojení senzorů k platformě Arduino . . . . .	24
2.2	Vývojový diagram pro zjištění efektivního proudu a spotřeby. . . . .	34
2.3	Vývojový diagram pro zjištění efektivního napětí a frekvence. . . . .	35
2.4	Schéma původní realizace chytrého elektroměru . . . . .	36
2.5	Výpis funkcí měření v prostředí Arduino IDE . . . . .	37
2.6	Výpis funkcí měření ve zjednodušeném formátu. . . . .	38
2.7	Čtení hodnoty frekvence aplikací GXDLMS Director . . . . .	39
2.8	Databáze záznamů měření v PHPmyadmin . . . . .	39
2.9	Výpis hodnot měření webového serveru . . . . .	40
2.10	Výsledné schéma chytrého elektroměru . . . . .	41



# Seznam tabulek

1.1	Přehled fcí ZMPT . . . . .	18
-----	----------------------------	----

# Úvod

V dnešní době se více a více věcí automatizuje a je žádoucí z procesů co nejvíce vytěsnit lidský faktor, nebo člověku co nejvíce usnadnit práci. Ukázkou tohoto jevu je vlna takzvaných chytrých zařízení. Téměř každý elektrický spotřebič nejen v domácnosti má v posledních letech svoji chytrou verzi. Konektivita těchto zařízení tvoří ekosystém, který uživatel dokáže jednoduše spravovat, často na stejném místě. Například z mobilního telefonu nebo přes aplikaci na stolním počítači. Nejen z tohoto důvodu po těchto zařízeních každým dnem roste poptávka.

Chytré zařízení je elektronické zařízení, které je připojeno do sítě dalších zařízení a dokáže do jisté míry autonomně vykonávat svoji činnost. O své činnosti je pak schopno komunikovat se zbytkem sítě a tím jim buď zprostředkovat data nutná pro jejich funkci, nebo usnadnit přístup těmto dat na vzdáleném bodu. Může také fungovat na základě vstupů z ostatních zařízení sítě.

Chytrá zařízení jsou například chytré telefony, tablety, chytré hodinky a auta, ledničky a jiné spotřebiče. Pro jednoduchou práci s elektronikou v rámci domácnosti, budovy, nebo i většího celku je žádoucí, aby jejich funkce byly co nejdostupnější odkudkoliv.

Hlavním tématem této práce je realizace chytrého elektroměru. To znamená návrh a reálné složení funkčního elektroměru splňující funkce chytrého elektroměru. Těmito funkcemi se rozumí naměření požadovaných veličiny v elektrickém obvodu, výpis, uložení těchto a zobrazení těchto dat na příhodném místě.

Cílem této diplomové práce je zrealizovat chytrý elektroměr. Ten funguje tak, že naměřené hodnoty, v našem případě hodnoty proudu, frekvence, napětí a spotřeby, ukládá a zpřístupňuje ke čtení ve webovém prohlížeči.

První část práce uvádí čtenáře do teorie látky týkající se praktického výstupu diplomové práce. Popisuje součástky, principy a další technologie použité na teoretické úrovni. To znamená uvedení hardwaru pro zapojení, popis jeho funkcí a kontext jeho aplikace v praktickém výstupu. Těmito komponenty jsou jednodeskový počítač Raspberry Pi model 3 B+ a Arduino model Mega. Dále jsou k realizaci měření samotného použity různé moduly. Světovým standardem pro komunikaci chytrých zařízení je specifikace DLMS/COSEM. Ten je použit pro přenášení měřených dat z elektroměru a teoretická část práce jej tedy také popisuje.

Druhá část práce slouží jako vysvětlení samotného výstupu práce. Komponenty jsou tedy popsány na hardwarové úrovni a je uveden způsob a odůvodnění fyzického zapojení. Dále je popsáno programové řešení praktického výstupu a výsledky práce jako celku. Programově bylo nutné vyřešit měření hodnot, odesílání těchto hodnot v protokolu DLMS, ukládání hodnot a zobrazení hodnot přes webový prohlížeč. V uvedení výsledků je čtenář proveden chronologicky postupem procesů práce chytrého

elektroměru od změřených hodnot po jejich zobrazení pro uživatele.

# 1 Teoretická část

V této části je rozepsána teorie týkající se praktického výstupu diplomové práce. Je zde tedy popis chytrého elektroměru jako takového, jednodeskového počítače Raspberry Pi, platformy Arduino, použitého proudového senzoru, modulu pro měření napětí a frekvence sítě a protokolu DLMS/COSEM použitého ke komunikaci.

## 1.1 Chytrý elektroměr

Chytrým elektroměrem je zařízení, které v daném elektrickém obvodu zaznamenává aktuální stav vybraných veličin, jakými jsou například proud nebo napětí. Tato naměřená data následně pomocí komunikačního protokolu posílá drátovým nebo bezdrátovým způsobem do odpovídající řídicí jednotky, v našem případě Raspberry Pi, která tato data vyhodnocuje a zpracovává pro další využití. Hlavní výhodou chytrého zařízení, jako je chytrý elektroměr, je snadný přístup k relevantním informacím. To znamená, že naměřená data nejsou zobrazena jen přímo na elektroměru, ale existuje způsob jak tato data číst dálkově. Například v mobilní aplikaci nebo přes webový prohlížeč. Praktický výstup této práce data zapisuje do souboru čitelného webovým prohlížečem[1].

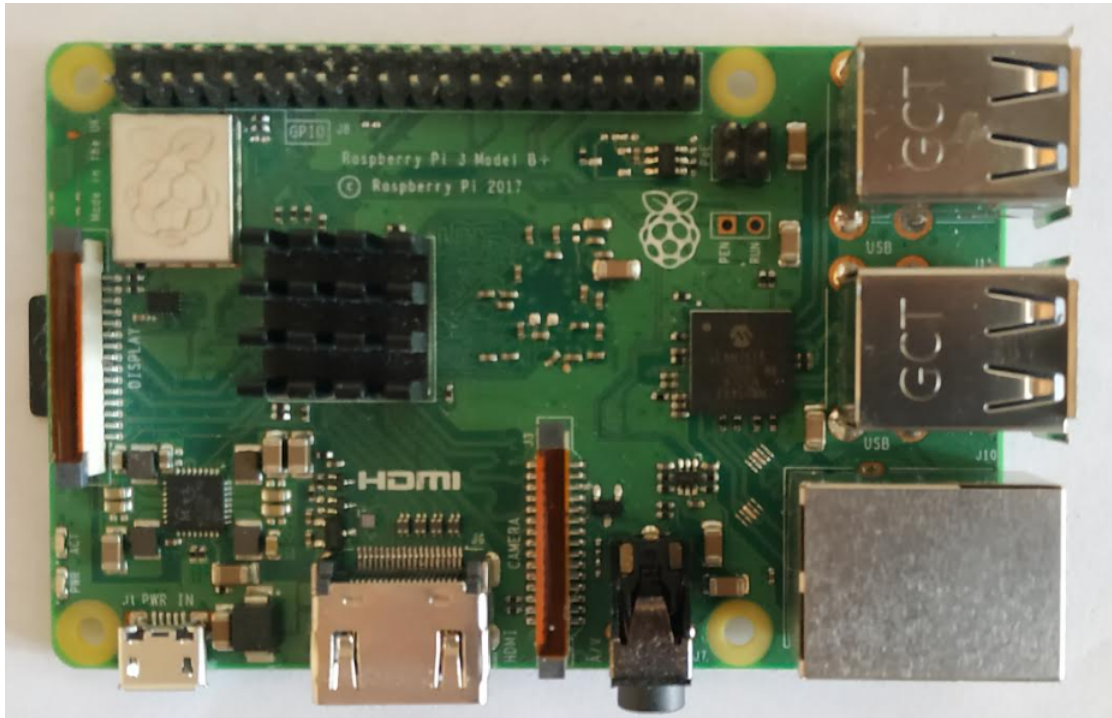
Komunikace mezi měřičem s řídicí jednotkou a zbytkem sítě je uskutečněn komunikačním protokolem buď drátově nebo bezdrátově (Bluetooth, Wi-fi, aj.). V případě našeho praktického výstupu je senzor zapojen přímo k odpovídajícím pinům na Raspberry Pi. Ten je dále připojen na Wi-fi síť takže teoreticky může bezdrátově odesílat data na server pro čtení zbytku domácí sítě. Chytrý měřič má oproti obyčejnému četné výhody. Mnoho dodavatelů je však prozatím nepodporuje, a zatím tedy slouží spíše pro vlastní využití[2].

## 1.2 Raspberry Pi

Jedná se o jednodeskový počítač vyvinut v roce 2012 britskou nadací Raspberry Pi Foundation. Jeho prvotním účelem bylo vytvořit levné a kompaktní zařízení, které zpřístupní programování studentům v rozvojových zemích. Díky své dostupnosti a kompaktnosti však získal globální popularitu. Chová se jako stolní počítač, ale díky svým parametrům může sloužit jako řídicí jednotka pro různá zařízení a nadšenci po celém světě jej právě takto využívají[3].

Od roku 2012 byly uvedeny čtyři generace Raspberry Pi, které se od sebe liší hardwarově. Každá generace zahrnuje model A a B. Model A je levnější a méně výkonnou verzí. Tato práce pracuje s generací 3 modelem B. Raspberry Pi má mnoho

možností operačního systému. Nejrozšířenějším z nich je Raspbian. Jedná se o operační systém založený na Debianu vyvinutý přímo nadací Raspberry Pi Foundation na míru Raspberry Pi. Disponuje množstvím nástrojů, které mimo jiné zjednodušují ovládání externích zařízení.



Obr. 1.1: Raspberry Pi model 3 B+

Raspberry Pi umožňuje ovládání i několika externích zařízení najednou. Limitem samotného zařízení je výpočetní síla a fyzické možnosti hardwaru. V teorii může tedy Raspberry Pi být řídicí jednotkou nejen pro chytrý elektroměr této diplomové práce, ale i pro celou chytrou domácnost nebo její část. Pokud je externí zařízení programovatelné ze vzdáleného bodu, je možné pro tyto účely využívat právě Raspberry Pi, které je velice kompaktní a cenově dostupné [3].

Původní verze praktického výstupu k realizaci elektroměru využívala výstupních pinů přímo na desce Raspberry Pi. Ta však měřila pouze stejnosměrný proud. Pro měření střídavého proudu je vhodné využití platformy Arduino, která disponuje vestavěným převodníkem. Nyní Raspberry Pi přijímá naměřené hodnoty a údaje o tarifu z platformy Arduino, a na zřízeném DLMS serveru v jazyce Java s nimi dále operuje (viz. 2.1).

## 1.3 Arduino

Arduino je open-source elektronická platforma založená na snadno použitelném hardwaru a softwaru. Desky Arduino jsou schopny číst vstupy připojených externích modulů a na tyto vstupy reagovat odpovídajícím výstupem. Vstupem může být například stisknutí tlačítka nebo spuštění senzoru světla. Výstupem může být aktivace zvukového modulu nebo spuštění LED kontrolky. Deska Arduino se řídí danou sadou instrukcí zaslanou mikrokontroléru na desce. K tomu je použit upravený programovací jazyk C++ a Arduino Software IDE. V případě této diplomové práce je to vstup hodnot k měření veličin obvodu elektrické sítě a výstupem je výpočet a následně odeslání těchto naměřených hodnot desce Raspberry Pi sériovým portem.

Kolem této platformy s otevřeným zdrojovým kódem se postupem času shromáždila celosvětová komunita tvůrců. Těmi jsou studenti, umělci, fanové, profesionálové a programátoři. Arduino je součástí tisíců projektů, od každodenních předmětů po složité vědecké přístroje. Arduino má mnoho využití a nové projekty pracující s touto platformou přibývají každým dnem.

Desky Arduino jsou relativně levné ve srovnání s jinými platformami mikrokontrolérů. Nejlevnější verzi modulu Arduino lze sestavit ručně a dokonce i předem sestavené moduly využívající Arduino se pohybují v nízkých cenových kategoriích.

Filozofie práce s deskou Arduino je taková, že se na ni mohou vkládat takzvané shieldy, ke kterým je možné zapojit konkrétní modul a ostatní kontakty desky prostupují dále pro použití dalších shieldů.

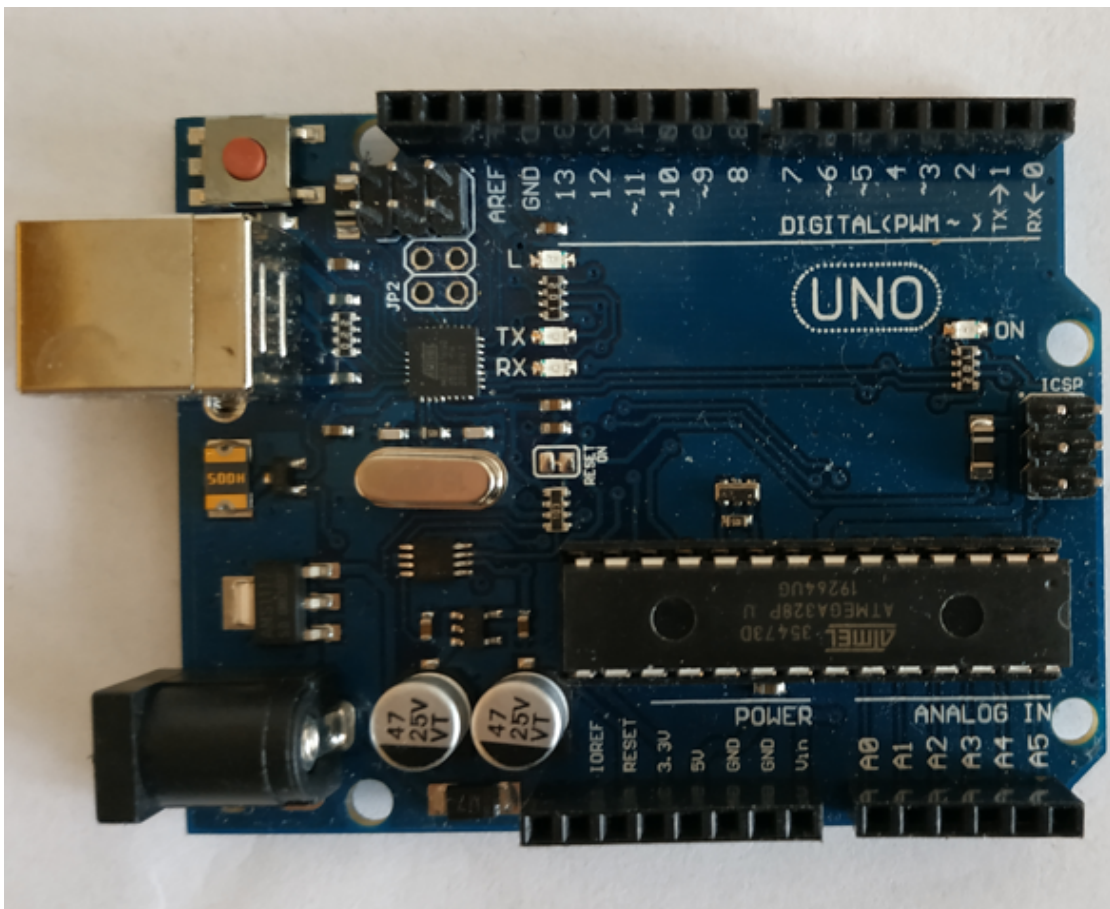
Na desce Arduino je v této práci nasazen shield, který zjednodušuje práci s kontakty. Samičí konektory z původní desky rozšiřuje a zakončuje je šroubovacími konektory (viz. obrázek 1.5). Kromě této změny je práce s deskou naprosto stejná, proto je v následujících schématech a obrázcích použita deska Arduino bez shieldu.

Arduino IDE běží na operačních systémech Windows, Macintosh OS X a Linux. Většina systémů mikrokontrolérů je však omezena na operační systém Windows.

Arduino vyvinul Institut Ivrea Interaction Design v roce 2005 jako snadno použitelný nástroj pro rychlé prototypování, který bere v potaz začátečníky a studenty bez znalosti elektroniky a programování. Jakmile se deska Arduino dostala do širší komunity, začala se měnit, aby se přizpůsobila novým potřebám a výzvám, a odlišila svou nabídku od jednoduchých 8-bitových desek po produkty pro aplikace internetu věcí (IoT).[4]

Platforma má tedy několik různých modelů. Tyto modely se liší různými parametry, jako jsou například počet digitálních pinů vstup/výstup, USB rozhraní nebo rozměr. Praktický výstup této diplomové práce využívá model UNO. Na desce Arduino v naší implementaci tedy neprobíhá žádná DLMS komunikace. Arduino pouze měří aktuální stav veličin elektrické sítě, formátuje je do dat v podobě, kterou umí

číst program na desce Raspberry Pi v jazyce Java, a posílá je sériovou linkou.



Obr. 1.2: Klon platformy Arduino UNO

## 1.4 Měření proudu

Jednou z funkcí chytrého elektroměru musí být přirozeně měření proudu. V praktickém výstupu práce navazující na tuto diplomovou práci bylo využito senzoru INA219. Ten dokázal měřit proud na principu bočníkového napětí. Nicméně tento senzor je použitelný pouze v případě stejnosměrného proudu. Tato práce má však za cíl vytvořit měřič schopný měřit údaje na síti běžné domácnosti, to znamená měření střídavého proudu.

Dnešní běžné domácnosti disponují třífázovým vedením. Elektroměr, který je výstupem této práce, však měří jen jednu fázi. Úprava na měření tří fází je však pouze triviálním úkonem. Stačí použít další měřicí moduly stejného typu pro každou fázi, a výsledné hodnoty následně sčítat, nebo pro ně vytvořit nový objekt v závislosti na veličině.

Pro účely praktického výstupu této diplomové práce byl využit neinvazivní proudový senzor. Jedná se o měřič, kterým se obklopí nulové nebo fázové vedení měřené sítě. Vzhledem k fungování senzoru na principu Hallova jevu by měření obou vedení zároveň vedlo k vzájemnému vyrušení a sonda by tedy vracela rozdíl vstupního a výstupního proudu. Pokud tedy není žádný únik proudu, vstup a výstup se rovnají a dojde ke vzájemnému vyrušení.[5]

Z této sondy je Arduino schopna vypočítat hodnotu proudu s využitím Hallova jevu.

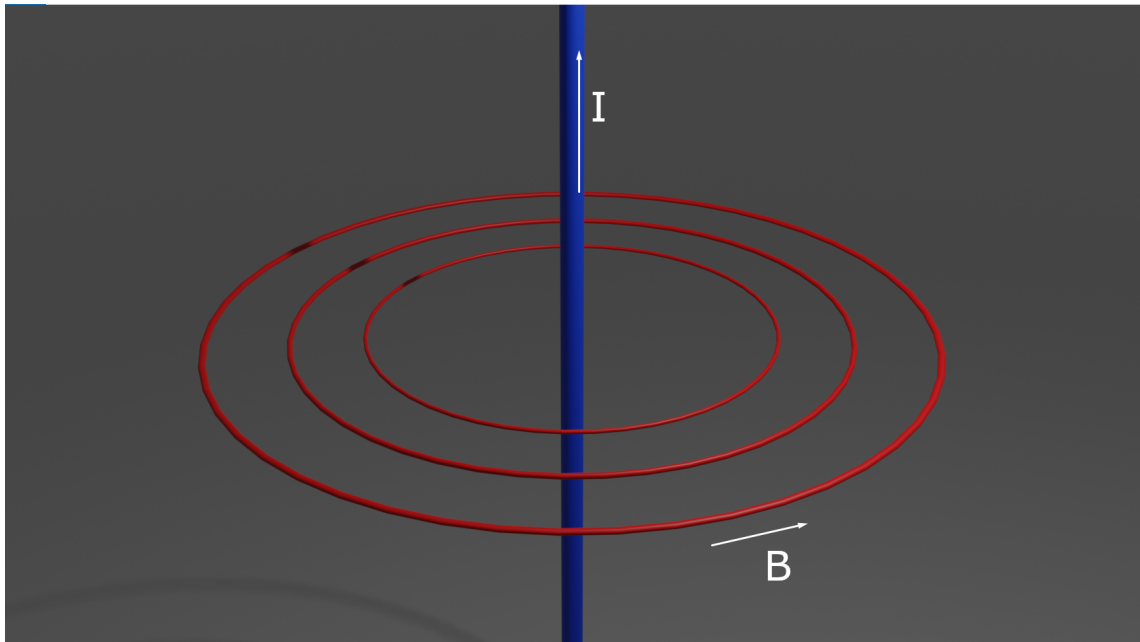


Obr. 1.3: Proudový senzor SCT-013-000

Vedení, kterým prochází elektrický proud, generuje ve svém okolí magnetické pole podle Ampérova pravidla pravé ruky.

Toto magnetické pole prochází feritovým jádrem použitého proudového senzoru. Kolem tohoto jádra se nachází sekundární cívka tvořena vodičem. Nyní se aplikuje stejný jev. Kolem magnetického pole se v cívce indukuje napětí. Tato cívka je na výstupech propojena skrze odpor konkrétní hodnoty. Deska Arduino je poté schopna





Obr. 1.4: Schéma Ampérova pravidla pravé ruky

měřit úbytek napětí na zmíněném odporu. Je naměřeno napětí a velikost použitého odporu je známá. S využitím upraveného vzorce Ohmova zákona,

$$I = \frac{U}{R}, \quad (1.1)$$

a dosazení známých hodnot je možné vypočítat proud procházející sekundární cívkou.

Nyní tedy známe proud, který prochází sekundární cívkou. Výrobce senzoru uvádí, že převod měřeného proudu na proud výstupní je 100 A na 50 mA. Pro zjištění skutečného proudu vodičem se tedy naměřená hodnota musí vynásobit dvěma tisíci.

Tím, že je měřen střídavý proud, je vyjádření velikosti proudu obtížnější než u proudu stejnosměrného. Ten se totiž v průběhu času neustále mění. Předpokládá se však jeho sinusový průběh. Pro tyto účely byly definovány hodnoty špičkový proud  $I_{\max}$  a efektivní proud  $I_{\text{ef}}$ . [6]

Špičkový proud  $I_{\max}$  je nejvyšší hodnota, které proud v daném časovém úseku dosahuje a efektivní proud  $I_{\text{ef}}$  je hodnota stejnosměrného proudu, který by při průchodu odporem generoval stejnou tepelnou energii jako měřený proud střídavý. Pro tyto dvě hodnoty platí vztah

$$I_{\text{ef}} = 1/\sqrt{2} * I_{\max}. \quad (1.2)$$

Při běžných výpočtech se však využívá vzorce

$$I_{\text{ef}} \approx 0,707 * I_{\text{max}}. \quad (1.3)$$

Při skutečném měření v praxi je teda třeba zjistit hodnotu špičkového proudu  $I_{\text{max}}$  a efektivní proud  $I_{\text{ef}}$  dopočítat s použitím konstanty zmíněné ve vzorci 1.5. Softwarové řešení je rozebráno v podkapitole 2.2.

## 1.5 Měření napětí a frekvence sítě

Pro měření hodnot napětí a frekvence v síti střídavého proudu byl v praktickém výstupu použit modul snímače střídavého napětí ZMPT101B. Deska Arduino je navržena pro napětí do velikosti 5 V. Vyšší napětí by mohlo vést k jejímu selhání. Proto je nutné pro práci s vysokým napětím využívat modulů jako je právě modul ZMPT101B.

Tab. 1.1: Hlavní technické parametry transformátoru ZMPT101B [7]

Model	ZMPT101B
Jmenovitý vstupní proud	2 mA
Jmenovitý výstupní proud	2 mA
Poměr otáček	1000:1000
Chyba fázového úhlu	$\leq 20$ minut (vstup 2 mA, vzorkovací odpor 100 $\Omega$ )
Provozní rozsah	0–1000 V, 0–10 mA (vzorkovací odpor 100 $\Omega$ )
Přípustná chybovost	$-0,3\% \leq f \leq 0,2\%$ (vstup 2 mA, vzorkovací odpor 100 $\Omega$ )
Izolační napětí	4000 V
Aplikace	Měření napětí a výkonu
Materiál zapouzdření	Epoxid
Instalace	PCB montáž (délka pinů > 3 mm)
Operační teplota	-40°C–60°C

Tento modul využívá snížovacího transformátoru ke snížení vstupního napětí. Toto napětí změří a odešle jeho analogovou hodnotu výstupním pinem. Deska Arduino dokáže data přijata analogovým pinem přechít a následně s nimi pracovat.[8]

Podobně jako je tomu u velikosti střídavého proudu, se dá k vyjádření hodnoty střídavého napětí použít jeho efektivní hodnota  $U_{\text{ef}}$ . I zde se dá využít hodnoty maximálního napětí  $U_{\text{max}}$  a stejné konstanty pro její určení.

$$U_{ef} \approx 0,707 * U_{max}. \quad (1.4)$$

Data přijatá z modulu ZMPT101B však slouží i k určení frekvence sítě a způsob zjištění efektivní hodnoty se tedy od výpočtu efektivního proudu  $I_{ef}$  liší.

Data jsou kontinuálně sbírána po určitou dobu a umocňována na druhou. Následně se určí průměr těchto hodnot. Tento průměr je odmocněn a výsledná hodnota odpovídá efektivní hodnotě napětí  $U_{ef}$ .

Měření frekvence  $f$  sítě probíhá na základě výstupu stejného modulu a proto je paralelní s měřením napětí. Frekvencí se zde rozumí počet sinusových period proudu nebo napětí za sekundu v měřené síti.

Zjištění této hodnoty spočívá v měření sítě po určitou dobu. V průběhu měření je zaznamenáno, kolikrát průběh přešel přes nulu ze záporných do kladných hodnot. Tento přechod značí průběh periody. Po určitém časovém úseku známe počet period a čas, který tento počet period trval. Nyní stačí použít vzorec

$$f = \frac{1}{T}, \quad (1.5)$$

kde  $T$  je počet period za jednu sekundu.

Bližší popis softwarového řešení pro měření napětí a frekvence sítě se nachází v podkapitole 2.2.

## 1.6 DLMS/COSEM

DLMS (Device Language Message Specification), je sada standardů vytvářených a udržovaných asociací DLMS User Association (DLMS UA) pro výměnu dat mezi měřicími zařízeními. Úkolem DLMS/COSEM je plnit především níže uvedené funkce:

- **Modelování**

Ukládání dat do struktur, objektů a určování pravidel pro přístup k nim na základě unikátních identifikátorů a umožňuje pohled na funkce, které měřicí zařízení podporuje.

- **Signalizace**

Komunikační služby a protokoly, které mapují jednotlivé prvky datového modelu na prvky aplikačního protokolu.

- **Transport**

Zahrnuje veškeré služby a protokoly, které se zúčastňují na přenosu zpráv mezi jednotlivými uzly.

DLMS/COSEM je založen na komunikačním modelu typu klient-server. Koncovými zařízeními jsou měřiče. Chytré měřiče mohou fungovat jako server i jako klient. Pokud jsou nakonfigurovány jako server, je se znalostí jejich veřejné adresy jednoduchá jejich konfigurace. Proto je většina chytrých elektroměrů nakonfigurována právě takto. DLMS User Association definuje protokoly do čtyř specifikačních dokumentů, takzvaných barevných příruček (*Colored books*). Konkrétně to jsou zelená příručka, žlutá příručka, modrá příručka a bílá příručka. Jejich podrobnější popis je níže. Tyto příručky jsou využity pro samotnou implementaci DLMS/COSEM, a jejich správu má na starost právě DLMS User Association.

- **Modrá Příručka**

Popisuje objektově orientovaný model měřiče COSEM a identifikační systém OBIS<sup>1</sup>.

- **Zelená Příručka**

Specifikuje architekturu a využívané protokoly DLMS/COSEM.

- **Žlutá Příručka**

Řeší všechny otázky týkající se testování.

Na modelu ISO/OSI působí DLMS/COSEM na třech vrstvách. Aplikační, relační a transportní. COSEM objekty prezentují data ve speciálních strukturách a objektech a patří tedy na prezentační vrstvu. DLMS aplikační vrstvy popisují a specifikují služby přístupující k COSEM objektům. DLMS operuje zase na vrstvě relační a transportní, protože způsob, jakým se tyto služby přenáší přes různá média, určují komunikační protokoly. Způsob přenosu dat na nižších vrstvách není v DLMS/COSEM nijak specifikován. DLMS/COSEM tedy není závislý na přenosovém médiu.

Aby se mohlo zařízení zúčastnit DLMS/COSEM komunikace, musí v jejím rámci nést svoji unikátní adresu jako svůj identifikátor v DLMS označovaný jako *Sys-T*. Tento identifikátor musí pro správné fungování zůstat statický. Adresa se skládá z osmi oktetů, kde první tři slouží jako identifikátor výrobce<sup>2</sup> a zbytek určuje výrobce sám za účelem zachování unikátních adres jednotlivých zařízení. Způsob, jakým tyto adresy výrobce přiděluje, je čistě na výrobcu. Důležitá je pouze unikátnost [11].

### 1.6.1 Služby DLMS

Služby poskytované DLMS se dají dělit do dvou hlavních skupin. První skupinou jsou takzvané asociační služby. Ty zajišťují navázání a ukončení spojení. Druhou

---

<sup>1</sup>OBIS identifikátor je šestice čísel. Každé číslo má svůj význam v závislosti na třídě. Tyto třídy jsou A-F a identifikují měřicí zařízení na základě vlastnosti zařízení. Tedy co vlastně měří a bližší informace k tomuto měření.

<sup>2</sup>Tento identifikátor výrobcu uděluje přímo DLMS User Association.

skupinu tvoří služby pro datové přenosy. Jejich účelem je samotný přenos dat mezi zařízeními. Slouží tedy ke čtení a zapisování na jednotlivých měřících zařízeních.

- **Asociační služby**

- **Navázání spojení**

- Takzvané AA spojení. Pro připojení na cílové zařízení se používá požadavek A-Associate Request. Cílové zařízení vrací odpověď A-Associate Response.

- **Ukončení spojení**

- AA ukončení spojení. Pokud je potřeba zrušit spojení, je na cílové zařízení odeslán požadavek A-Release Request. To vrací odpověď A-Release Response

- **Služby datového přenosu**

Dělí se na LongName a Shortname Služby. LongName a ShortName reference slouží jako adresa jednotlivých COSEM objektů a tyto služby umožňují přístup k jednotlivým atributům. LongName přistupují k celému objektu a Shortname ke konkrétním atributům objektu.

- **ShortName služby**

- \* **Read**

- Čtení atributů COSEM rozhraní. Dotazy readRequest a odpovědi readResponse.

- \* **Write**

- Zápis atributů COSEM rozhraní. Dotazy writeRequest a odpovědi writeResponse.

- \* **UnconfirmedWrite**

- Téměř shodná se službou Write. Rozdíl je ten, že zápis nemusí být potvrzen cílovým zařízením.

- **LongName služby**

- \* **GET**

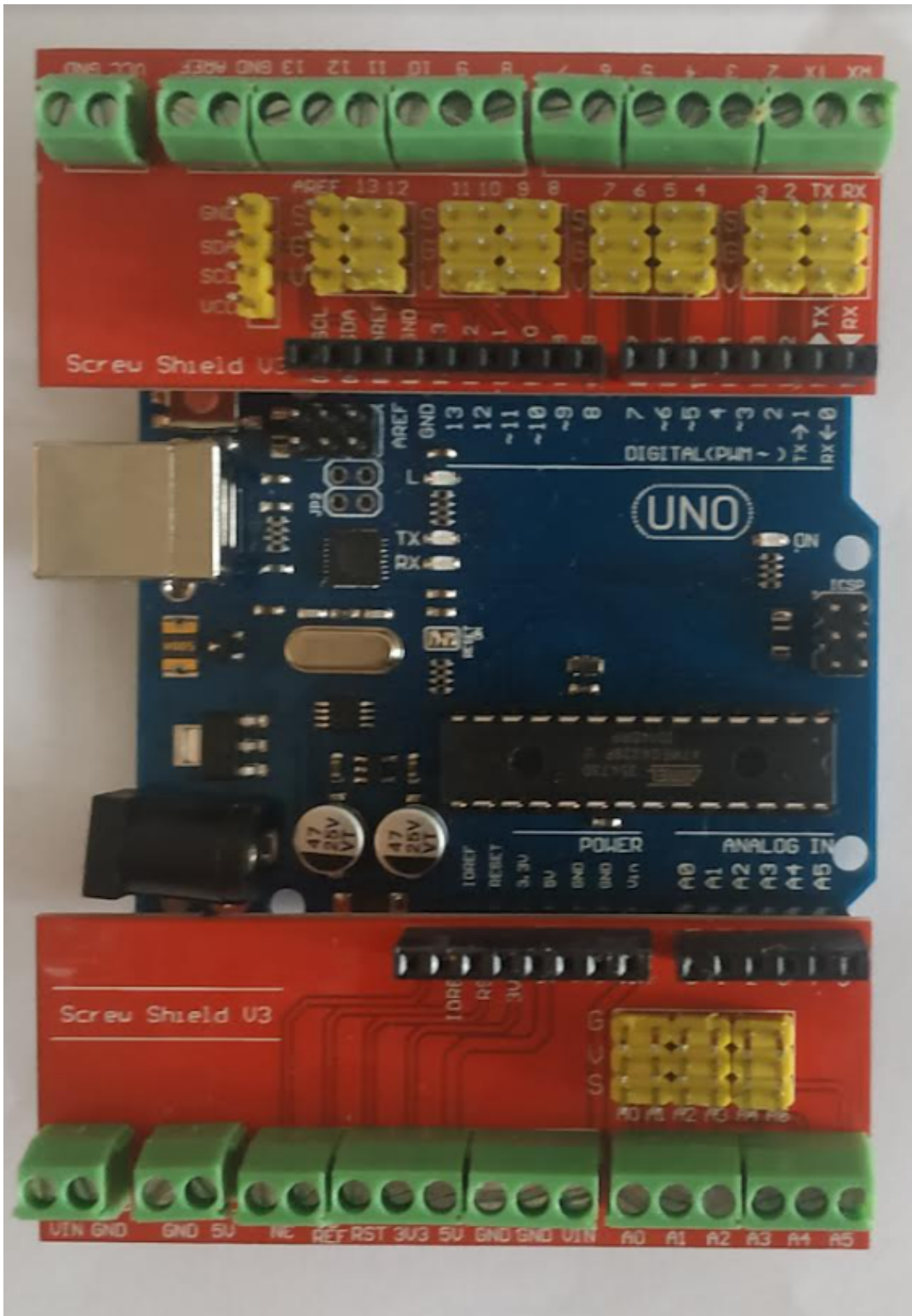
- Čtení atributů COSEM rozhraní. Dotazy get-request a odpovědi get-response.

- \* **SET**

- Zápis atributů COSEM rozhraní. Dotazy set-request a odpovědi set-response.

- \* **Action**

- Volání metod z COSEM rozhraní. Dotazy action-request a odpovědi action-response.



Obr. 1.5: Klon platformy Arduino UNO s pridaným shieldem pro šroubovací konektory

## 2 Praktická část

V této části diplomové práce je popsáno programové řešení a hardware použitý v praktickém výstupu. Následující podkapitoly obsahují popis komponentů, které byly využity pro fyzické zapojení, schéma samotného zapojení a shrnutí programového řešení výstupu práce.

### 2.1 Zapojení

V původní verzi chytrého elektroměru nebyla využita deska Arduino. Zapojení probíhalo přímo přes piny desky Raspberry Pi a výpočty probíhaly také zde. Pro měření střídavého proudu je však vhodné využít právě zmíněné Arduino. Nejenže disponuje vestavěným převaděčem střídavého proudu, ale také se jedná o mnohem levnější položku. Při nesprávném připojení k vysokému napětí by se tedy musela nahradit pouze deska Arduino, případně některý z modulů.

Nyní Raspberry Pi slouží jako napájecí zdroj pro desku Arduino a přijímá od ní přímo konkrétní naměřené hodnoty sériovým portem. Ty ukládá do databáze a přepočítává pro odeslání protokolem DLMS/COSEM. Bližší popis funkcionality se nachází v podkapitole 2.2.

Veškeré měření sítě tedy probíhá na straně desky Arduino. Všechny moduly a senzory potřebné k měření hodnot v síti jsou připojeny k desce Arduino podle schématu na obrázku 2.1.

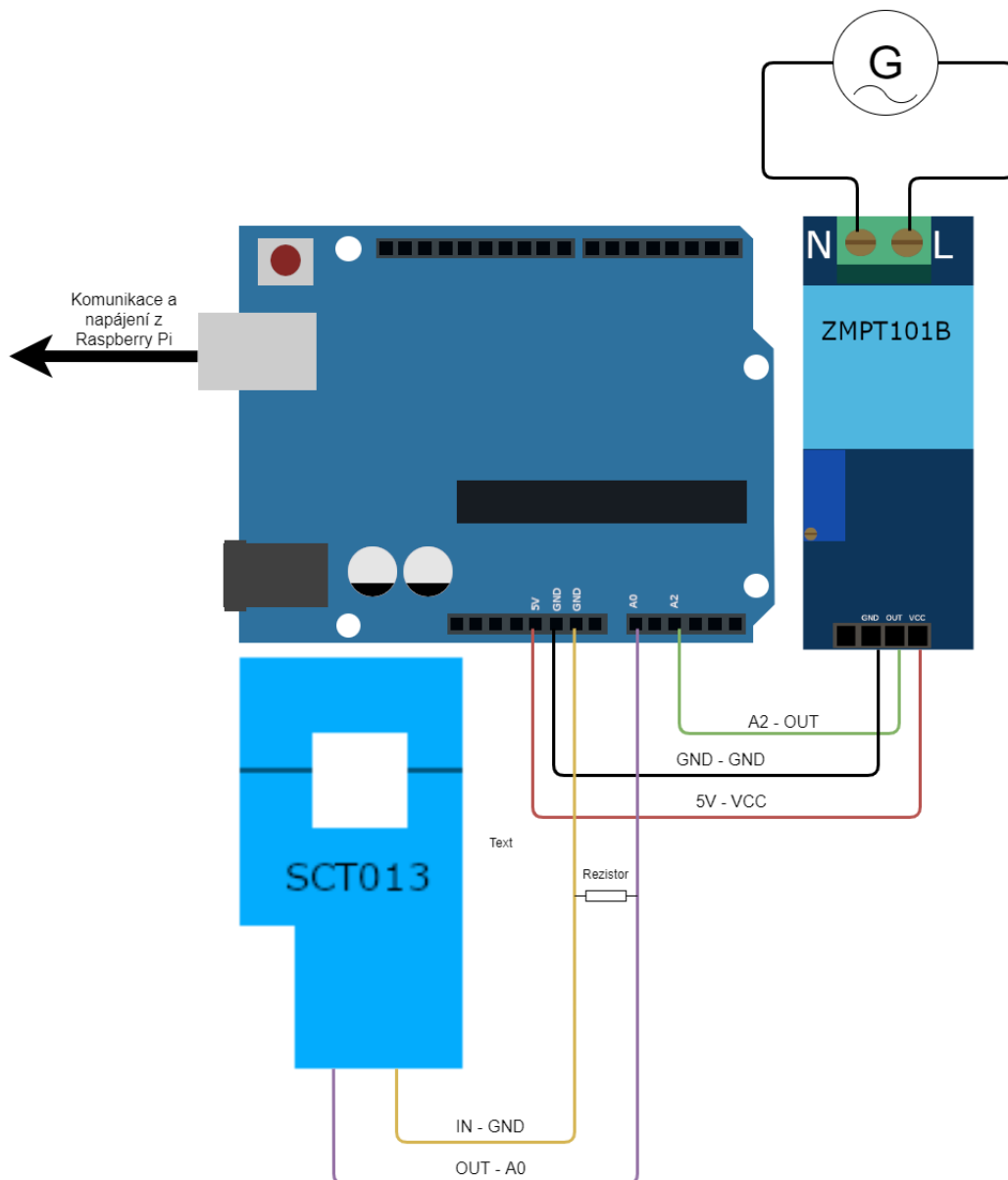
#### 2.1.1 Proudový senzor

Jak již bylo řečeno v podkapitole 1.4, sonda pro měření proudu se umístí na fázový nebo nulový vodič a na výstupech senzoru se začne generovat napětí podle proudu procházejícího sondou. Pokud jsou tyto výstupy propojeny rezistorem známé velikosti, je možné změřením napětí na tomto rezistoru zjistit proud generovaný senzorem. Deska Arduino dokáže změřit právě napětí mezi uzemňovacím pinem a pinem analogového vstupu. Pokud tedy zapojíme rezistor a senzor po měření proudu tak, jak je znázorněno na obrázku 2.1, je možné hodnotu napětí změřit a vhodným programovým řešením vypočítat proud procházející měřicí sondou. Programové řešení měření proudu je blíže popsáno v podkapitole 2.2.

#### 2.1.2 Senzor napětí

Senzor napětí je na rozdíl od senzoru proudu modulem přímo určeným k použití na desce Arduino. Pro svou funkci potřebuje připojit ke zdroji 5 V, sdílet uzemnění s

Arduinem a také propojit výstupní kontakt ZMPT101B se vstupním pinem desky Arduino. Zapojení senzoru ZMPT101B k desce Arduino je znázorněno na obrázku 2.1.



Obr. 2.1: Schéma zapojení senzorů k platformě Arduino



## 2.2 Programové řešení

K realizaci chytrého elektroměru bylo třeba nejen jednotlivé komponenty správně zapojit, ale také naprogramovat desky Arduino a Raspberry Pi tak, aby se vstupními daty vhodně zacházely a aby následný výstup ukazoval přesné údaje. Programové řešení na straně Arduina spočívalo v přijímání dat z měřících modulů, výpočet hodnot konkrétních veličin a odesílání těchto hodnot sériovým portem desce Raspberry Pi. Raspberry Pi muselo přijímat hodnoty z platformy Arduino sériovým portem, tyto hodnoty ukládat na vlastním úložišti, vkládat je do DLMS objektů a být schopné hodnoty odesílat prostřednictvím protokolu DLMS/COSEM. Programování na desce Arduino je v upraveném jazyce C++ a na desce Raspberry Pi v jazyce Java. Softwarové řešení se skládá z dílčích řešení. Zaprvé bylo nutné naprogramovat měření samotné. Poté musí Raspberry Pi nějakým způsobem měřená data přijímat a ukládat. Dalším úkolem bylo přiřazovat datům hodnotu dle tarifu spotřeby energie a nakonec měřené hodnoty posílat protokolem DLMS/COSEM. Následující kapitoly popisují řešení jednotlivých problémů.

### 2.2.1 Použité knihovny

Praktický výstup diplomové práce využívá knihoven pro zřízení DLMS komunikace společnosti Gurux. Zřízení a implementace veškeré komunikace DLMS je nad časový rámec diplomové práce. Kód, který poskytuje společnost Gurux na svém Githubu, je open source a jde implementovat právě na platformě Arduino v jazyce C++, nebo v jazyce Java. Výsledný elektroměr využívá knihovny pro zřízení serveru na platformě Raspberry Pi v jazyce Java.[16]

V průběhu vývoje elektroměru byl zřízen server přímo na zařízení Arduino, ale u tohoto řešení se data posílala pouze po sériové lince, a komunikace v lokální síti pomocí DLMS byla značně komplikovaná. Nyní je tedy server zřízen až na Raspberry Pi, které je schopné odesílat například prostřednictvím TCP protokolu.

### 2.2.2 Funkce pro měření proudu a spotřeby

Měření proudu probíhá na desce Arduino. Metoda pro toto měření je tedy vytvořena v jazyce C++ uzpůsobenému pro tuto platformu. Pro zjištění hodnoty proudu je nutné z výstupního napětí ze senzoru zjistit úbytek přes odpor. Toto napětí má sinusový průběh a je tedy nutné vypočítat jeho efektivní hodnotu. Efektivní hodnota proudu  $I_{ef}$  je v této metodě řešena přes výpočet z hodnoty maximálního proudu  $I_{max}$  pomocí vzorce z podkapitoly 1.4.

Jak je řečeno v podkapitole 1.4, vstup ze senzoru proudu je napětí generované na jeho sekundární cívice. Toto napětí prochází odporem známé hodnoty. Tímto způsobem jsou známy všechny hodnoty potřebné pro výpočet. Pro správnou funkci byly definovány proměnné pro hodnotu odporu, která byla změřena na kleštovém ampérmetru DMM266. Konkrétně jde o odpor  $988 \Omega$ . Dále bylo nutné zaznamenávat právě měřenou hodnotu, kvůli výpočtu efektivního proudu podle proudu maximálního také proměnné pro měření časového intervalu a uložení nejvyšší naměřené hodnoty. Proces zjištění hodnoty proudu tedy probíhá tak, že po určitou dobu deska Arduino měří výstup z proudového senzoru a uloží nejvyšší změřenou hodnotu jako maximální proud  $I_{\max}$ . Z něj vypočítá hodnotu efektivního proudu  $I_{\text{ef}}$ . Protože je známa jak hodnota proudu  $I$ , tak hodnota napětí  $U$ , je možné vypočítat také okamžitý příkon v síti  $P$ . K tomu je využit vzorec

$$P = UI. \quad (2.1)$$

Tedy, když známe příkon sítě a dobu, za kterou byl tento příkon změřen, je možné využít vzorce pro výpočet spotřeby elektrické energie

$$E = Pt, \quad (2.2)$$

kde  $E$  je spotřeba elektrické energie ve Watthodinách,  $P$  je příkon sítě ve Watech a  $t$  je čas, po který byl tento příkon měřen v hodinách.

Hodnota celkové spotřeby je jediná, která je vypočítávána až na straně Raspberry Pi. Při jejím výpočtu se totiž odesílání jednotlivých záznamů, které bez jejího výpočtu trvalo právě jednu sekundu (průměrně 1002 milisekund), zdrželo právě tímto výpočtem. Pro zjednodušení výpočtů je tedy ze strany Arduina místo hodnoty spotřeby odesílán čas, za který byl odesílaný záznam změřen v milisekundách a na Raspberry Pi byla vytvořena funkce, která pomocí tohoto času a ostatních hodnot ze záznamu, dopočítá hodnotu spotřeby.

Tyto hodnoty jsou vloženy do DLMS objektů ve formátu, který podporuje knihovna společnosti Gurux pro zřízení DLMS serveru na desce Raspberry Pi a ty pak může klientská strana, jako je aplikace DLMS Director společnosti Gurux číst. Schéma měření hodnot proudu, příkonu a spotřeby je znázorněno v obrázku 2.2.

### 2.2.3 Funkce pro měření napětí a frekvence

Jak bylo řečeno v kapitole 1.5, modul ZMPT101B posílá analogovou hodnotu napětí měřeného v síti. Hodnota tohoto napětí má stejně jako hodnota proudu sinusový průběh. Tím, že u měření proudu deska Arduino zaznamenává kladné hodnoty napětí, je přepočítání s využitím konstanty ze vzorce z podkapitoly 1.4 jednoduché. Při

měření napětí však deska Arduino přijímá číselnou hodnotu. Tento vstup ve výchozím stavu osciluje sinusovým průběhem kolem hodnoty 512. To však neznamená, že skutečné napětí sítě osciluje kolem hodnoty 512 V. Tato hodnota je přijímána ze zařízení ZMPT101B jako nulová a jedná se o offset daný výrobcem. Pro zjednodušení výpočtů jsou přijímané hodnoty programově sníženy právě o 512 V.

Teoreticky je možné převádět vstup na jeho absolutní hodnotu, uložit maximální napětí  $U_{\max}$ , a pak využít stejnou konstantu pro výpočet efektivního napětí  $U_{\text{ef}}$ . Efektivní hodnota se často označuje indexem RMS. To kvůli anglickému označení „Root Mean Square“, tedy střední kvadratické hodnotě. Měřiče v praxi využívají průměrující matematické vzorce k přesnému měření čistých sinusových průběhů. Je bezpečné předpokládat že střídavé napětí v elektrické síti na území České republiky má právě sinusový průběh a je možné tyto vzorce využít v řešení této diplomové práce.

V praxi tedy měření spočívá v tom, že deska Arduino, stejně jako je tomu u měření proudu, přijímá hodnoty v časových intervalech. Pro zjištění střední kvadratické hodnoty je každý záznam umocněn na druhou a přičten k celkové sumě záznamů právě probíhajícího měření. Z této sumy se poté vypočte kvadratický průměr a jeho druhá odmocnina je hledané efektivní napětí  $U_{\text{ef}}$ . Tento výsledek je převeden do formátu, který dokáže zpracovat knihovna pro DLMS komunikaci a je poslán jako součást DLMS dat sériovou linkou.

V průběhu měření napětí probíhá také výpočet frekvence v síti. Výstup senzoru osciluje sinusově podél nulové hodnoty. Frekvence označuje počet těchto period za sekundu. Před umocněním měřené hodnoty a jejím přičtením k celkové sumě záznamů pro výpočet střední kvadratické hodnoty se ověřuje, zda měřená hodnota překročila nulu do záporné hodnoty. Pokud se tak stane, znamená to, že právě proběhla perioda a navýší se proměnná určená pro jejich počítání. Po skončení časového úseku měření, v tomto případě jedna sekunda, se proměnná počítající periody rovná frekvenci v měřené elektrické síti.

## 2.2.4 Implementace protokolu DLMS/COSEM

Pro realizaci chytrého elektroměru bylo třeba zprovoznit serverovou stranu protokolu DLMS/COSEM. Server je implementován na desce Raspberry Pi pomocí zmíněných knihoven společnosti Gurux [16]. Kód serveru bylo nutné upravit tak, aby byla data přijatá sériovou linkou přečtena na straně serveru. Dále musí server tato data vhodně rozdělena podle veličiny, kterou reprezentují, vložit do objektů. K tomu bylo třeba definovat příslušné proměnné, ale hlavně jednotlivým veličinám určit správné OBIS kódy. OBIS kód je číselný kód, který představuje měřenou fyzikální veličinu. Samotné odesílání hodnot provádí. Tato DLMS data poté serverová aplikace na straně

Raspberry Pi posílá přes protokol TCP každé připojené klientské straně, která si o data zažádá. Protokol DLMS/COSEM není závislý na přenosovém médiu, takže implementovat přenos jiným způsobem je otázkou mírných úprav. Open source kód od společnosti Gurux však používá právě tento přenos, a tak je tomu uzpůsoben i náš elektroměr. Deska Raspberry Pi je při komunikaci spojena s jakoukoliv klientskou stranou pomocí TCP spojení. Tato implementace tedy podporuje komunikaci s jakoukoliv klientskou stranou, která je schopna toto spojení navázat.

Pro ověření funkčnosti serveru a správnosti dat byla použita aplikace GXDLMS Director. Také byla implementována klientská aplikace a skript v jazyce Python, který z jejího výpisu vyjme hodnoty měření, a ukládá je do databáze ve vlastním úložišti.

### 2.2.5 Měření spotřeby na základě tarifů

Jedním z cílů bylo zařídit podporu měření spotřeby elektřiny ve dvou tarifech. To znamená, že elektroměr by měl být schopen měřit dvě různé hodnoty spotřeby v závislosti na čase měření. Deska Arduino však není schopná bez přídatných modulů určit čas, a to komplikuje určování, kdy přičítat hodnotu spotřeby do objektů jednotlivých tarifů. Pro tento problém bylo navrženo několik řešení.

V první verzi elektroměru byl server zřízen už na straně Arduina a Raspberry Pi sloužilo jako vzorová klientská strana. Arduino však není schopno určovat čas a pokyny pro přepínání tarifu by muselo získávat softwarově z Raspberry Pi buď přes sériovou linku nebo výstupní piny. Další možností bylo připojit přepínač pouze na Arduino a to naprogramovat tak, aby podle stavu přepínače měřilo buď vysoký nebo, nízký tarif.

Po přechodu k verzi elektroměru se serverem na straně Raspberry Pi se přešlo na čistě softwarovou implementaci přepínání tarifů spotřeby. Pro každý tarif byl vytvořen samostatný objekt a spotřeba se přičítala podle aktuálního času systému. Tímto způsobem elektroměr nevyžaduje lidský vstup a je plně automatizován. V praxi však elektroměry samy neurčují jaký je zrovna tarif. To určuje dodavatel energie. Naprogramovat tedy tyto časy přímo na straně elektroměru neodpovídá řešení v praxi používaných elektroměrů.

Pro simulaci reálné situace je přepínání tarifů opět řešeno reálným přepínačem na desce Arduino. Arduino poté posílá informaci o tom, pod jakým tarifem byl konkrétní záznam naměřen, a server na základě toho přičítá hodnotu spotřeby k odpovídajícímu objektu. Toto přepínání tarifu nyní tedy musí provádět uživatel, ale nejvíce se blíží řešení v praxi používaných elektroměrů, které jsou řízeny sice automaticky, ale velice podobným přístupem.

Teoreticky je možné použít výstup z Raspberry Pi, které by bylo naprogramováno tak, aby odesílalo napětí v závislosti na čase a proces přepínání tarifů by byl automatizován. Nicméně v praxi uživatel sám neurčuje, kdy začíná nízký nebo vysoký tarif. Po elektroměry v praxi se k tomuto využívá takzvané hromadné dálkové ovládání (HDO). Jedná se o soubor prostředků, které mohou vysílat signály pro zapínání či vypínání konkrétních spotřebičů, ale hlavně přepínání tarifů. Tyto tarify nemá pod kontrolou uživatel, ale přímo dodavatel energie. Proto je kódování přesných časů nepraktické řešení. Dále je teoreticky možné tento spínač ovládat vstupem z HDO dodavatele energie.

## 2.2.6 Ukládání dat

Ukládání dat probíhá dvěma způsoby Ukládání probíhá jak na straně serveru, tak na straně vzorového klienta.

Strana serveru má velice jednoduchou realizaci ukládání naměřených dat. Vzhledem k tomu, že server by měl ukazovat pouze aktuální hodnoty a hodnoty, které je nutné si pamatovat jsou spotřeby obou měřených tarifů, byla vytvořena funkce pro ukládání jejich posledních hodnot do textového souboru. Tyto textové soubory jsou uloženy na zařízení serveru pro případ, kdy by bylo zařízení Raspberry Pi vypnuto a hodnoty spotřeby by tím byly z objektu ztraceny. V takovém případě je možné tyto hodnoty vložit do objektu před inicializací programu a hodnota spotřeby zůstane uchována.

Ukládání naměřených hodnot z DLMS dat je na klientské straně Raspberry Pi uskutečněno přes databázi MariaDB. Na platformě Raspberry Pi je spuštěný bash skript, který čte výstup stavu elektroměru klientské aplikace. Bash skript je řada příkazů zapsaných v souboru, které jsou postupně čteny a prováděny programem. Program se provádí řádek po řádku. Je možné například spustit další skript, přečíst jeho výpis, vyčíst požadované hodnoty a ty ukládat. V případě naší implementace bash skript spustí klientskou aplikaci, uloží její výpis do souboru, a z něj poté podle OBIS kódů vyjme hodnoty měření. Tyto hodnoty ukládá do databáze MariaDB pro další využití. Každý záznam v databázi je definován identifikátorem, časovým razítkem, veličinou která je měřena a jeho hodnotou. V případě měření spotřeby jsou vytvořeny dvě označení veličiny pro vysoký a nízký tarif. K této databázi má poté přístup webový server.

## 2.2.7 Zobrazování dat

Na Raspberry Pi je zřízen NGINX server. Ten slouží k zobrazování nejaktuálnějších měřených hodnot ve webovém prohlížeči. Server je nastaven tak, že vstupuje do databáze záznamu měření a vyjme z ní aktuální hodnoty. Tyto hodnoty jsou poté

zobrazeny na stránce prohlížeče. Uživatel má tedy přístup k informacím týkající se měřené sítě. Konkrétně se jedná o hodnoty celkového proudu v síti, frekvence, napětí, aktuálního výkonu a spotřeby pod jednotlivými tarify. Tím, že data jsou ukládána v databázi, je v teorii možné zobrazovat i data za určitá období nebo z konkrétního času. Výhoda tohoto je možnost sledování stavu sítě v závislosti na čase.

Toto řešení je však použito jen při volání spuštění bash skriptu, který pošle klientskou aplikaci, čte z ní a data do databáze ukládá. Je tedy plně funkční pro klientskou stranu.

Pro stranu elektroměru samotného však toto rozšíření nemá využití. Jeho účelem je odesílat DLMS data pro klientskou stranu čitelná. Toto zobrazování je tedy vytvořeno jako vzorový klientský uzel uskutečněný také na platformě Raspberry Pi.

## 2.3 Výsledky

V této podkapitole je shrnuto řešení praktického výstupu práce a ukázka funkcionality. Je zde krok po kroku ukázán průběh procesu měření, práce s daty, a výsledné komunikace a zobrazení hodnot.

V průběhu vývoje, kdy byl DLMS server zřízen na desce Arduino, komunikace DLMS probíhala pouze interně a hodnoty měření byly dostupné na webovém serveru. Schéma tohoto provedení je na obrázku 2.4. Bylo však žádoucí uskutečnit DLMS komunikaci i mimo toto zařízení. Dále je tedy popsána verze elektroměru, která podporuje DLMS komunikaci přes protokol TCP.

Kromě úprav kódu z Github repozitáře společnosti Gurux byly vytvořeny metody pro měření jednotlivých veličin. Toto měření je prvním krokem celého procesu. Jako první krok byly tedy ve vývojovém prostředí Arduino IDE vytvořeny funkce pro měření hodnot vybraných veličin. Na obrázku 2.5 je zobrazen výstup jednotlivých metod po sériové lince ve vývojovém prostředí Arduino IDE. Pro všechna ukázková měření byl měřen obvod se zapnutou žárovkou o výkonu 30 W.

Po implementaci DLMS serveru na desce Raspberry Pi bylo ke čtení dat nutné naprogramovat vkládání nebo výpočet těchto hodnot do DLMS objektů. K tomu byl zjednodušen kód na Arduinu tak, aby data psal ve formátu

F:U:I:P:Tarif:t,

Kde F je hodnota frekvence sítě v Hertzích, U napětí ve Voltech, I proud v miliampérech a P hodnota příkonu sítě ve Wattech. Na místě tarifu je vypsána buď hodnota 0 nebo 1 a funkce Java na základě této informace přiřítá k příslušné hodnotě spotřeby. Na obrázku 2.6 je zobrazen tento zjednodušený výpis.

Na desku Raspberry Pi tedy přichází pravidelně řádky s formátovanými informacemi měření.

Do aplikace serveru je přidána funkce, která tento vstup rozdělí podle symbolů dvojtečky a do pole o velikosti 6 přiřadí hodnoty do příslušných objektů.

Problém je ten, že tato data přes sériovou linku přichází bez jakékoli synchronizace. To znamená, že při spuštění funkce pro čtení sériové linky už je pravděpodobně deska Arduino v průběhu výpisu a na straně serverové aplikace je přijatý necelý řádek. Při rozdělování hodnot do objektů poté nemusí odpovídat indexace, nebo se může zhroutit celý program.

To bylo vyřešeno tím, že pokaždé když funkce pro zápis přijme výpis z desky Arduino, a rozdělí jej do pole, proběhne kontrola velikosti tohoto pole. Pokud je velikost nižší než 6, tento výpis ignoruje, a čeká na další. Stále se může stát, že touto kontrolou projde vadná hodnota. Může totiž nastat situace, kdy je výpis zaznamenán v průběhu první hodnoty. Pokud je tedy naměřena frekvence 50 Hz, ale Raspberry

Pi tento zápis zaznamená až u symbolu 0, aplikace usoudí podle velikosti pole, že hodnoty jsou validní, a zapíše do objektu frekvence hodnotu 0. Nicméně tento problém již není relevantní. Vzhledem k funkčnosti je frekvence jedinou veličinou, která může být takto chybně zapsána a tato chyba je opravena hned v následujícím výpisu, který přijde o sekundu později. Hodnoty spotřeby v obou tarifech toto nijakým způsobem neovlivňuje a to jsou jediné hodnoty, které je potřeba si ukládat a pamatovat dlouhodobě.

Aplikace z repositáře Gurux již implementuje objekt příkonu. Byly tedy vytvořeny objekty ostatních měřených hodnot a funkce pro výpočty a zápis hodnot do těchto objektů.

Po uložení hodnot spotřeby pro oba tarify aplikace serveru odesílá DLMS data měření klientské straně, která si o tato data zažádá. Společnost Gurux přímo poskytuje aplikaci GXDLMS Director, která umí fungovat jako klient a číst DLMS data z měřičů. Tato aplikace byla využita k testování správnosti výstupů našeho elektroměru. Do kódu serveru byly vloženy metody pro měření potřebných veličin a vytvořeny objekty pro držení jejich hodnot. Při připojení elektroměru k aplikaci GXDLMS Directoru přes TCP spojení k příslušnému vstupnímu portu byla ověřeno, že měřené hodnoty v DLMS datech byly odpovídající. Tato skutečnost je znázorněna obrázkem 2.7 čtení frekvence aplikací GXDLMS Director z našeho elektroměru[17].

Pro sběr a zobrazení dat byla na platformě Raspberry Pi navíc zřízená klientská strana protokolu DLMS, databáze a webový server. Aby bylo v teorii možné odchylovat data z více zařízení v síti, byl elektroměr nastaven pro pravidelné odesílání DLMS dat. Elektroměr tedy jednou za daný časový interval odešle takzvanou Push message. Jedná se o zprávu, kterou si klientská strana nemusí vyžádat a stačí, aby ji pouze odchytila a zpracovala na své straně.

Kromě zápisu hodnot spotřeby do textového souboru může ukládání dat probíhat přes klientskou aplikaci a databázi MariaDB. Pomocí bash skriptu jsou z výpisu klientské funkce DLMS vyčteny informace potřebné pro jejich zápis a ty jsou do této databáze uloženy podle patřičných parametrů. Na obrázku 2.8 je zobrazena databáze v prostředí PHPmyadmin s hodnotami přijatými z klientské aplikace DLMS.

Posledním krokem procesu je obsah této databáze zprostředkovat uživateli snadno přístupným způsobem. Proto bylo implementováno zobrazování hodnot měření ve webovém prohlížeči. Pro tyto účely je na zařízení Raspberry Pi zřízen webový server NGINX. Ten má přístup do databáze MariaDB a v prohlížeči zobrazuje naměřené hodnoty získané z klientské aplikace pomocí skriptu v jazyce PHP. Ukázka tohoto výpisu ze znázorněna na obrázku 2.9.

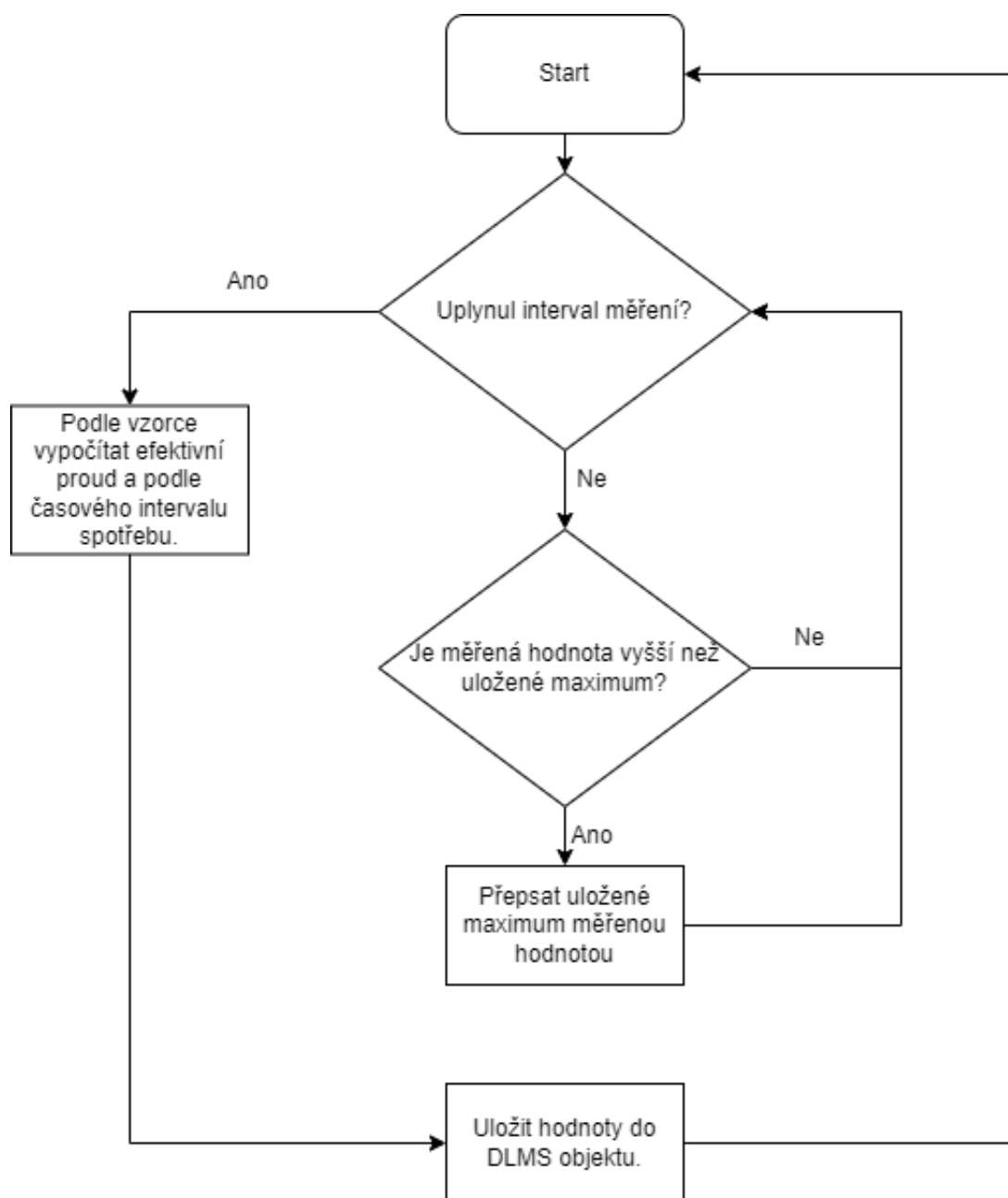
Tím, že je server zřízen na straně Raspberry Pi, které má vlastní úložiště, je možné ukládat hodnoty měření. Nicméně jediná hodnota, kterou je nutné si pamatovat je spotřeba v obou tarifech. Tyto hodnoty jsou tedy průběžně ukládány do



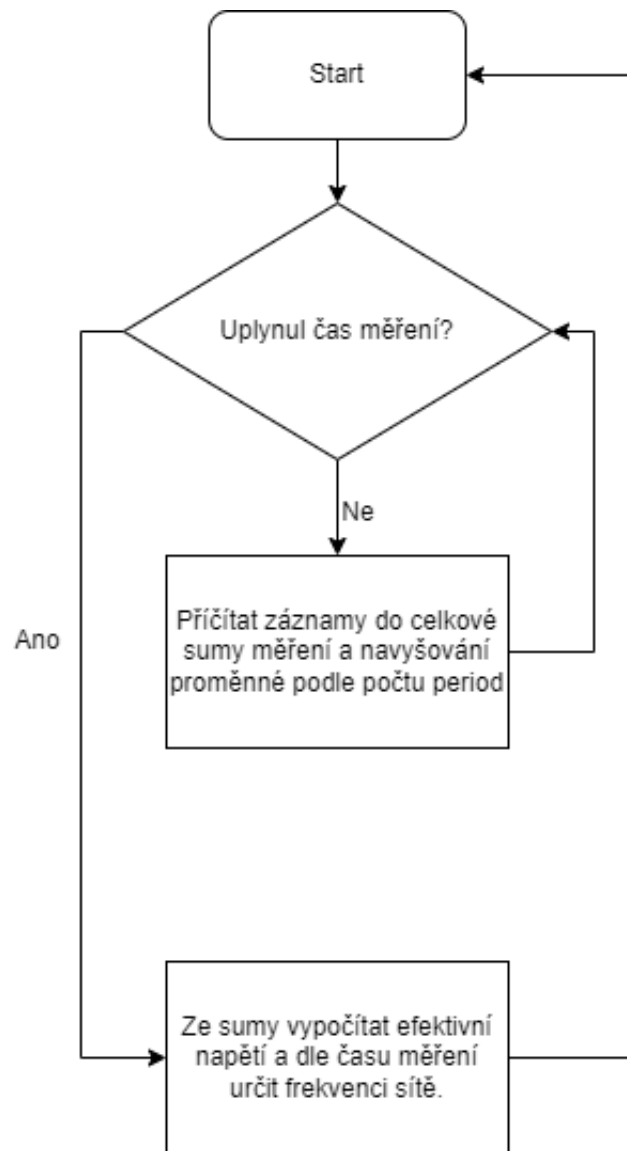
textových souborů. Arduino slouží jako měřicí sonda a Raspberry Pi jako samotný elektroměr pro klientské strany a vytváří DLMS data na základě vstupů ze strany Arduina. Schéma výsledného elektroměru je zobrazeno na obrázku 2.10.

Tímto je tedy zrealizován chytrý elektroměr, který je schopen měřit údaje sítě střídavého proudu, tato data vhodně zapisovat, a poté je odesílat do zbytku sítě.

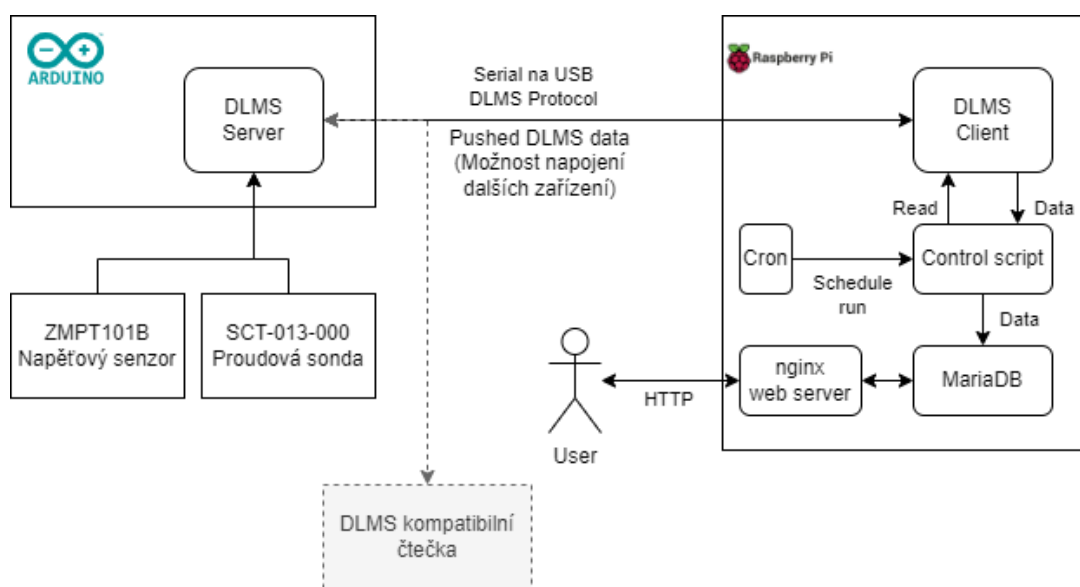
Dále je uskutečněna klientská strana, která data umí ukládat do databáze a tuto databázi zobrazit na vlastním webovém serveru.



Obr. 2.2: Vývojový diagram pro zjištění efektivního proudu a spotřeby.



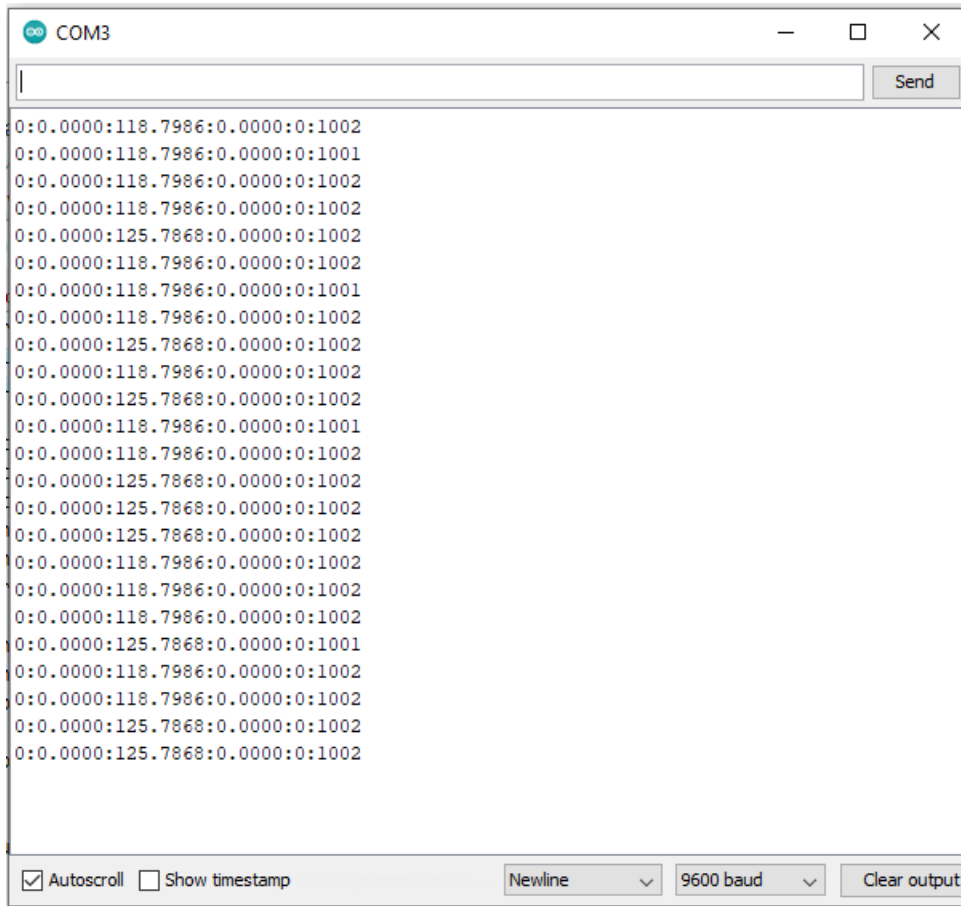
Obr. 2.3: Vývojový diagram pro zjištění efektivního napětí a frekvence.



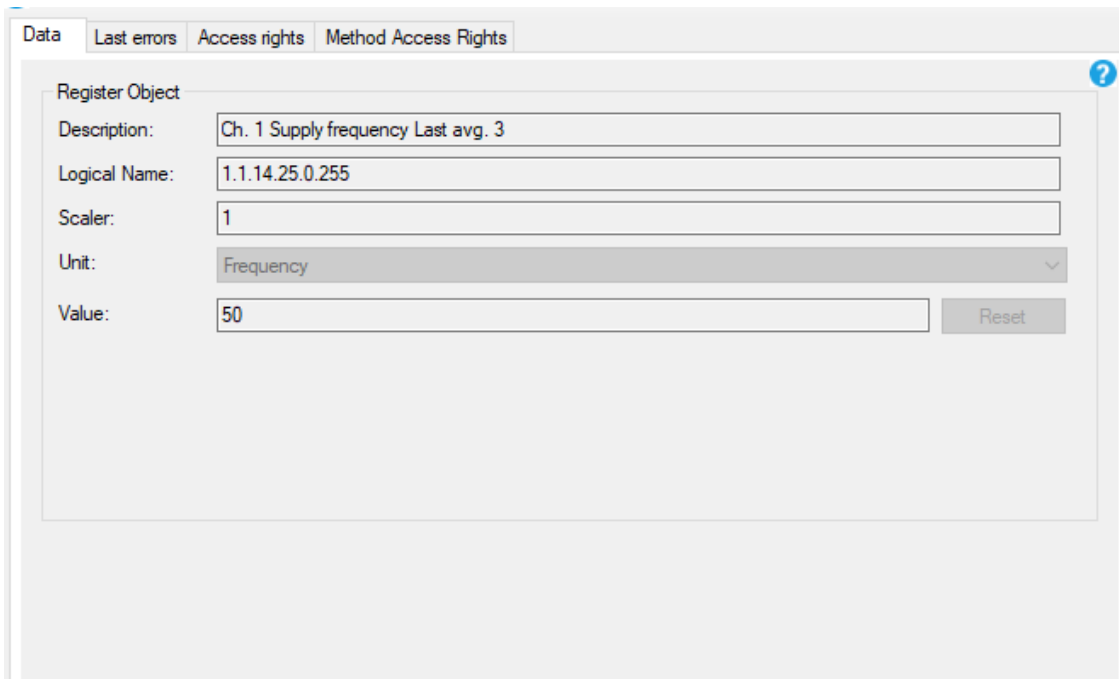
Obr. 2.4: Schéma původní realizace chytrého elektroměru



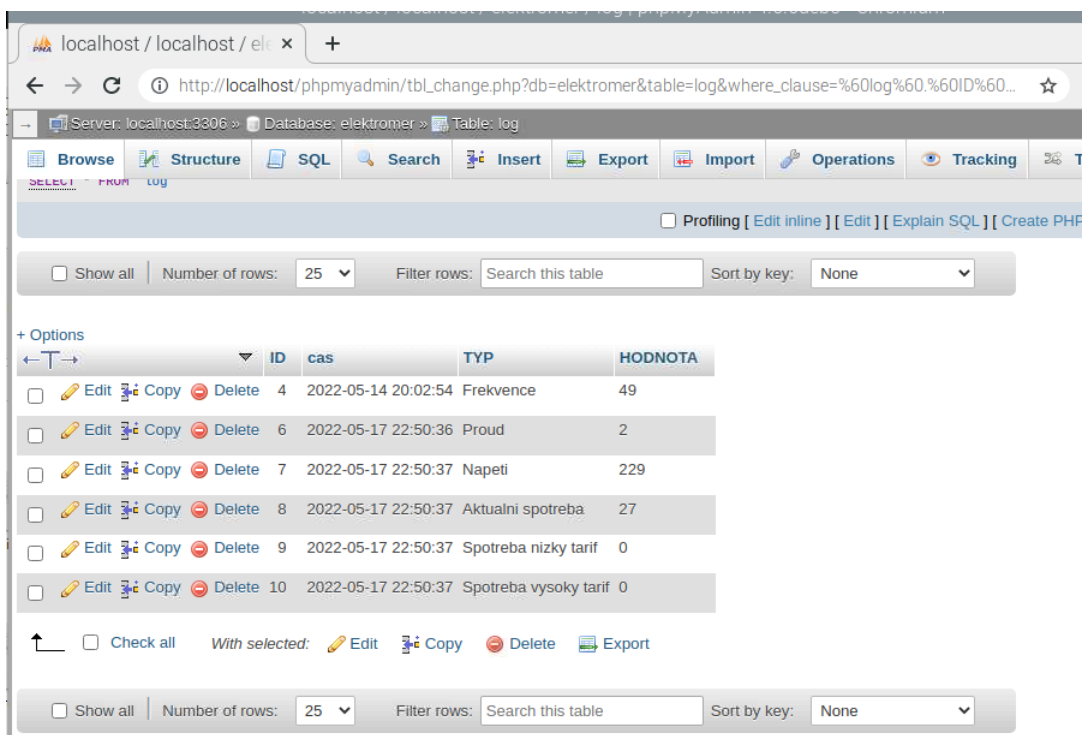
Obr. 2.5: Výpis funkcí měření v prostředí Arduino IDE



Obr. 2.6: Výpis funkcí měření ve zjednodušeném formátu.



Obr. 2.7: Čtení hodnoty frekvence aplikací GXDLMS Director

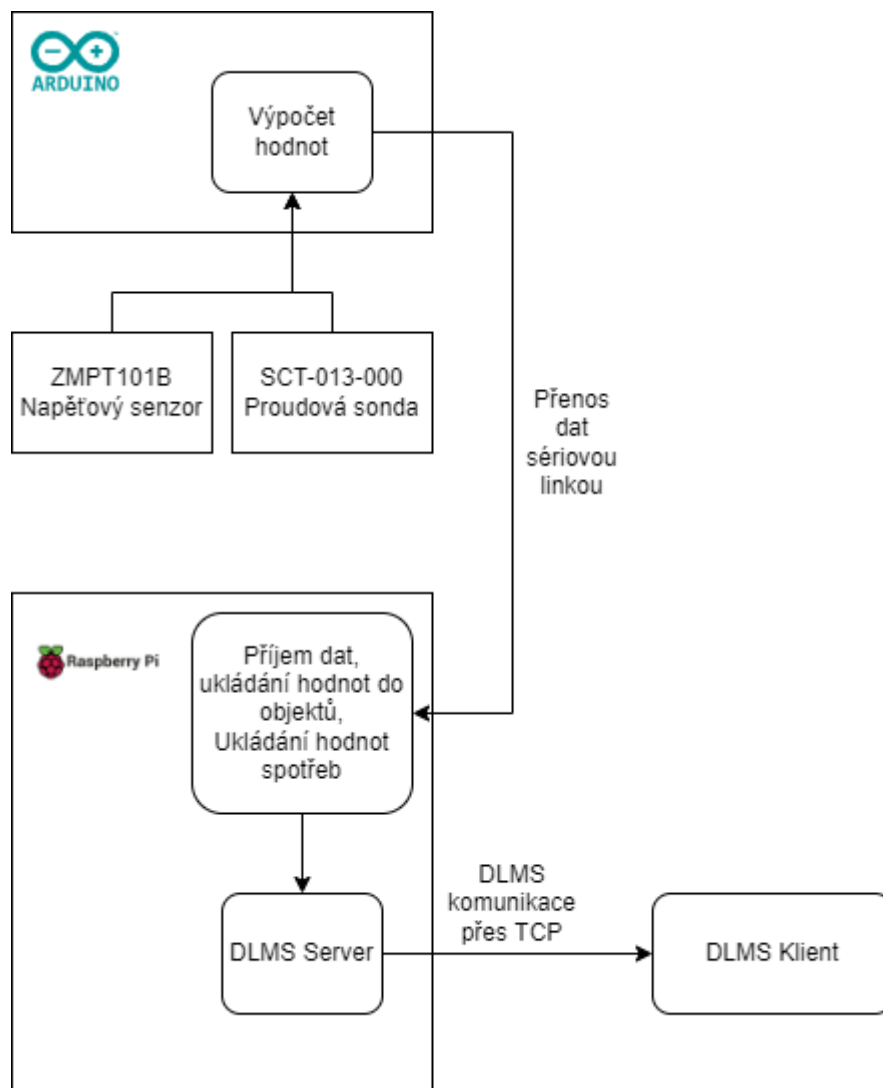


Obr. 2.8: Databáze záznamů měření v PHPmyadmin

ID	cas	TYP	HODNOTA
4	2022-05-14 20:02:54	Frekvence	49
6	2022-05-17 22:50:36	Proud	2
7	2022-05-17 22:50:37	Napeti	229
8	2022-05-17 22:50:37	Aktualni spotreba	27
9	2022-05-17 22:50:37	Spotreba nizky tarif	0
10	2022-05-17 22:50:37	Spotreba vysoky tarif	

Obr. 2.9: Výpis hodnot měření webového serveru





Obr. 2.10: Výsledné schéma chytrého elektroměru

# Závěr

Diplomová práce seznamuje čtenáře se základní problematikou týkající se specifikace DLMS/COSEM a hlavně komunikačního protokolu DLMS. Jejím praktickým výstupem je funkční chytrý elektroměr, takže zahrnuje i teorii týkající se použitého hardwaru. To znamená uvedení do problematiky platform Arduino a Raspberry Pi, popis fyzikálních principů, podle kterých měřicí sondy fungují a způsob, jakým naměřená data zprostředkovávají dále .

Hlavním cílem této práce bylo vytvořit funkční chytrý elektroměr s využitím jednodeskového počítače Raspberry Pi, který dokáže měřit údaje elektrické sítě se střídavým proudem. Konkrétně jsou těmito hodnotami celková spotřeba, aktuální spotřeba a frekvence v síti. Další důležité vlastnosti jsou podpora měření spotřeby ve dvou tarifech a odesílání těchto hodnot v podobě DLMS dat přes protokol TCP.

Pro měření samotné byla u výsledného elektroměru využita platforma Arduino. Hlavními důvody jsou podpora mnoha externích přídatných modulů pro měření a vestavěný převaděč střídavého proudu. Proto bylo nutné pro Arduino vytvořit kód, který ze vstupů použitých modulů vypočítá požadované hodnoty a následně je odešle sériovou linkou na desku Raspberry Pi.

Použitými moduly byl senzor napětí ZMPT101B, jehož výstup je analogová hodnota napětí a proudový senzor SCT-013-000, který na výstupu generuje napětí na základě Hallova jevu. Výpočet efektivních hodnot probíhá až softwarově na straně Arduina.

Jednodeskový počítač Raspberry Pi slouží jako serverová strana protokolu DLMS. Jedná se tedy o stranu elektroměru, která odesílá DLMS data klientským stranám. Pro zřízení serveru bylo využito kódu v jazyce Java dostupného z Github repositáře společnosti Gurux.

Tento kód musel být upraven pro podporu několika objektů odpovídajících veličin. Také bylo nutné implementovat několik funkcí pro správnou komunikaci s platformou Arduino. Dále musel být implementován výpočet, ukládání a zápis těchto hodnot do příslušných objektů. Odesílání těchto objektů aplikace podporovala ve výchozím stavu.

Podpora tarifů spočívá v navyšování hodnoty jednoho ze dvou objektů spotřeby. To, která hodnota je navyšována závisí na výstupu z desky Arduino. Na ní je připevněn přepínač a s jeho pomocí je tedy možné přepínat měřený tarif. Přepínači se dalo přednost před softwarovým řešením proto, že elektroměry v praxi jsou řízeny hromadným dálkovým ovládním ze strany dodavatele energie právě přes přepínač a toto řešení je tím simulováno.

Všechny cíle této práce tedy byly splněny. Elektroměr měří požadované hodnoty, zpracovává je do DLMS objektů a odesílá je TCP spojením jako DLMS data. Elek-

troměr také podporuje měření celkové spotřeby ve dvou tarifech. Navíc je řešení rozšířeno o vzorovou klientskou stranu, ze které je možné ukládat data do databáze a zobrazovat je uživateli ve webovém prohlížeči.

# Literatura

- [1] KOHOUT, D.: *Zátěžový generátor zpráv DLMS/COSEM*. Brno: 2019, 59 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Ing. Tomáš Lieskovan.
- [2] Advantages and disadvantages of smart meters. *Look After My Bills*. [online], [cit. 2021-12-12]. Dostupné z URL: <<https://lookaftermybills.com/blog/advantages-disadvantages-of-smart-meters/>>
- [3] Raspberry Pi Documentation. *Raspberry Pi OS* [online], [cit. 2021-12-12]. Dostupné z URL: <<https://www.raspberrypi.com/documentation/computers/os.html>>
- [4] What is Arduino? *Arduino* [online], [cit. 2021-12-12]. Dostupné z URL: <<https://www.arduino.cc/en/Guide/Introduction>>
- [5] The basics of current sensors *DigiKey* [online], [cit. 2021-12-12]. Dostupné z URL: <<https://www.digikey.com/en/articles/the-basics-of-current-sensors>>
- [6] Efektivní hodnota střídavého proudu *elektross* [online], [cit. 2021-12-12]. Dostupné z URL: <[http://elektross.gjn.cz/soucastky/jeden\\_prechod/ef\\_hodnota.html](http://elektross.gjn.cz/soucastky/jeden_prechod/ef_hodnota.html)>
- [7] ZMPT101B Current-type Voltage Transformer *Qingxian Zeming Langxi Electronic* [online], [cit. 2021-12-12]. Dostupné z URL: <<http://5nrorwxhmqjijik.leadongcdn.com/attachment/kjilKBmoioSRqlkqjoipSR7ww7fgzb73m/ZMPT101B-specification.pdf>>
- [8] ZMPT101B AC Voltage Sensor Module Primer *electroschematics.com* [online], [cit. 2021-12-12]. Dostupné z URL: <<https://www.electroschematics.com/voltage-sensor/>>
- [9] Raspberry Pi 3 Model B 64-bit 1GB RAM. *RPishop* [online], [cit. 2021-12-12]. Dostupné z URL: <<https://rpishop.cz/raspberry-pi-3b/283-raspberry-pi-3-model-b-64-bit-5060214370028.html>>
- [10] 100A AC Proudový senzor neinvazivní SCT-013-000. *Drátek* [online], [cit. 2022-04-04]. Dostupné z URL: <<https://dratek.cz/arduino/1248-100a-ac-proudova-sonda-neinvazivni-sct-013-000.html>>

- [11] BUŠ, O.: *Simulátor DLMS koncentrátoru*. Brno: 2020, 61 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Ing. Tomáš Lieskovan.
- [12] INA219 DIY DC Current Sensor Breakout Module - Blue Deal extreme [online], [cit. 2021-12-12]. Dostupné z URL: <<https://www.dx.com/p/ina219-diy-dc-current-sensor-breakout-module-blue-2053458.html>>
- [13] INA219 Zero-Drift, Bidirectional Current/Power Monitor With I2C Interface [online], [cit. 2021-12-12]. Dostupné z URL: <[https://www.rototron.info/wp-content/uploads/INA219\\_datasheet.pdf](https://www.rototron.info/wp-content/uploads/INA219_datasheet.pdf)>
- [14] Raspberry Pi INA219 Tutorial [online], [cit. 2021-12-12]. Dostupné z URL: <<https://www.rototron.info/raspberry-pi-ina219-tutorial/>>
- [15] Chris Borrill GitHub [online], [cit. 2021-12-12]. Dostupné z URL: <<https://github.innominds.com/chrisb2>>
- [16] GuruX GitHub [online], [cit. 2021-12-12]. Dostupné z URL: <<https://github.com/Gurux/gurux.dlms.java>>
- [17] Easy-to-use UI for Gurux DLMS Component *Gurux for DLMS Smart Meters* [online], [cit. 2021-12-12]. Dostupné z URL: <<https://www.gurux.fi/GXDLMSDirector>>

## Seznam symbolů a zkratek

<b>DLMS</b>	Device language message specification
<b>DLMS UA</b>	DLMS user association
<b>COSEM</b>	Companion specification for energy metering
<b>OBIS</b>	Oracle Business Intelligence Systems
<b>LED</b>	Light-Emitting Diode
<b>IoT</b>	Internet of Things
<b>ISO</b>	International organization of Standardization
<b>OSI</b>	Open Systems Interconnection
<b>csv</b>	Comma separated values
<b>IDE</b>	Integrated development environment
<b>USB</b>	Universal serial bus
<b>RMS</b>	Root mean square
<b>PCB</b>	Polychlorované bifenyly
<b>HDO</b>	Hromadné dálkové ovládání
<b>TCP</b>	Transmission Control Protocol