

AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ

Praca dyplomowa magisterska

*Intelligentne narzędzia zarządzania rozproszonymi źródłami
i magazynami energii elektrycznej*

*Intelligent tools for management of distributed
energy sources and storages*

Autor: mgr inż. Krzysztof Woźny
Kierunek studiów: Informatyka
Opiekun pracy: dr inż. Andrzej Firlit

Kraków, 2019

Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.”, a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.) „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej „sądem koleżeńskim”, oświadczam, że niniejszą pracę dyplomową wykonałem osobiście i samodzielnie i że nie korzystałem ze źródeł innych niż wymienione w pracy.

Spis treści

Wykaz skrótów i oznaczeń	5
Wstęp	7
1. Cel i zakres pracy	11
2. Sieci „inteligentne” stan obecny i perspektywa rozwoju	13
2.1. Koncepcja budowy sieci „inteligentnej”	14
2.2. Prosument jako aktywny uczestnik sieci elektroenergetycznej.....	16
2.3. Generacja rozproszona	19
2.4. Potencjał OZE.....	21
2.4.1. Zasoby energetyczne wiatru.....	21
2.4.2. Energia promieniowania słonecznego.....	22
2.5. Magazyny energii	23
3. Opis wykorzystanych narzędzi	25
3.1. Narzędzia techniczne	25
3.1.1. Sterownik PLC	25
3.1.2. Moduły wykonawcze sterownika PLC	27
3.1.3. Moduł komunikacyjny GSM/GPRS.....	28
3.1.4. Panel operatora	29
3.1.5. Listwa kontrolno-pomiarowa	30
3.1.6. Komputer jednopłytkowy SBC	30
3.1.7. Drukarka 3D	31
3.1.8. Kamera termowizyjna	32
3.2. Oprogramowanie	33
3.2.1. <i>e!COCKPIT</i>	34
3.2.2. GX Soft	35
3.2.3. Interpreter Python.....	35
3.2.4. AutoCAD LT.....	37
3.2.5. SOLIDWORKS.....	37
3.2.6. Ultimaker Cura	38
3.2.7. InfluxDB.....	39

3.2.8. Grafana	40
4. Projekt systemu	43
4.1. Smart Power Guard	44
4.1.1. Oprogramowanie sterownika	48
4.2. Smart EMS	56
4.2.1. Oprogramowanie układu mikroprocesorowego	60
4.2.1. Projekt układu scalonego i obudowy	75
5. Praktyczna realizacja systemów pomiarowych.....	77
5.1. Smart Power Guard	77
5.2. Smart EMS	84
Podsumowanie	96
Literatura	98
Spis tabel i rysunków	100

Wykaz skrótów i oznaczeń

KE	– Klaster Energetyczny
WE	– Wirtualna elektrownia z ang. Virtual Power Plant - VPP
P, Q, S	– moc czynna, bierna, pozorna
P_n	– moc zainstalowana
GR	– Generacja rozproszona
OZE	– Odnawialne źródła energii
DER	– Rozproszone źródła energii z ang. Distributed Energy Resources
SEE	– System elektroenergetyczny
RZE	– Rozproszone źródła energii
KSE	– Krajowy system energetyczny
JST	– Jednostki samorządu terytorialnego
PHES	– Elektrownie szczytowo–pompowe z ang. Pumped Hydroelectric Energy Storage
BESS	– z ang. Battery Energy Storage System
CAES	– Pneumatyczne zasobniki energii z ang. Compressed Air Energy Storage
SCES	– Zasobniki oparte o superkondensatory z ang. Supercapacitor Energy Storage
FES	– Kinetyczne zasobniki energii z ang. Flywheel Energy Storage
SMES	– Nadprzewodnikowe zasobniki energii z ang. Super Conductor Magnetic Energy Storage
SQL	– Strukturalny język zapytań używany do tworzenia, modyfikowania baz danych oraz do umieszczania i pobierania danych z baz danych z ang. Structured Query Language
SCADA	– System nadzorujący przebieg procesu technologicznego lub produkcyjnego z ang. Supervisory Control And Data Acquisition
KEiASPE	– Katedra Energoelektroniki i Automatyki Systemów Przetwarzania Energii
OSD	– Operator Systemu Dystrybucyjnego – przedsiębiorstwo zajmujące się dystrybucją energii elektrycznej
OSP	– Operator Sieci Przesyłowych – podmiot odpowiedzialny za przesył energii elektrycznej
ICT	– Narzędzia informatyczno-telekomunikacyjne z ang. information and communication technologies
EMS	– System zarządzania energią z ang. Energy Management System
DCC	– System dystrybucyjny z ang. Distribution Dispatching Center

JEE	– Jakość Energii Elektrycznej
DB	– Baza danych z ang. Database
DSM	– Zarządzanie popytem z ang. demand side management
DSR	– Zarządza się zapotrzebowaniem odbiorcy z ang. Demand Side Response
ODBC	– Otwarte łącze baz danych z ang. Open DataBase Connectivity
JDBC	– Łącze do baz danych w języku Java z ang. Java DataBase Connectivity
GPZ	– Główny Punkt Zasilania
GUI	– Graficzny interfejs użytkownika z ang. graphical user interface
JWCD	– Jednostka wytwórcze centralnie dysponowane

Wstęp

Obecnie energia elektryczna jest jednym z najczęściej wykorzystywanych typów energii, którą ludzie nauczyli się pozyskiwać i przetwarzać. Zapotrzebowanie na energię elektryczną wciąż rośnie, a pierwotne źródła energii mają coraz mniejsze zasoby. Jak informuje Międzynarodowa Agencja Energetyczna (IEA) w opracowaniu *Global Energy & CO₂ Status Report*, światowe zużycie energii ogółem w 2018 r. wzrosło niemalże dwukrotnie szybciej niż w 2010 r. z danych wynika również dwukrotnie szybszy wzrost produkcji energii z odnawialnych źródeł energii (OZE) – w szczególności z wiatru i promieniowania słonecznego. Niemniej jednak w przeważającej wielkości to wciąż paliwa kopalne pokryły rosnący popyt. Z raportów IEA wynika, że wzrost zapotrzebowania na energię na świecie w 2018 r. względem roku poprzedniego wyniósł 2,3% , natomiast na samą energię elektryczną o 3,9% (900 TWh). Głównym czynnikiem mającym wpływ na zwiększenie zapotrzebowania na energię elektryczną jest rosnąca liczba urządzeń w systemie elektroenergetycznym (SEE). Tendencja ta jest zauważalna zarówno w gospodarstwach domowych, jak i zakładach produkcyjnych. Wielu ekspertów jako przyczynę wzrostu zapotrzebowania na energię podaje także zmiany klimatu, które bezpośrednio przekładają się na eskalację zapotrzebowania na ogrzewanie i przede wszystkim chłodzenie (klimatyzację). Pomimo intensywnego rozwoju odnawialnych źródeł energii w dalszym ciągu zauważalny jest wzrost emisji CO₂, w 2018 r. W USA wzrost wyniósł o 3,1%, Indie 2,8%, Chiny 2,5%. Natomiast w Europie zanotowano spadek emisji o 1,3%. Za spadek emisji CO₂ w Europie uznaje się zwiększenie ilości rozproszonych OZE powstałych w miejsce starych elektrowni węglowych oraz przechodzenie z węgla na gaz, a także budowanie nowoczesnych instalacji wychwytywania i magazynowania CO₂, do ponownego wykorzystania *carbon capture use and storage* (CCUS). Intensywny wzrost ilości OZE i GR takich jak elektrownie: wodne, wiatrowe, słoneczne, geotermalne, biomasy stały się przyczyną wielu problemów i wyzwań dla obecnego systemu elektroenergetycznego. Tradycyjna koncepcja SEE zakłada przesył energii elektrycznej z elektrowni o dużej mocy przez sieci przesyłowe i rozdzielcze do odbiorcy końcowego. Natomiast rozwój OZE powoduje, iż system w przyszłości przekształci się w „inteligentny” system elektroenergetyczny zwany *Smart Grid* (SG). Nowym podmiotem SEE, który w obecnym czasie bardzo dynamicznie się rozwija jest „prosument” łączący cechy odbiorcy i wytwórcy energii elektrycznej. Gwałtowny wzrost ilości prosumentów oraz rozproszonych wytwórców energii spowodował znaczący przepływ mocy i energii w SEE, gdzie energia jest przesyłana od sieci niskich napięć poprzez sieci rozdzielcze, aż do sieci przesyłowych.

Energia elektryczna (EE) jako towar dostępny na rynku jest bardzo nietypowa i wymaga ciągłego bilansowania mocy i energii. Jedną z cech EE na dzień powstania niniejszej pracy jest brak możliwości efektywnego magazynowania energii na dużą skalę. Oczywiście istnieje wiele metod przechowywania energii elektrycznej zarówno w postaci pierwotnej jak i pośrednio, niemniej jednak żadna z nich nie jest efektywna kosztowo aby magazyn mógł zapewniać bezpieczeństwo energetyczne dla dużego obszaru systemu elektroenergetycznego. Trudność

bilansowania wynika z faktu, że wyprodukowana energia musi być cały czas odbierana, przy czym zapotrzebowanie jest ciągle i zmienne w czasie. Zmienność ta powoduje, że w SEE musi być utrzymywana rezerwa mocy w dużych elektrowniach zawodowych. Zwiększenie ilości rozproszonych źródeł energii w krajowym miksie energetycznym o stosunkowo małych mocach może zapewnić niezbędną nadwyżkę mocy w godzinach szczytu. Powoduje to konieczność rozbudowy sieci oraz zastosowania nowych układów pomiarowych i sterowania.

W latach siedemdziesiątych XX wieku pojawiła się koncepcja inteligentnych sieci *Smart Grid*, wówczas określenie to odnosiło się do sieci przesyłowych. Obecnie (2019) pojęcie to ma bardzo szerokie znaczenie i porusza wiele zagadnień z różnych dziedzin takich jak elektrotechnika, automatyka, elektronika, informatyka, telekomunikacja. W literaturze podane są różne definicje dla sieci „inteligentnych” *Smart Grid*. W przypadku sieci elektroenergetycznych w dużym uproszczeniu to system połączeń i komunikacji pomiędzy wszystkimi uczestnikami rynku energii, który ma zapewnić: bezpieczeństwo energetyczne poprzez samoczynne reagowanie na awarie i przerwy w dostawie energii, obniżenie kosztów i poprawę efektywności energetycznej oraz integrację rozproszonych źródeł w tym OZE z SEE. Rozwój sieci typu *smart* jest bardzo intensywny, a w bieżącym okresie operatorzy systemów dystrybucyjnych (OSD) prowadzą badania nad wykorzystaniem „sztucznej inteligencji” tzn. uczenia maszynowego do kreowania zachowań w celu zwiększenia komfortu (poprawy dostaw jakości energii elektrycznej) oraz otrzymania korzyści makroekonomicznych. Nieodzownym elementem *Smart Grid* są „inteligentne” systemy pomiarowe typu *smart meter* (SM), które umożliwiają gromadzenie wielu parametrów pomiarowych. Posiadają również możliwość dwukierunkowej komunikacji przy wykorzystaniu łączności elektronicznej. Przykładem takiego systemu mogą być liczniki typu AMI (*ang. Advanced Metering Infrastructure*), które służą do pomiarów energii elektrycznej współpracując z oprogramowaniem nadrzędnym do zarządzania danymi pomiarowymi MDM (*ang. Meter Data Management*). Zastosowanie dwukierunkowej komunikacji umożliwia np. OSD kontrolować dostawy energii, a także zarządzać przepływami w sieci, szybciej reagować na awarię, czy zarządzać wyłączeniami. z drugiej strony sieć „inteligentna” umożliwia odbiorcy końcowemu aktywne zarządzanie zapotrzebowaniem, a w przypadku prosumenta także produkcją. System *Smart Grid* wykorzystuje szereg czujników, mierników, a także rozbudowane oprogramowanie i aplikacje wspierające podejmowanie decyzji. Aplikacje wspierają takie procesy jak planowanie, poprzez wsparcie systemami prognozowania (zapotrzebowania, produkcji, cen energii). Rozwój sztucznej „inteligencji” (*ang. artificial intelligence, AI*), zwłaszcza w kontekście logiki rozmytej, sieci neuronowych, systemów ekspertowych i robotyki. W szczególności rozwój AI jest to powiązany z tzw. Internetem rzeczy (IoT) oraz przemysłem 4.0. Oba pojęcia oznaczają integrację „inteligentnych” urządzeń i maszyn do poprawy komfortu życia oraz w przemyśle do efektu ekonomicznego. W tym momencie wielu deweloperów w porozumieniu z OSD integruje system zarządzania „inteligentnym” budynkiem BMS (*ang. Building Management System*) z systemami pomiarowymi za pomocą sieci typu HAN (*ang. Home Area Network*).

Takie rozwiązanie w perspektywie czasu daje możliwość przełączenia wydzielonych odbiorów energochłonnych np. na lokalne źródła energii, lub całkowite ich wyłączenie.

Eskalacja energetyki rozproszonej w tym OZE jest również zauważalna na terenie naszego kraju. Jak wynika z danych Urzędu Regulacji Energetyki (URE) po wygaszeniu w 2016 r. systemu zielonych certyfikatów zanotowano stopniowy spadek powstawania nowych instalacji OZE. Wybudowano wówczas 1,445 GW (głównie elektrownie wiatrowe), aby inwestycje mogły funkcjonować jeszcze przed wygaszeniem systemu certyfikatów. Wprowadzony zamiennie system aukcyjny na początku działalności nie był jasno sprecyzowanym, czego skutkiem było niemal dziesięciokrotny spadek przyłączeń nowych źródeł wytwórczych (w roku 2017 było to 122,8 MW, a w 2018 r. 55 MW¹). Rok 2019 „tchnął nowego ducha” w rozwój instalacji OZE. Odnotowany przez URE wzrost mocy wytwórczej OZE w pierwszej połowie tego roku wyniósł 226,46 MW², głównie przez powstanie nowych farm fotowoltaicznych (z 112,3 MW na 259, 25 MW), elektrowni na biomasę (1362,8 MW do 1467,9 MW) o 105,1 MW, natomiast nieznaczny przyrost w biogazowniach (1,4 MW) i elektrowniach wiatrowych (16,7 MW). Uwzględniając dane udostępnione przez Polskie Sieci Elektroenergetyczne (PSE) zawierające całkowitą moc instalacji fotowoltaicznych łącznie z mikroinstalacjami prosumenckimi w pierwszej połowie 2019 roku wzrosła o 295,1 MW (z 471,4 MW na 766,5 MW), w sierpniu podano 882,6 MW, a pierwszego września wynik wynosił 940,9 MW. Tak gwałtowny przyrost źródeł o niestabilnym charakterze produkcji jest podłożem wielu problemów w SEE. Dla przykładu, farmy wiatrowe przyczyniły się do gwałtownych zmian w przepływie mocy, a w zależności od mocy zwarciowej SEE są także znaczącym źródłem wahań napięcia. z kolei duży wolumen prosumenckich mikroinstalacji fotowoltaicznych skutkuje wzrostem napięcia oraz asymetrii (odnotowano przypadki poboru w dwóch fazach oraz produkcji w trzeciej i odwrotnie).

Ministerstwo Energii (ME) również dostrzega potrzebę lokalnego bilansowania produkcji i zapotrzebowania o czym świadczy uruchomienie dwóch konkursów na Certyfikat Pilotażowego Klastra Energii. Koncepcja ta ma ułatwić dostęp do stabilnych dostaw energii i samowystarczalności na poziomie gminy, czy powiatu. Do pierwszego konkursu zgłoszono 115 wniosków, z czego 33 otrzymały certyfikaty. Do drugiego konkursu zakwalifikowano 84 wnioski, gdzie ponownie 33 z nich otrzymały certyfikaty. Wszystkie KE jednoznacznie określiły, że pierwszym krokiem do prawidłowego funkcjonowania jest inteligentny system pomiarowy, który umożliwi nadzorowanie wolumenu produkcji i zapotrzebowania, który będzie umożliwiał aplikację dodatkowych funkcjonalności względem zwykłych liczników energii. Opracowany system w ramach pracy magisterskiej jest propozycją łatwego i ekonomicznie lukratywnego układu bilansowania mikroinstalacji wytwórczych oraz odbiorów.

^{1,2} Dane URE, nie uwzględniają instalacji mikroinstalacji prosumenckich, a jedynie instalacje posiadające koncesję i wpis do rejestru wytwórców oraz zarejestrowane instalacje biogazowe.

1. Cel i zakres pracy

Niniejsza praca przedstawia aktualne na rok 2019 zagadnienia „inteligentnych sieci” w kontekście systemu elektroenergetycznego oraz Klastrow Energii. Praca łączy zagadnienia z kilku dziedzin, począwszy od elektrotechniki, elektroniki poprzez telekomunikację, automatykę do nowoczesnych narzędzi informatycznych oraz integrację poszczególnych komponentów w jeden system.

Bezpośrednim celem pracy był projekt i budowa „inteligentnego” narzędzia do zarządzania rozproszonymi źródłami i magazynami energii elektrycznej dla mikroinstalacji prosumenckich (bilansowania).

Kolejnym celem było opracowanie algorytmów i bazy danych pomiarowych różnych typów źródeł oraz bazy dla optymalizacji pracy zasobnika energii, a także wykonanie narzędzia kontroli mocy dla odbiorów o dużej mocy.

Ponadto praca zawiera cele pośrednie, którymi są: analiza obecnego stanu założeń dla bilansowania OZE oraz organizacja i konfiguracja transmisji danych.

System składa się z opracowanego urządzenia kontrolno-pomiarowego działającego wspólnie z instalacjami wytwórczymi, pompami ciepła oraz odbiorami. Urządzenie ma współpracować z narzędziami typu *open source*, co dodatkowo ma zwiększyć uniwersalność i potencjał jego zastosowania w Klastrach Energii. Założono również, że urządzenie może pracować autonomicznie oraz komunikować się do systemu nadrzędnego w zależności od potrzeb użytkownika. Oprócz pomiarów energetycznych urządzenie może również rejestrować temperaturę oraz, w zależności od wersji, może sterować wybranymi odbiorami.

Oprogramowanie zostało podzielone na część dla urządzenia pomiarowego oraz system wizualizacji. W zależności od zaawansowania użytkownika istnieją dwa typy interfejsów do komunikowania się z urządzeniem – graficzny z poziomu dowolnej przeglądarki internetowej oraz za pomocą konsoli poprzez SSH (*ang. secure shell*). W obu przypadkach komunikacja może zostać nawiązana również w przypadku urządzeń mobilnych z systemem Android oraz iOS. Wykorzystanie urządzeń mobilnych w kontekście HMI (*ang. Human Machine Interface*) umożliwia podgląd quasi-online bieżącego zużycia i produkcji oraz bilansu, a także danych historycznych³.

Testy urządzenia przeprowadzono w warunkach laboratoryjnych oraz na rzeczywistych instalacjach prosumenckich. Jako odnawialne źródło energii wykorzystano mikroinstalację fotowoltaiczną oraz mikroturbinę wiatrową, dodatkowo urządzenie było również testowane na obiektach w układzie pompa ciepła i instalacja fotowoltaiczna⁴. W ramach testów laboratoryjnych badano współpracę urządzenia z magazynem energii o mocy 10 kW.

³ Opcja dostępna w zależności od realizacji: urządzenie wraz z lokalną bazą danych.

⁴ Urządzenie wymaga dalszych testów z uwagi na brak pełnego sezonu grzewczego w trakcie prowadzenia testów na gotowym prototypie.

Praca magisterska jest efektem czynności wykonywanych w dwóch projektach badawczych ukierunkowanych na wypracowanie strategii działania i funkcjonalności narzędzi dla energetyki rozproszonej.

Pierwszy z projektów *Renewable Energy and Load Flexibility in Industry (RELflex, Generacja rozproszona i elastyczne struktury popytu na energię w przemyśle)* jest realizowany w ramach programu *ERA-Net Smart Energy Systems* i jest współfinansowany przez *Smart Grids Plus, Horizon 2020, Bundesministerium Für Wirtschaft Und Energie, Narodowe Centrum Badań i Rozwoju*. Realizacja projektu odbywa się we współpracy z następującymi partnerami zarówno polskimi jak i zagranicznymi: Europejski Instytut Miedzi, Fraunhofer, Akademia Górniczo-Hutnicza w Krakowie, Hochschule Magdeburg-Stendal, Willa na Olczańskim Wierchu, Hotel Bachledówka, aRTE möbel GmbH.



„Celem projektu RELflex jest opracowanie, testowanie i ocena rozwiązań oraz zastosowań w zakresie elastyczności istotnych dla energetyki procesów przemysłowych w (sektorze małych i średnich przedsiębiorstw) MŚP poprzez dynamiczne zarządzanie obciążeniami podlegającymi kontroli, wytwarzanie energii odnawialnej oraz magazynowanie energii”⁵.

Drugi projekt *Rozwój energetyki rozproszonej w klastrach energii (KlastER)* jest realizowany w ramach konkursu *Społeczny i gospodarczy rozwój Polski w warunkach globalizujących się rynków GOSPOSTRATEG* finansowanego z *Narodowego Centrum Badań i Rozwoju*. Realizacja projektu odbywa się we współpracy z następującymi partnerami: Ministerstwo Energii – Lider⁶, Akademia Górniczo Hutnicza w Krakowie, Narodowe Centrum Badań Jądrowych.



„Celem projektu KlastER jest wypracowanie Strategii rozwoju klastrów energii w Polsce, w oparciu o szereg przeprowadzonych analiz oraz pilotażowe uruchomienia klastrów z wykorzystaniem opracowanych rozwiązań technologicznych i organizacyjnych. Realizacja projektu KlastER umożliwi skuteczne wdrażanie wpisanego na listę projektów Strategii na rzecz Odpowiedzialnego Rozwoju kompleksowego projektu „Energetyka rozproszona”, za którego implementację podmiotem odpowiedzialnym jest Ministerstwo Energii. W ramach projektu przewiduje się kompleksowe badania uwarunkowań i barier technicznych, prawnych, ekonomicznych i społecznych rozwoju energetyki rozproszonej, w tym klastrów energii”⁷.

⁵ http://www.keiaspe.agh.edu.pl/keiaspe/articles/0254/RELflex_ulozka_a5_PL.pdf – dostęp 01.09.2019 r.

⁶ Obecnie Ministerstwo Energii zostało podzielone zlikwidowane: Nadzór nad spółkami energetycznymi przejmuje Ministerstwo Zasobów Narodowych, oraz powstało nowe Ministerstwo Klimatu w ramach którego będzie kontynuowany projekt.

⁷ <https://www.er.agh.edu.pl/projekt-klaster/> – dostęp 01.09.2019 r.

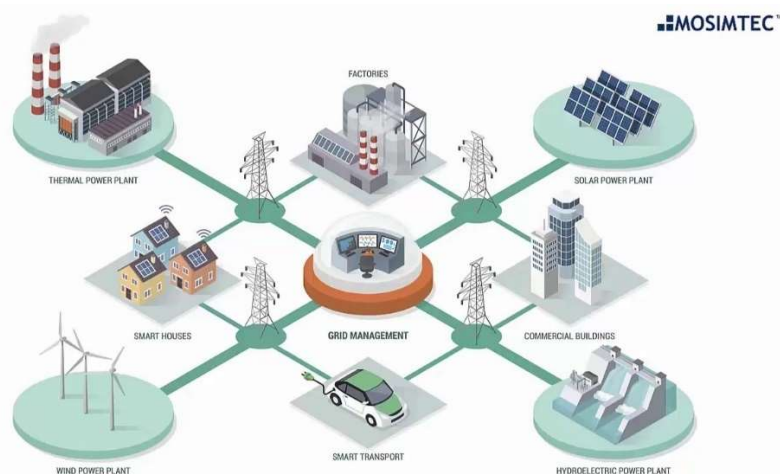
2. Sieci „inteligentne” stan obecny i perspektywa rozwoju

Rozdział przedstawia krótkie wprowadzenie do tematu sieci „inteligentnych” typu *Smart Grid* oraz zawiera opis wybranych pojęć, które zostały wykorzystane w dalszej części pracy. Otóż samo pojęcie obecnie ma bardzo szerokie znaczenie i nie jest jednoznacznie sprecyzowane. W literaturze możemy spotkać bardzo wiele definicji sieci „inteligentnych”, zarówno w kontekście sieci telekomunikacyjnych, informatycznych jak i systemu elektroenergetycznego. Niniejsza praca porusza zagadnienie *Smart Grid* w aspekcie SEE wspartego narzędziami informatycznymi. W pracy poruszono również zagadnienia z zakresu telekomunikacji. Praca nie rozważa tematyki sieci „inteligentnych” w kontekście automatycznej implementacji i konfiguracji infrastruktury telekomunikacyjnej.

Smart Grid w aspekcie sieci elektroenergetycznej zakłada przede wszystkim integrację na wielu poziomach zarówno energetyki zawodowej, jak i źródeł rozproszonych, w tym OZE, mającej na celu poprawę efektywności energetycznej. Często w opisach sieci *Smart Grid* pomija się najważniejszą cechę jej funkcjonalności, a mianowicie zapewnienie bezpieczeństwa energetycznego mającego na celu eliminację przerw w zasilaniu i szybkie reagowanie na stany awaryjne w sieci za pomocą rozbudowanych systemów diagnostycznych. Kolejnym niezmiernie trudnym zagadnieniem jest zapewnienie bezpiecznej i efektywnej ekonomicznie transmisji danych. Kolejnym istotnym zagadnieniem SG jest funkcjonalność integracji rozproszonych źródeł, zwłaszcza w kontekście małych i mikroinstalacji bazujących na modułach fotowoltaicznych, turbinach wodnych, mikroturbinach wiatrowych, a także przydomowych biogazowniach, które nie są w dyspozycji OSD czy OSP. Sieci SG mają również na celu minimalizowanie kosztów wytwarzania i dystrybucji energii poprzez tworzenie lokalnych obszarów bilansowania, gdzie energia jest wytwarzana w pobliżu punkt poboru. Potencjał sieci SG istnieje również w świadczeniu usług dodatkowych, zarówno na rzecz operatorów, jak i lokalnych odbiorców/wytwórców. Ekonomiczność usług dodatkowych w sieciach SG jest ogromna, a poniżej przedstawiono kilka przykładów takich usług:

- przesunięcia czasowe i geograficzne zapotrzebowania na energię,
- ograniczenie inwestycji w przebudowę lub budowę nowych linii,
- bilansowanie lokalnych obszarów (np. poprzez KE),
- rezerwa mocy w systemie,
- kompensacja mocy biernej,
- redukcja negatywnego oddziaływania źródeł niespokojnych,
- poprawa jakości dostaw energii elektrycznej,
- możliwość funkcjonowania w ramach „wyspy energetycznej”,
- sezonowe gromadzenie mocy,
- regulacja częstotliwości,
- regulacja napięcia (w długim i krótkim czasie),

Sieć SG w przyszłości będzie zintegrowana z systemem zarządzania budynkami BMS. W Europie wielu deweloperów oraz OSD współpracuje już na etapie projektowania budynków i sieci dystrybucyjnych, gdzie system automatyki budynku jest zintegrowany np. z licznikami AMI poprzez ogólnie stosowane protokoły komunikacji. Ogólna koncepcja systemu została przedstawiona na Rys. 2.1.



Rys. 2.1 Poglądowa koncepcja Smart Grid⁸

W sieciach „inteligentnych” pokłada się duże nadzieje na rozwój gospodarki i innowacyjności, a także dostrzega główny potencjał czwartej rewolucji Industry 4.0, gdzie głównym aspektem jest automatyzacja oraz przetwarzanie danych zwłaszcza rozproszone w chmurze, a także w technologii IoT. Należy pamiętać, że sieć elektroenergetyczna jest jednym z największych osiągnięć XX wieku, a ponadto została nazwana „największą maszyną świata”. Warto wspomnieć, iż poza oczywistymi aspektami SG w kontekście SEE są to także wszelkiego rodzaju działania niezwiązane z technicznymi czynnikami, takimi jak: mechanizmy rynkowe, elementy ekonomiczne, zawieranie kontraktów, ale także perspektywy społeczne i kulturowe, a także widoczne zmiany behawioralne.

2.1. Koncepcja budowy sieci „inteligentnej”

W trakcie realizacji niniejszej pracy brano pod uwagę wiele aspektów SG, co skutkowało powstaniem wielu koncepcji realizacji założonych na początku celów. Finalnie skupiono się na zagadnieniu bilansowania oraz stworzeniu narzędzia do aktywnej kontroli mocy. Obecnie bardzo często poruszany jest temat eksploracji danych pomiarowych i zastosowaniu sztucznej „inteligencji” do wyznaczania nowych trendów i poszukiwaniu korelacji ówczesnie niedostrzeżonych lub nieistniejących. Warto pamiętać, że skuteczność tego typu narzędzi zależy od rzetelności danych pomiarowych, które stanowią bazę wiedzy dla sztucznej „inteligencji”. Dlatego w tym miejscu należy przedstawić narzędzie będące podstawą dla prawidłowego funkcjonowania SG, a bez którego SG nie może się rozwinąć

⁸ Źródło: <https://www.istockphoto.com/pl/wektor/smart-grid-and-power-supply-gm673573970-123518981> - dostęp z dnia 25.11.2019 r

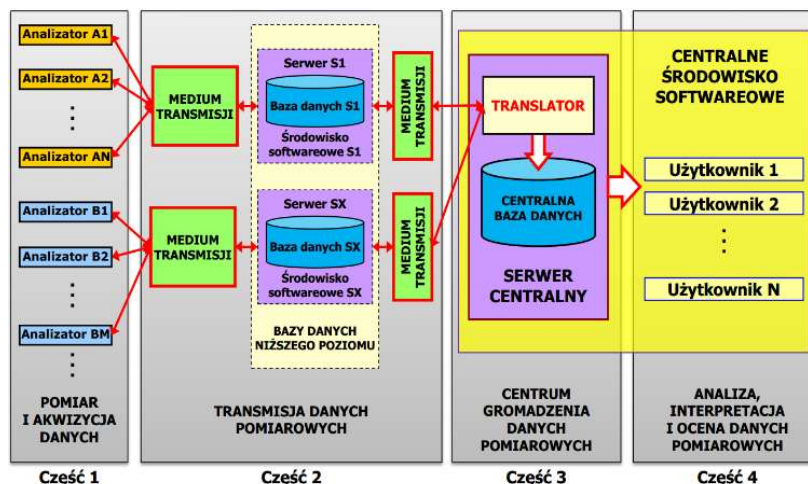
mianowicie jest to *smart metering (SM)*. W przeważającej większości spotykanych opisów sprowadza SM do Pomiarów energii elektrycznej, a w szczególności liczników AMI. Jednakże SM nie ogranicza się tylko do liczników EE, mogą to być urządzenia pomiarowe do odczytu i kontroli takich obszarów jak sieci dystrybucyjne wody, gazu, ciepła, ale także zaawansowane analizatory z funkcjami bezpieczeństwa. SM w uogólnieniu jest to elektroniczne urządzenie do akwizycji danych pomiarowych, które następnie przekazuje je do systemu nadrzędnego, w odróżnieniu od liczników AMR (*ang. automatic meter reading*) jest wyposażone w dwukierunkową komunikację oraz cechuje się możliwością współpracy z sieciami HAN.

Struktura systemu typu *Smart Meter*

System typu SM wchodzący w skład SG z założenia jest dedykowany do długoterminowej ciągłej pracy. Ważną kwestią nowoczesnych systemów jest ich otwartość i możliwość zastosowania urządzeń pochodzących od różnych producentów, stopień uniwersalności jest zależny od wielu czynników technicznych związanych z częścią sprzętową i programową. W przypadku systemu obsługującego urządzenia wielu producentów napotykamy problem kompatybilności zarówno urządzeń jak i formatów zapisywanych danych. Dlatego coraz częściej sięga się po rozwiązania typu *open source* i uniwersalne protokoły wymiany danych. Ponadto rozwój infrastruktury telekomunikacyjnej umożliwia transmisję danych w miejsca wcześniej niedostępne. W związku z popularyzacją IoT coraz częściej sięga się po wykorzystanie standardu Ethernet oraz Wi-Fi w przemysłowych urządzeniach wykonawczych wcześniej obsługiwanych łączami szeregowymi. Systematycznie urządzenia są wyposażane w zaawansowane protokoły komunikacyjne bazujące na sieci Ethernet, takie jak CC-Link IE, EtherCAT, EtherNet/IP, PROFINET, POWERLINK, SERCOS czy nieustający Modbus (ModbusTCP). Powyższe protokoły mają zapewnić determinizm czasowy niewystępujący w standardowym Ethernetie. Tendencję tą potwierdza wyposażanie urządzeń takich producentów jak: Bosch, Sick, Rockwell Automation, Phoenix Contact, Omron w standard EtherNet/IP (*ang. Ethernet Industrial Protocol*). W odróżnieniu od zwykłego Ethernetu dodatkowo zastosowano protokół CIP (*ang. Common Industrial Protocol*) i dwa tryby transmisji danych. Dane nie podlegające reżimowi czasowemu m.in. agregowane dane pomiarowe i konfiguracyjne przesyłane są za pomocą protokołu TCP (*ang. Transmission Control Protocol*), natomiast dane krytyczne objęte reżimem czasowym są przesyłane z użyciem protokołu UDP (*ang. User Datagram Protocol*). Warto wspomnieć, że komunikaty są dodatkowo synchronizowane z użyciem protokołu PTP (*ang. Precision Time Protocol*), co teoretycznie umożliwia synchronizację na poziomie mikrosekundy.

Obecne systemy SM mają strukturę wielowarstwową – począwszy od akwizycji danych pomiarowych poprzez transmisje i repozytoria danych do zaawansowanej analizy gromadzonych informacji. Przykładową strukturę systemu SM na bazie rozproszonego

systemu ciągłego monitorowania parametrów jakości dostaw energii elektrycznej (JDEE) przedstawiono na Rys. 2.2.



Rys. 2.2 Schemat ogólny systemu Smart Meter na przykładzie systemu monitorowania jakości dostaw energii elektrycznej⁹

Zasadniczo w najbardziej uogólnionym podziale SM zbudowany jest z dwóch warstw:

- sprzętowej (hardware): w skład, której wchodzi wszystkie urządzenia fizyczne,
- programowej (software): aplikacje do konfiguracji, zarządzania i analizy gromadzonych informacji.

Natomiast przedstawiony na Rys. 2.2 podział na 4 warstwy jest wykorzystywany w większości dostępnych na rynku rozwiązań:

- warstwa 1 – pomiar i akwizycja danych, urządzenia kontrolno-pomiarowe wraz z osprzętem towarzyszącym, czujnikami, przekładnikami, przekaźnikami etc.,
- warstwa 2 – transmisja danych pomiarowych m.in. media transmisyjnego i moduły komunikacyjne, koncentratory danych,
- warstwa 3 – centrum gromadzenia danych pomiarowych, zawierające serwer centralny komunikujący się z główną bazą danych oraz bazami innych podsystemów, w przypadku bardzo rozbudowanych struktur posiadająca silnik eksplorujący hurtownie danych,
- warstwa 4 – środowisko softwareowe wykorzystane do wizualizacji i analizy oraz interpretacji danych pomiarowych, w praktyce warstwa jest uruchamiana na zewnętrznych końcówkach klienckich (w przypadku małych systemów warstwa 4 może zostać uruchomiona na jednej maszynie z warstwą 3).

2.2. Prosument jako aktywny uczestnik sieci elektroenergetycznej

Najczęściej ówczesnie przytaczany kontekst określania prosumenta odnosi się do producenta, a zarazem konsumenta EE zlokalizowanego na poziomie sieci niskiego

⁹ Źródło: ROZPROSZONE SYSTEMY MONITOROWANIA JAKOŚCI ENERGII ELEKTRYCZNEJ, dr.inż. Andrzej Firlit – Laboratorium jakości energii elektrycznej 04.06.2014

napięcia. Jednakże termin prosument pochodzi od połączenia dwóch słów producent i konsument, co oznacza podmiot zaangażowany w proces wytwarzania produktu, i jednocześnie jego konsumpcję (adekwatnie w przypadku towarów oraz usług). Rozpatrując sposób kooperacji pomiędzy prosumentami można rozróżnić trzy typy prosumpcji:

- prosumpcja indywidualna – niezależność i samodzielność cechująca się brakiem kooperacji z innymi prosumentami,
- intra-prosumpcja – grupowa prosumpcja zachodząca w granicach kooperacji prosumentów (dobrym przykładem takiej działalności są wspomniane we wstępie narzędzia typu *open source*),
- inter-prosumpcja – grupa prosumentów współpracująca z podmiotem zewnętrznym najczęściej producentem mająca na celu modyfikację produktu¹⁰.

Niniejsza praca opisuje zagadnienie prosumenta w ujęciu produkcji EE z OZE. Pierwsza wzmianka o prosumencie w polskim ustawodawstwie¹¹ definiowała go jako odbiorcę końcowego, który kupuje energię jednocześnie ją produkując w posiadanej przez siebie instalacji wytwórczej, a także zużywa ją na własne potrzeby, niezwiązane z działalnością gospodarczą. Rok 2019 zaowocował zmianą definicji prosumenta i obecnie ma ona następujące brzmienie.

„27 a) prosument energii odnawialnej – odbiorcę końcowego wytwarzającego energię elektryczną wyłącznie z odnawialnych źródeł energii na własne potrzeby w mikroinstalacji, pod warunkiem, że w przypadku odbiorcy końcowego niebędącego odbiorcą energii elektrycznej w gospodarstwie domowym, nie stanowi to przedmiotu przeważającej działalności gospodarczej określonej zgodnie z przepisami wydanymi na podstawie art. 40 ust. 2 ustawy z dnia 29 czerwca 1995 r. o statystyce publicznej (Dz. U. z 2019 r. poz. 649 i 730)”¹².

Zmiana definicji oraz innych poszczególnych zapisów w ustawie o OZE pociąga za sobą nowe zasady rozliczania prosumenta. Od 14 sierpnia 2019 r. określenie prosument dotyczy również podmiotów gospodarczych, a tym samym z tzw. system opustów dotyczy również MŚP oraz jednostek samorządu terytorialnego (JST). Oznacza to, iż podmioty gospodarcze (tak samo jak indywidualne gospodarstwa domowe) mogą wytwarzać energię elektryczną na własne potrzeby wraz z możliwością „zmagazynowania” nadwyżek w sieci OSD. Nadwyżki EE oddane do SEE mogą zostać wykorzystane w późniejszym okresie w stosunku:

- 1 do 0,7 – dla instalacji większej niż 10 kW mocy zainstalowanej,
- 1 do 0,8 – dla instalacji mniejszej niż 10 kW mocy zainstalowanej,

¹⁰ T. Szymusiak, Prosument–Prosumpcja–Prosumeryzm. Ekonomiczne oraz społeczne korzyści prosumpcji na przykładzie Polski oraz Niemiec (podejście naukowe), Wydawnictwo Bezkrępy Wiedzy, 2015

¹¹ Ustawa z dnia 20 lutego 2015 r. o odnawialnych źródłach energii (Dz. U. z 2018 r., poz. 2389 t.j. ze zm.)

¹² Ustawa z dnia 19 lipca 2019 r. o zmianie ustawy o odnawialnych źródłach energii oraz niektórych innych ustaw (Dz. U. z 2019 r., poz. 1525)

Różnica pozostaje do dyspozycji sprzedawcy na poczet pokrycia kosztów rozliczeń.

Prosument jest również zwolniony ze zmiennej części opłaty dystrybucyjnej. W nowej wersji ustawy zapisano również obowiązek przenoszenia oddanych nadwyżek do kolejnych okresów rozliczeniowych. Nowelizacja ustawy umożliwi także dopuszczenie podmiotów trzecich jako do sprzedawców dla prosumentów, a zarazem sprzedawców zobowiązanych w przypadku, gdy kontraktuje on energię na zasadach rynkowych. Jako ułatwienie do prowadzenia statystyk i rozliczeń narzucono na OSD konieczność przekazywania danych pomiarowych o agregacji godzinowej, aby zapewnić ochronę interesów prosumenta oraz bezpieczeństwo i niezawodność KSEE.

Zmiennie uległ również zapis o mocy zainstalowanych instalacji i obecnie jego brzmienie jest następujące:

„16) montażu pomp ciepła, wolnostojących kolektorów słonecznych, urządzeń fotowoltaicznych o mocy zainstalowanej elektrycznej nie większej niż 50 kW oraz mikroinstalacji biogazu rolniczego w rozumieniu art. 19 ust. 1 ustawy z dnia 20 lutego 2015 r. o odnawialnych źródłach energii (Dz. U. z 2018 r. poz. 2389, z późn. zm.³⁾) z zastrzeżeniem, że do urządzeń fotowoltaicznych o mocy zainstalowanej elektrycznej większej niż 6,5 kW oraz mikroinstalacji biogazu rolniczego, stosuje się obowiązek uzgodnienia pod względem zgodności z wymaganiami ochrony przeciwpożarowej projektu budowlanego, o którym mowa w art. 6b ustawy z dnia 24 sierpnia 1991 r. o ochronie przeciwpożarowej (Dz. U. z 2019 r. poz. 1372 i 1518), oraz zawiadomienia organów Państwowej Straży Pożarnej, o którym mowa w art. 56 ust. 1a tej ustawy;¹³⁾”

Poprzednia wersja ustawy zakładała maksymalną moc instalacji prosumenckich wynoszącą 40 kW. Obecnie zwiększono maksymalną moc o 25% i wynosi ona 50 kW, co w przypadku bilansowania MŚP i JST może mieć istotne znaczenie. Warto również wspomnieć o zmianach w kierunku liberalizacji zasad uzyskania pozwolenia na budowę mikroinstalacji oraz wprowadzaniu zmian w ustawie z dnia 27 marca 2003 r. o planowaniu i zagospodarowaniu przestrzennym (Dz. U. z 2018 r. poz. 1945 oraz z 2019 r. poz. 60, 235, 730 i 1009). Nowe zapisy przewidują możliwość lokalizacji mikroinstalacji na terenach o innym przeznaczeniu niż produkcyjne, za wyjątkiem zakazu zapisanego przez lokalny plan zagospodarowania terenu. Drugą istotną zmianą jest zniesienie zasady tzw. dobrego sąsiedztwa przy ocenie wydawania warunków zabudowy dla instalacji OZE.

Na powyższym nie kończą się pozytywne zmiany legislacyjne w tym roku (2019). W ustawie z dnia 27 kwietnia 2001 r. – Prawo ochrony środowiska (Dz. U. z 2019 r. poz. 1396, z późn. zm.³⁾) wprowadzono następującą zmianę:

"Przez budownictwo objęte społecznym programem mieszkaniowym rozumie się obiekty budownictwa mieszkaniowego lub ich części, z wyłączeniem lokali użytkowych, lokale mieszkalne w budynkach niemieszkalnych sklasyfikowanych w Polskiej

¹³ Ustawa z dnia 19 lipca 2019 r. o zmianie ustawy o odnawialnych źródłach energii oraz niektórych innych ustaw (Dz. U. z 2019 r., poz. 1525)

Klasyfikacji Obiektów Budowlanych w dziale 12, oraz obiekty sklasyfikowane w Polskiej Klasyfikacji Obiektów Budowlanych w klasie ex 1264 - wyłącznie budynki instytucji ochrony zdrowia świadczących usługi zakwaterowania z opieką lekarską i pielęgniarską, zwłaszcza dla ludzi starszych i niepełnosprawnych, a także mikroinstalację, o której mowa w art. 2 pkt 19 ustawy z dnia 20 lutego 2015 r. o odnawialnych źródłach energii (Dz. U. z 2018 r. poz. 2389, z późn. zm.), funkcjonalnie z nimi związaną, z zastrzeżeniem ust. 12b.¹⁴

Dotychczas instalacje fotowoltaiczne w zależności od miejsca usytuowania były objęte różnymi stawkami VAT. Dla lokalizacji w obrębie dachu budynku było to 8%, natomiast dla innych w tym znajdujących się na gruncie 23%. W przypadku prosumentów zmiana oznacza, że właściciele domów jednorodzinnych o powierzchni do 300 m² mają ujednoczoną stawkę VAT wynoszącą 8% na budowę instalacji fotowoltaicznej funkcjonalnie związanej z budynkiem. Zapis rozwiązuje bardzo częsty problem techniczny z ulokowaniem instalacji na dachu budynku, bowiem nowa stawka obowiązuje niezależnie od miejsca montażu.

2.3. Generacja rozproszona

Podrozdział zawiera definicję generacji rozproszonej w sektorze energetycznym, w szczególności dotyczącej energii elektrycznej. Następnie przedstawiono definicję OZE oraz

Generacja rozproszona (GR) - pojęcie generacji rozproszonej odnosi się głównie do systemu bądź zespołu wytwarzającego energię elektryczną, któremu może towarzyszyć wytwarzanie energii skojarzonej energii cieplnej (kogeneracja).

Generację rozproszoną wyróżniają następujące cechy:

- generacja rozproszona nie podlega centralnemu planowaniu rozmieszczenia i jest eksploatowana przez niezależnych użytkowników bądź producentów energii elektrycznej,
- generacja rozproszona nie podlega centralnemu dysponowaniu („wirtualne elektrownie” oraz tzw. „agregatory” skupiający wiele niezależnych źródeł kontrolowanych jako jedna jednostka wytwórcza jest niezgodne definicją),
- źródła GR przyłączane są do sieci rozdzielczych lub sieci elektroenergetycznej odbiorcy za układem kontrolno-rozliczeniowym (poziomy napięcie są zależne od danego kraju),
- jednostki GR nie biorą aktywnego udziału w regulacji częstotliwości i napięcia,
- przyjmuje się, że układy GR mają moc nominalną w zakresie 50 – 150 MW. Natomiast moc nominalna GR jest przedmiotem sporów i nie jest jednoznacznie

¹⁴ Ustawa z dnia 16 października 2019 r. o zmianie ustawy Prawo ochrony środowiska oraz niektórych innych ustaw (Dz. U. z 2019 r., poz. 2166)

określona. W literaturze można spotkać wiele podziałów GR względem mocy, natomiast dla warunków Polskich najczęściej przyjmowany podział jest następujący:

- mikrogeneracja rozproszona 1W – 50 kW – zmiana w ustawie o OZE,
- mała generacja rozproszona 1 kW – 5 MW,
- średnia generacja rozproszona 5 MW – 50 MW,
- duża generacja rozproszona 50 MW – 150 MW,

Spotykane są również podziały gdzie maksymalna moc GR wynosi 300 MW.

Najczęściej spotykany model współpracy GR z KSEE polega na przyłączeniu źródeł mniejszych niż 1 MW do sieci niskiego napięcia, czasami również do średniego napięcia. Kolejno moce powyżej 1 MW pracują w sieciach średniego napięcia, a od mocy 50 MW najczęściej współpracują z sieciami rozdzielczymi 110 kV wykorzystując transformatory. Można przyjąć, że dla warunków prawidłowego funkcjonowania KSEE, źródła rozproszone są przyłączane do sieci o napięciu nieprzekraczającym 110 kV i mocy nie przekraczającej 50 MW – głównie jest to podyktowane maksymalnym prądem obciążenia linii oraz napięciem górnym typowych transformatorów na 110 kV (63 MVA).

Odnawialne źródła energii (OZE) – sformułowanie OZE określa się „wieczne” (niewyczerpalne lub odnawialne w krótkim czasokresie), naturalne źródła energii takie jak wiatr czy słońce. Użycie terminu OZE najczęściej jest używane w kontekście energii elektrycznej, jednakowoż termin odnosi się również do energii cieplnej (kolektory słoneczne, energia geotermalna).

Źródłami energii odnawialnej są:

- biomasa (biomasa stała, biopaliwa),
- biogaz (gazy wysypiskowe i z oczyszczalni ścieków),
- woda,
- wiatr,
- promieniowanie słoneczne (elektrownie cieplne i ogniwa fotowoltaiczne),
- energia geotermalna,
- odpady biodegradowalne,
- energia pływów morskich i falowania,

Bardzo częstym procederem jest mylenie pojęć GR z OZE, należy jednoznacznie powiedzieć, że nie każde źródło rozproszone będzie OZE i odwrotnie. Przykładem źródła OZE, którego nie sposób rozproszyć jest 300 MW farma wiatrowa. Natomiast źródłem rozproszonym o mocy 800 kW może być mała elektrociepłownia zasilania gazem ziemnym, ale już ta sama elektrociepłownia zasilania biogazem będzie OZE. Istnieją również przypadki, gdzie z całego wolumenu produkowanej energii tylko jej część jest klasyfikowana jako OZE. Taka sytuacja zachodzi w wielu spalarniach śmieci, w których istnieje możliwość wykorzystania odpadów biodegradowalnych jako paliwa.

2.4. Potencjał OZE

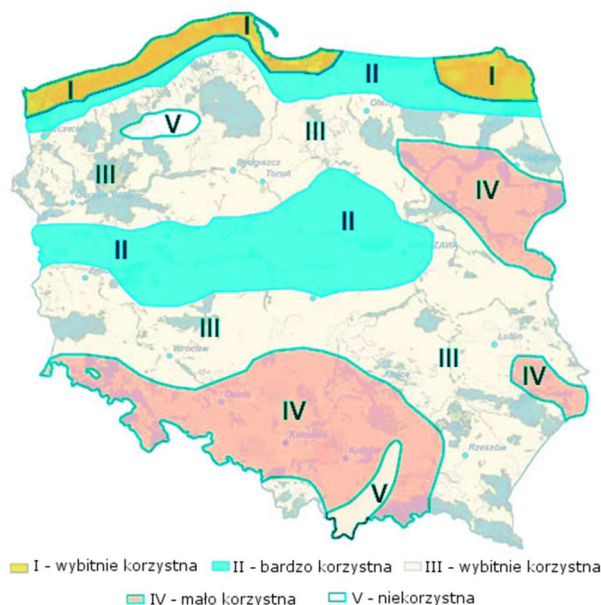
Przytoczone we wstępie niniejszej pracy magisterskiej fakty dotyczące intensywności powstawania instalacji fotowoltaicznych w bieżącym roku jednoznacznie potwierdzają potencjał „drzemiący” w OZE. W przypadku mikroinstalacji prosumenckich najkorzystniejsze obecnie (2019) są instalacje fotowoltaiczne, kolektory słoneczne, pompy ciepła oraz mikroturbiny wiatrowe. W dalszej części podrozdziału przedstawiono rozkład natężenia promieniowania słonecznego oraz stref energetycznych wiatru na tle mapy Polski.

Z uwagi na dostępność obiektów testowych w dalszych rozważaniach ograniczono się do źródeł odnawialnych w postaci wiatru i energii promieniowania słonecznego.

2.4.1. Zasoby energetyczne wiatru

Wiatr powstaje głównie za przyczyną promieniowania słonecznego i ruchu obrotowego Ziemi. Nierównomierne nagrzewanie powierzchni jest przyczyną powstawania mas powietrza o różnym ciśnieniu i temperaturze. Różnica ciśnienia wymusza przepływ mas powietrza, tym samym wzbudzając zjawisko wiatru.

Najistotniejszą cechą wiatru dla instalacji wytwórczych EE jest jego prędkość, która zależy od następujących czynników: struktura terenu, lokalizacja geograficzna (w tym odległość od dużych akwenów wodnych), wysokość nad poziomem morza. Poszczególne strefy i ich rozmieszczenie na mapie Polski przedstawiono na Rys. 2.3.



Rys. 2.3 Strefy energetyczne wiatru na obszarze Polski¹⁵,

Przy inwestycji w siłownie wiatrowe istotnym jest dostęp do danych o zmienności wieloletniej oraz dobowej. Zmiany prędkości wiatru uznane są za proces stochastyczny

¹⁵ Źródło: opracowanie własne

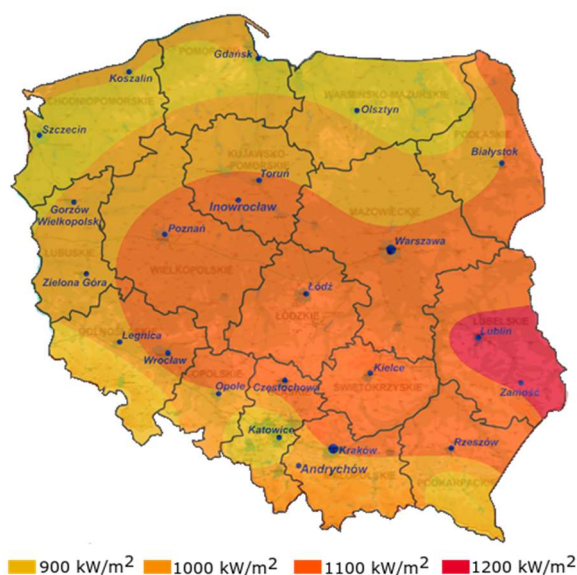
i jest on przedmiotem wielu badań prognozowania z wykorzystaniem sztucznej „inteligencji”.

2.4.2. Energia promieniowania słonecznego

Słońce jest największym dostępnym źródłem energii, które w ciągu 30 minut dostarcza tak ogromną ilość energii do naszej planety, że mogła by ona zaspokoić całoroczne zapotrzebowanie dla ogółu populacji człowieka. Ilość promieniowania, która dociera do powierzchni jest zależna od lokalizacji geograficznej, pory dnia i roku, wysokości nad horyzontem, zachmurzenia, zanieczyszczenia powietrza etc. Podczas planowania wolumenu generacji i projektowania instalacji przetwarzania energii słonecznej na elektryczną lub ciepłą należy uwzględnić następujące parametry:

- nasłonecznienie – jest to stosunek promieniowania słonecznego padającego na powierzchnię w zadanym czasie (kWh/m^2),
- usłonecznienie – liczba godzin, przez które promieniowanie słoneczne dociera do powierzchni Ziemi (w zadanym czasokresie),
- gęstość mocy promieniowania słonecznego – suma całkowitego promieniowania słonecznego (bezpośredniego, rozproszonego, odbitego) docierającego do danej powierzchni (W/m^2).

Polskie warunki (Rys. 2.4) należą do umiarkowanych i dużych średniorocznych obszarach nasłonecznionych. Usłonecznienie dla Polski stanowi ok. 18% czasu w ciągu roku i zawiera się w granicach 1300-1600 godzin. Średnioroczna wartość usłonecznienia wynosi 1600 godzin rocznie, natomiast ilość promieniowania słonecznego docierającego na daną powierzchnię znajduje się w przedziale 950-1250 kWh/m^2 .



Rys. 2.4 Średnioroczne nasłonecznienie dla obszaru Polski¹⁶

¹⁶ Źródło: opracowanie własne

W przypadku energii promieniowania słonecznego, dostępnych jest co najmniej kilka metod jej przetwarzania, takich jak konwersja fotowoltaiczna, fototermiczna, fotochemiczna, wieże słoneczne, a także termoliza wody. Dostępne rozwiązania techniczne różnią się pod względem sprawności oraz efektu energii końcowej. Charakter niniejszej pracy determinuje ograniczenie się w dalszych rozważaniach do efektu fotowoltaicznego, ponieważ do testów opracowanych narzędzi wykorzystano pomiary energii elektrycznej z instalacji fotowoltaicznych.

2.5. Magazyny energii

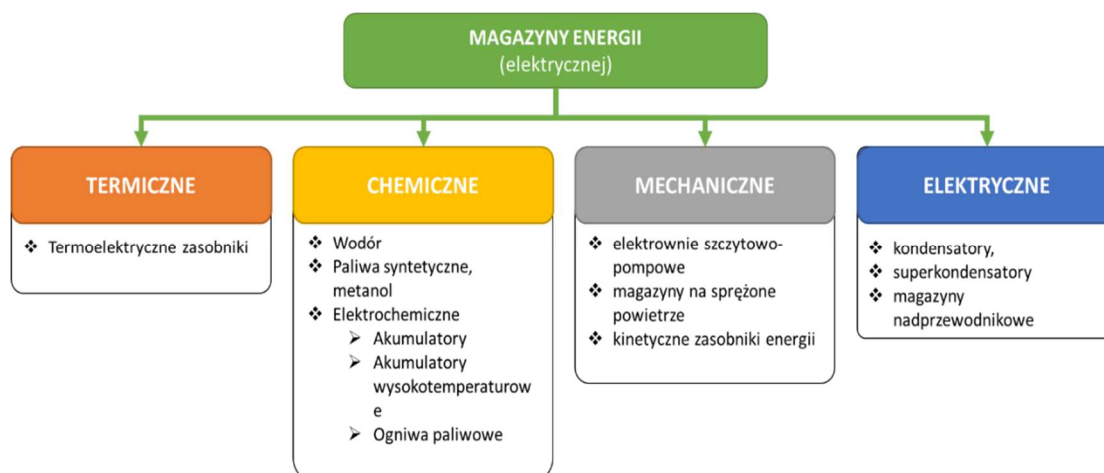
Technologia magazynowania energii ma ogromny potencjał w zakresie poprawy funkcjonowania sieci elektroenergetycznych, a w szczególności dla wzrostu ilości wytwarzanej energii z OZE, na które nałożono ograniczenia mocy. Jest również jedną z alternatyw dla paliw ropopochodnych wykorzystywanych w krajowym sektorze transportu. Potencjał zwiększenia i optymalizacji niezawodności przepływu energii ze źródeł o zmiennym charakterze produkcji t.j. generacja wiatrowa czy słoneczna drzemie w wizji wysokosprawnej technologii magazynowania.

Technologie magazynowania

Proponowany podział magazynów energii ze względu na technologię bezpośrednio związaną z magazynowaniem energii do zastosowań w systemie elektroenergetycznym jest następujący:

Proponowany podział magazynów energii (Rys. 2.5) ze względu na technologię bezpośrednio związaną z magazynowaniem energii do zastosowań w systemie elektroenergetycznym jest następujący:

- Termiczne;
 - termoelektryczne zasobniki,
- Chemiczne;
 - wodór,
 - paliwa syntetyczne, metanol,
 - elektrochemiczne,
 - akumulatory,
 - akumulatory wysokotemperaturowe,
 - ogniwa paliwowe,
- Mechaniczne;
 - elektrownie szczytowo-pompowe,
 - magazyny na sprężone powietrze,
 - kinetyczne zasobniki energii,
- Elektryczne,
- kondensatory,
- superkondensatory,
- magazyny nadprzewodnikowe.



Rys. 2.5 Podział magazynów w zależności od technologii¹⁷

W dalszych rozważaniach wykorzystano zasobnik bazujący na technologii chemicznej w postaci akumulatorów L-ion. Pomiary przeprowadzono w warunkach laboratoryjnych.

¹⁷ Źródło: opracowanie własne

3. Opis wykorzystanych narzędzi

Rozdział przedstawia wykorzystane w realizacji niniejszej pracy narzędzia zarówno programistyczne jak i sprzętowe, które posłużyły do projektu i budowy urządzeń *smart meter*. Użyte rozwiązania techniczne i aplikacyjne podzielono na dwie części: oprogramowanie i osprzęt techniczny. Poszczególne urządzenia przedstawione są w dużym skrócie i uproszczeniu, ponieważ nie były one przedmiotem niniejszej pracy, a jedynie narzędziem do jej realizacji.

3.1. Narzędzia techniczne

Podpunkt przybliży wybrane istotne narzędzia techniczne i urządzenia, które posłużyły do wykonania opracowanych w ramach pracy urządzeń. Specyfikacje urządzeń ze szczegółowymi opisami są dostępne w witrynach producentów.

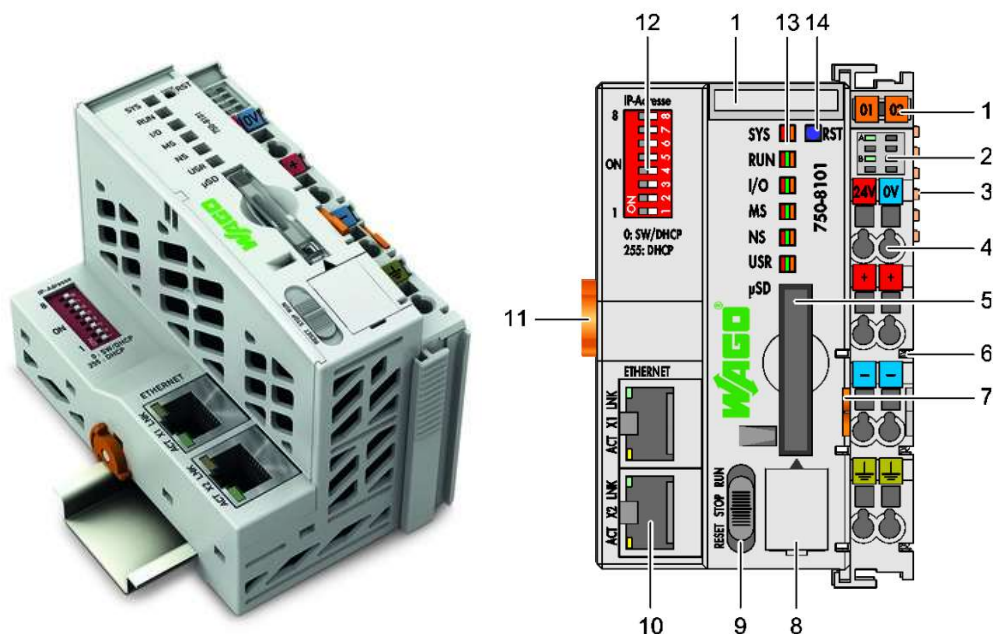
3.1.1. Sterownik PLC

Sterownik PLC (*ang. programmable logic controller*) to programowalny sterownik logiczny, który jest dedykowany do współpracy z urządzeniami technologicznymi w przemyśle. Zasada działania sterownika PLC bazuje na cyklicznym obiegu pamięci programu, gdzie w pierwszej kolejności jest prowadzona autodiagnostyka, a następnie odczytywane są stany wejść. W kolejnym etapie wykonuje się zaimplementowany algorytm, a w ostatniej fazie zdania atomowe ustawiają parametry wyjść.

Na rynku dostępnych jest wiele modeli sterowników PLC – od najprostszych w kompaktowej obudowie zawierającej jednostkę centralną CPU (*ang. central processing unit*) wraz z modułami wejściowymi, wyjściowymi i komunikacją. Zdecydowana jednak większość przypadków użycia to aplikacje niestandardowe wymagające dopasowania sterownika do obiektu sterowanego. Dlatego na chwilę obecną (2019) przeważająca część stosowanych sterowników ma budowę modułową. Należy również dodać, że aplikacje linii automatyki przemysłowej niegdyś realizowane przez systemy DCS (*ang. distributed control system*) aktualnie są zastępowane z powodzeniem przez sterowniki PLC o dużych mocach obliczeniowych.

W podpunkcie przedstawiono zastosowany sterownik PLC wraz z modułami kontrolno-pomiarowymi. Sterownik PFC100 wyposażony w e!RUNTIME WAGO wyróżnia się on niezwykle wysoką wydajnością zachowując przy tym niewielkie wymiary Rys. 3.1. Sterownik jest wyposażony w dwa złącza ETHERNET i dodatkowo w mikroprzełączniki DIP. Wysoki poziom bezpieczeństwa informatycznego zapewniony jest dzięki wbudowanej obsłudze SSL/TLS, VPN i firewall. Sterownik wspiera liczne protokoły, m.in. Modbus-TCP oraz otwartą komunikację TCP-UDP. Urządzenie jest wyposażone również w gniazdo karty microSD. Do tworzenia aplikacji na sterownik PFC100 przeznaczone jest środowisko inżynierskie e!COCKPIT

wykorzystujące technologię CODESYS 3. Dzięki osadzeniu w systemie operacyjnym czasu rzeczywistego Linux®, wykorzystaniu HTML5 i współdziałaniu z wbudowanym systemem zarządzania sterownikiem przez www. Aplikacje pracują niezawodnie i pozwalają na optymalne wykorzystanie zasobów sterownika.



Rys. 3.1 Sterownik PLC PFC100¹⁸

Wykaz oznaczeń:

- 1) oznaczniki,
- 2) kontrolka LED zasilania sterownika,
- 3) złączeni magistrali danych,
- 4) zaciski zasilania sterownika,
- 5) port karty pamięci
- 6) złącza zasilania modułów I/O,
- 7) pasek zwalniający sterownik z szyny DIN,
- 8) złącze serwisowe (pod zaślepką),
- 9) przełącznik trybu pracy sterownika,
- 10) złącza Ethernet,
- 11) zabezpieczenie sterownika przed wypięciem z szyny DIN,
- 12) przełącznik wyboru adresu,
- 13) kontrolki LED pracy sterownika,
- 14) przycisk reset (wewnątrz wpustu).

Szczegółowy opis funkcjonalności i instrukcja obsługi sterownika znajduje się w załączniku nr 2.

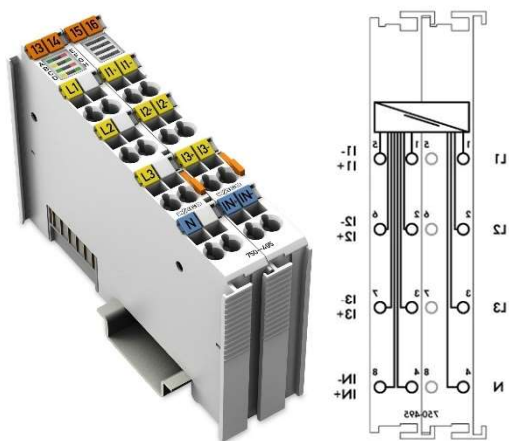
¹⁸ Źródło: nota katalogowa

3.1.2. Moduły wykonawcze sterownika PLC

Opracowane urządzenie składa się z następujących modułów: pomiar energii elektrycznej, moduł z wyjściami cyfrowymi umożliwiającymi kontrolę obwodów zewnętrznych¹⁹. Sterownik został również wyposażony w specjalistyczny moduł do komunikacji z siecią LonWorks²⁰.

Moduł pomiaru prądu i napięcia

Do pomiaru parametrów elektrycznych w przypadku urządzenia dedykowanego do pracy w rzeczywistych warunkach przemysłowych wykorzystano moduł pomiarowy skierowany do współpracy ze sterownikami Wago serii 750-8XX. Moduł umożliwia pomiar prądów i napięć oraz parametrów energetycznych niezależnie dla każdej fazy. Konstrukcję modułu kieruje się jest do pomiarów w sieciach 4-ro przewodowych. Moduł pomiarowy jest przystosowany do pracy w obwodach o prądzie do 5 a i napięciu do 690 V, Rys. 3.2.



Rys. 3.2 Moduł pomiaru mocy 750-495²¹

Autor prac przeprowadził konfigurację ustawień modułu zgodnie z zaleceniami pomiarowymi zleceniodawcy. W tabeli 1 przedstawiono wykaz mierzonych parametrów elektrycznych przy pomocy modułu pomiarowego.

Szczegółowe informacje dotyczące modułu znajdują się w nocie katalogowej.

Moduł do komunikacji z systemem BMS

Kolejny moduł to 753-648, który służy do komunikacji z siecią LonWorks²². Rys. 3.3. Moduł I/O LON® umożliwia szybkie i łatwe podłączenie sieci LonWorks® do sterownika oraz innych modułów I/O z serii 750. Swobodnie konfigurowany interfejs sieciowy pozwala zarządzać maksymalnie 249 zmiennymi.

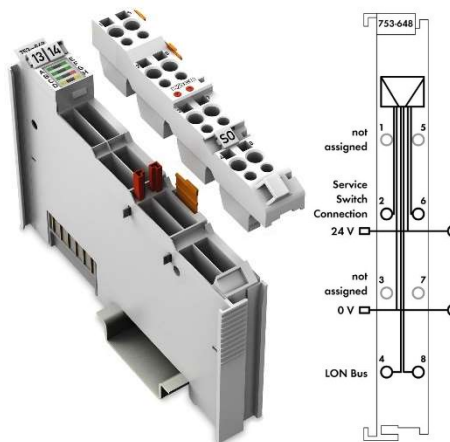
¹⁹Sterowanie obwodami zewnętrznymi jest opcją dodatkową i wymaga przeprowadzenia konfiguracji.

²⁰ Konfiguracja modułu LON jest możliwa po doprowadzeniu magistrali komunikacyjnej.

²¹ Źródło: nota katalogowa

²² LonWorks (ang. *local operating network*) to dedykowana platforma sieciowa stworzona specjalnie do zastosowań w automatyce przemysłowej, która umożliwia komunikację różnymi mediami transmisji.

Moduł I/O LON® 753-648 oferowany przez WAGO pozwala zintegrować z WAGO-I/O-SYSTEM 750 z urządzeniami obiektowymi pracującymi w sieci LonWorks®. Zaletami tego typu rozwiązania jest elastyczność i swobodna interfejsu dla zmiennych obiektowych. Do konfiguracji modułu wykorzystywany jest konfigurator WAGO LON®, który wchodzi w skład pakietu WAGO-I/O-PRO. Służy on do programowania i uruchamiania modułu I/O. Tworzy on również plik .xif („External Interface File“), umożliwiającą bezproblemową integrację interfejsu sieciowego z dowolnym narzędziem zarządzania siecią LonWorks®. Sieć LonWorks® może zostać prawidłowo skonfigurowana po doprowadzeniu magistrali obiektowej LON®. Do sterownika mogą być przyłączone maksymalnie dwa moduły. Liczba modułów jest zależna od rozmiaru pamięci wykorzystanej w danej aplikacji oraz od typu sterownika



Rys. 3.3 Moduł komunikacji z siecią LonWorks 753-648²³

Szczegółowe informacje dotyczące modułów przedstawionych w punkcie 3.2.2 przedstawiono w załączniku 5.

3.1.3. Moduł komunikacyjny GSM/GPRS

Jako medium komunikacji zastosowano urządzenie GPRS-A (Rys. 3.4), które jest uniwersalnym modułem monitorującym, mogącym pracować autonomicznie lub w ramach systemów automatyki. Urządzenie wyposażone jest w telefon GSM, obsługujący transmisję danych w technologii 2G. W systemach alarmowych moduł ten może służyć do realizacji monitoringu przy współpracy z dowolną centralą alarmującą. Połączenie z nią odbywa się za pośrednictwem odpowiednio skonfigurowanych wyjść sterownika PLC. Urządzenie posiada osiem wejść, które można ustawić jako cyfrowe (NO, NC) lub analogowe. Te, które pracują jako analogowe, mogą dodatkowo być wykorzystywane do monitorowania szerokiego spektrum zewnętrznych urządzeń, takich jak np. czujniki temperatury, ciśnienia, wilgotności. GPRS-A posiada także magistralę 1-Wire, do której można podłączyć maksymalnie 8 cyfrowych czujników temperatury DS-T1. Moduł może wysyłać dane pomiarowe z użyciem otwartych

²³ Źródło: nota katalogowa

protokołów komunikacyjnych: MQTT, JSON oraz MODBUS RTU, poprzez GPRS, np. odpowiedzi na określone zdarzenia typu włączenie ogrzewania przy znacznym spadku temperatury. Moduł oferuje powiadomianie maksymalnie 8 użytkowników na kilka sposobów: wiadomościami SMS lub PUSH, bądź poprzez usługę CLIP. Na poczet instytucji, w której testowano opracowane urządzenie SM skonfigurowano dwa numery do powiadomień SMS i jeden kontrolny serwisowy. Urządzenie posiada 4 programowalne wyjścia, którymi można sterować zdalnie z wykorzystaniem SMS, CLIP, aplikacji mobilnej GX CONTROL lub programu konfiguracyjnego GX Soft. GPRS-A może służyć do zdalnego sterowania innymi podłączonymi do modułu urządzeniami. Programowanie i konfiguracja odbywa się na komputerze z zainstalowanym programem GX Soft. Połączenie z modułem może być lokalne, poprzez port RS-232 (TTL) modułu lub zdalnie²⁴, przez GPRS (wymagane stałe IP). Urządzenie zostało przedstawione na rysunku 14



Rys. 3.4 Moduł komunikacji i powiadomień SMS, GPRS-A²⁵

Szczegółowe informacje na temat modułu zostały przedstawione w załączniku 6, natomiast na płycie CD dołączono plik konfiguracyjny do programu GX Soft.

3.1.4. Panel operatora

Jako panel operatora do współpracy z PLC zastosowano 10-calowy ekran dotykowy z systemem operacyjnym Android 8.0. Na systemie operacyjnym została zainstalowana aplikacja WAGO WebVisu ze skonfigurowanym automatycznym połączeniem do wizualizacji.

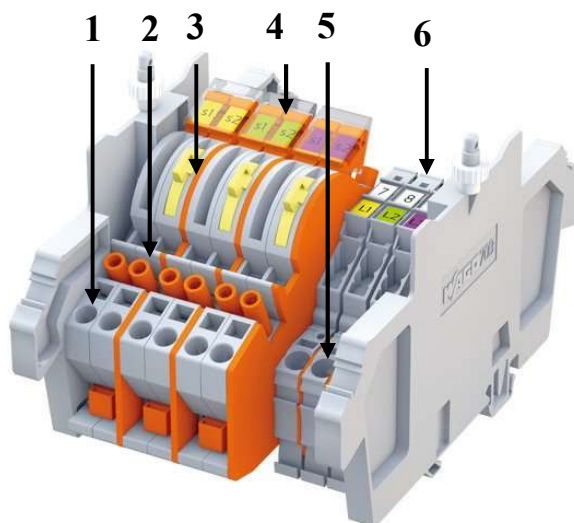
Panel został zamontowany na wysięgniku dla uproszczenia pracy operatora. Mocowanie panelu umożliwia jego prosty demontaż w przypadku prac remontowych w rozdzielni w celu uniknięcia jego uszkodzenia. Ekran HMI łączy się ze sterownikiem przy pomocy bezprzewodowej sieci WLAN. Koncepcja tego typu zwiększa mobilność zastosowanego rozwiązania. Widok panelu operatora przedstawiono na rysunku 2

²⁴ Zdalna konfiguracja jest możliwa po wcześniejszej konfiguracji karty SIM i nawiązaniu transmisji.

²⁵ Źródło: opracowanie własne

3.1.5. Listwa kontrolno-pomiarowa

Listwy kontrolno-pomiarowe przeznaczone są w obwodach rozliczeniowych EE. Głównym zastosowaniem listew jest łączenie obwodów wtórnych przekładników prądowych i napięciowych z obwodami prądowymi i napięciowymi układów pomiarowych. Zaletą ich stosowania jest zapewnienie bezprzerwowej dostawy energii elektrycznej do odbiorów w przypadku np. pomiarów okresowych lub wymiany urządzeń pomiarowych.



Rys. 3.5 Listwa kontrolno-pomiarowa 847-998²⁶

Wykaz oznaczeń z Rys. 3.5:

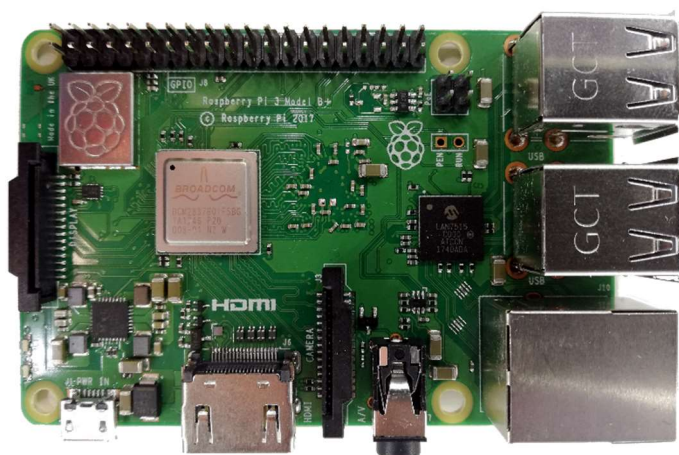
- 1) zaciski przewodów toru prądowego,
- 2) złącze pomiarowe toru prądowego,
- 3) blokada łączy toru prądowego,
- 4) łącznik toru prądowego,
- 5) zaciski przewodów napięciowych,
- 6) łącznik przewodów napięciowych wraz z wkładką bezpiecznikową,

3.1.6. Komputer jednopłytkowy SBC

Jako platformę do implementacji opracowanych algorytmów na poczet urządzenia do bilansowania użytkownika końcowego i prosumenta wykorzystano popularny komputer jednopłytkowy od Raspberry Pi Foundation. Oczywiście na rynku istnieje wiele układów tego typu, niemniej jednak zdecydowano się na wykorzystanie Raspberry Pi z uwagi na rozbudowaną społeczność wielbicieli tego rozwiązania. Jako narzędzie pracy wykorzystano model Raspberry Pi 3 Model B+ (Rys. 3.6). Jest to mikrokomputer o wymiarach 85,6×56×21 mm. Na płycie znajduje się układ SoC Broadcom BCM2837B0 z czterema 64-bitowymi rdzeniami ARM Cortex-A53,

²⁶ Źródło: opracowanie własne na podstawie noty katalogowej

o taktowaniu 1,2 GHz. Posiada również zintegrowany układ graficzny o taktowaniu zegarem 400 MHz. Ponadto minikomputer posiada wbudowane 1 GB pamięci RAM LPDDR2 taktowanej 900 MHz oraz układ radiowy Cypress CYW43455, który zapewnia dwupasmowe (2,4 GHz / 5 GHz) Wi-Fi 802.11n/ac, a także Bluetooth 4.2/LE. Płytkę została wyposażona w kontroler USB/Gigabit Ethernet (100/1000 Mbit/s) SMSC LAN7515 z funkcją PoE (*ang. Power-over-Ethernet*), lecz wymaga on dodatkowej przejściówki do zasilania się. Układ posiada również 4 porty USB 2.0. oraz złącze HDMI. Standardowo minikomputer Raspberry wyposażony jest w złącze kamery CSI, szeregowy interfejs pod wyświetlacz DSI, złącze zasilania microUSB. Oczywiście układ tak jak wersje wcześniejsze posiada 40-pinowe złącze GPIO. Pamięć masowa urządzenia to zewnętrzna karta pamięci microSD.



Rys. 3.6 Raspberry Pi 3B+²⁷

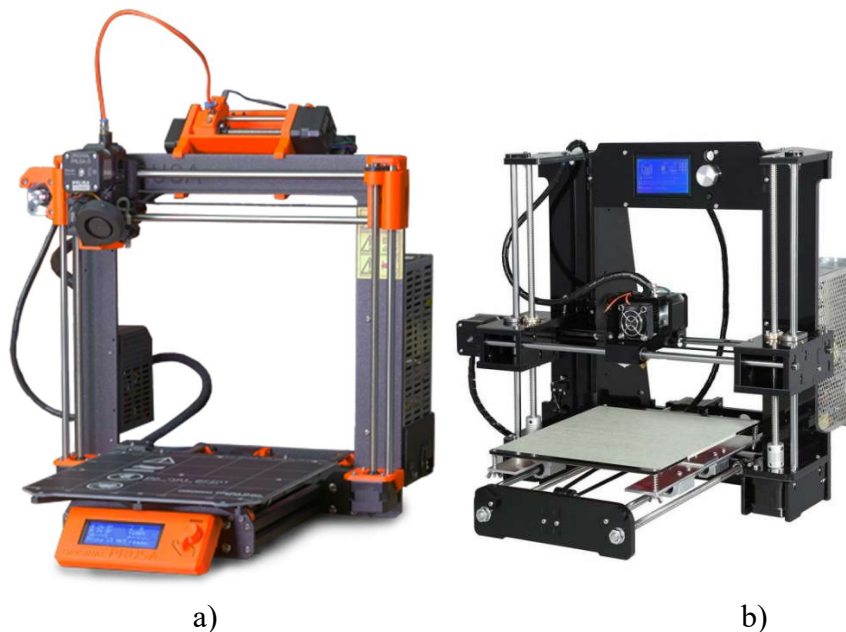
Układ mikrokomputera jest zarządzany z poziomu dedykowanego systemu operacyjnego Linux raspberrypi w wersji 4.19.58-v7+. Bazą systemu operacyjnego Raspbian jest stabilny i wolny system operacyjny Debian. Na dzień tworzenia urządzenia do bilansowania korzystano z wersji 10.0 (Buster).

3.1.7. Drukarka 3D

Do prototypowania urządzenia wykorzystano m.in. drukarkę 3D w technologii FDM (*ang. Fused Deposition Modeling*), w której wykorzystuje się filament w postaci termoplastu dostarczonego do głowicy drukującej w postaci żyłki. Druk odbywa się w sferycznym układzie współrzędnych. W pierwszej kolejności nakładana jest warstwa w osiach XY, a po jej zakończeniu głowica lub stół (zależne od drukarki) przemieszcza się w osi z i nakładana jest kolejna warstwa. Zaprojektowany model obudowy i łączny dla układu scalonego był w pierwszej wersji prototypowany na drukarce Anet A6, lecz docelowy model do testów w rzeczywistym środowisku został wydrukowany na

²⁷ Źródło: opracowanie własne

drukarce Prusa i3 MK3S (Rys. 3.7). Pole robocze drukarki Prusa wynosi 250 x 210 x 210 mm. Drukarka posiada dyszę 0,4 mm do filamentu 1,75 mm. Minimalna wysokość warstw to 0,05 mm. Dodatkowymi funkcjami ułatwiającymi druk są: automatyczne poziomowanie podgrzanego stołu z kompensacją chłodnych rogów, czujnik filamentu, wykrywanie braku zasilania i automatyczne wznowianie druku, wymienna magnetyczna powierzchnia druku. Drukarka obsługuje następujące materiały: PLA, ABS, PET, HIPS, Flex PP, Ninjaflex, Laywood, Laybrick, Nylon, Bamboofill, Bronzefill, ASA, T-Glase, filamenty wzmacniane włóknami węglowymi, poliwęglany.



Rys. 3.7 Drukarki 3D w technologii FDM: a) Prusa i3 MK3S, b) Anet A6²⁸

3.1.8. Kamera termowizyjna

Przed umieszczeniem prototypu w rozdzielni lub rozdzielnicy należało sprawdzić czy opracowane narzędzie kontrolno-pomiarowe po zamknięciu w obudowę nie stwarza zagrożenia i czy nie spowoduje zadziałania zabezpieczeń termicznych szafek pomiarowych do testów termicznych wykorzystano kamerę termowizyjną CompactPro (Rys. 3.8). Kamera termowizyjna posiada czujnik temperatury 320 x 240 pikseli oraz pole 32 stopni. Kamera zawiera automatyczne sterowanie i ustawienie współczynnika emisyjności. Kompaktowość kamery pozwala na wykorzystanie jej praktycznie w większości zastosowań, począwszy od badania szczelności okien czy rurociągów do badania układów elektronicznych i ciemniach, ponieważ urządzenie działa również w całkowitej ciemności.

²⁸ Źródło: opracowanie własne



Rys. 3.8 Kamera Seek Thermal Pro²⁹

Kamera posiada korzystny stosunek ceny do jakości, najczęściej wykorzystywana jest przez majsterkowiczów, inspektorów i rzeczoznawców. Parametry kamery przedstawiają się następująco: pole widzenia (FOV) 32°, maksymalny współczynnik emisyjności 0.97, minimalny współczynnik emisyjności 0.30, matryca bolometryczna 320 x 240 px, częstotliwość odświeżania 15 Hz, minimalna odległość ogniskowania 15 cm, czułość termiczna 75 mK. Zakres pomiarowy temperatury, w którym możemy przeprowadzić pomiar wynosi od -40 do +330 °C

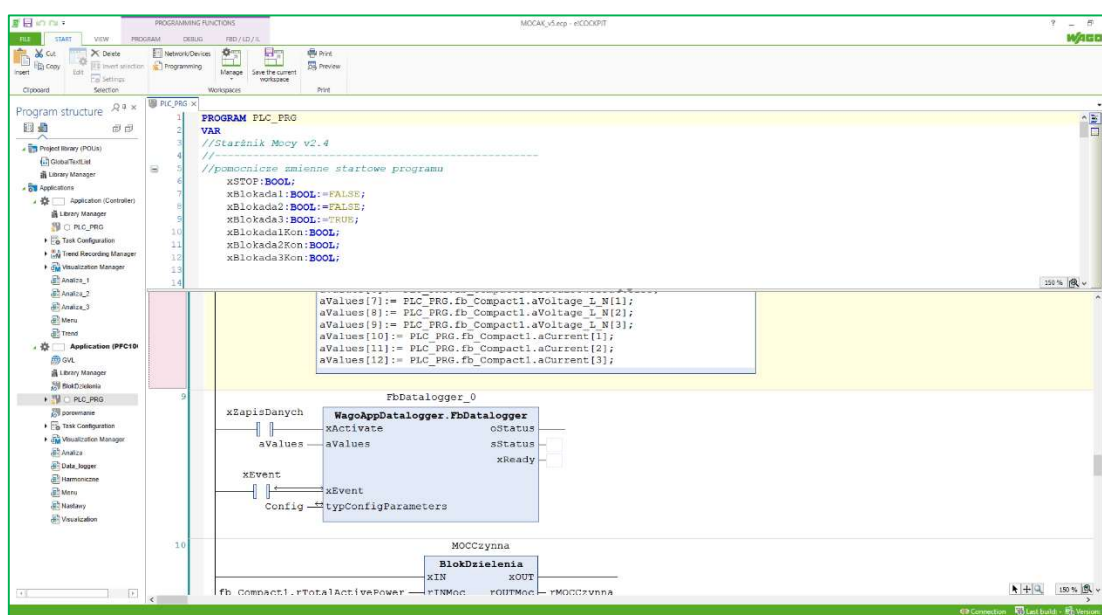
3.2. Oprogramowanie

Poniżej opisano programy wykorzystane na etapie projektowania i tworzenia schematów elektrycznych, mechanicznych oraz tworzenia algorytmów i oprogramowania. Wykorzystane oprogramowanie można podzielić na komercyjne rozwiązania, podlegające płatnej licencji oraz rozwiązania typu *open source*. W przypadku, gdy do budowy urządzenia wykorzystano moduły PLC gotowe z rynku wymagało to od autora posiadania stosownych licencji komercyjnych niezbędnych do tworzenia oprogramowania. Natomiast prototyp narzędzia do bilansowania został oprogramowany w narzędziach typu *open source*. Przewagą narzędzi otwartych jest przede wszystkim bezpieczeństwo oraz stabilna praca. Wynika to z faktu, że kodem źródłowym zajmuje się wielu programistów jednocześnie, co pozwala na szybką eliminację błędów. Kolejną ważną cechą jest możliwość modyfikacji kodu i dostosowania go do własnych potrzeb. Umożliwia to łatwą implementację i prototypowanie, ponieważ możemy wykorzystać istniejące rozwiązanie oraz modyfikować je do własnych potrzeb. Warto wspomnieć, że w wielu przypadkach otwarte narzędzia skracają czas wdrożenia narzędzi i koszty z tym związane, ponieważ można zastosować istniejące niezależne komponenty i po modyfikacji wykorzystać jako jedno narzędzie. Jednocześnie należy zaznaczyć, że sukces wdrożenia i jego czas jest silnie zależny od zaawansowanej wiedzy programistycznej i doświadczenia.

²⁹ Źródło: opracowanie własne

3.2.1. e!COCKPIT

Oprogramowanie e!COCKPIT (Rys. 3.9) jest zintegrowanym środowiskiem programistycznym, wspierającym działania związane z automatyzacją począwszy od konfiguracji sprzętu, programowania, symulacji i wizualizacji, aż po rozruch w jednym programie. Narzędzie stwarza szerokie możliwości wyboru metody programowania sterowników, ponieważ wspiera języki programowania zgodne z IEC 61131-3: schemat drabinkowy (LD), lista instrukcji (IL), język bloków funkcyjnych (FBD), tekst strukturalny (ST), język bloków sekwencyjnych (SFC), język schematu ciągłego (CFC). Ponadto oprogramowanie umożliwia łącznie różnych języków programowania. Tworzone części programu mogą być symulowane i debugowane na maszynie programisty. Dostępna jest również możliwość programowania zorientowanego obiektowo. Zaimplementowane aplikacje pozwalają na tworzenie nowoczesnych interfejsów graficznych z wykorzystaniem funkcji *Drag & Drop*, a interfejs edytora umożliwia dostęp do zmiennych programu IEC. Podobnie jak w przypadku tworzonych algorytmów mamy możliwość prowadzenia symulacji. e!COCKPIT wspiera aktualne standardy HTML5 lub CSS. Co ważniejsze, środowisko posiada rozbudowane funkcje diagnostyczne zarówno w takcie pracy „offline” oraz podczas rozruchów maszyn. Komunikaty i statusy możemy dzielić i wyświetlać w osobnych oknach co znacząco przyspiesza lokalizację ewentualnych błędów.

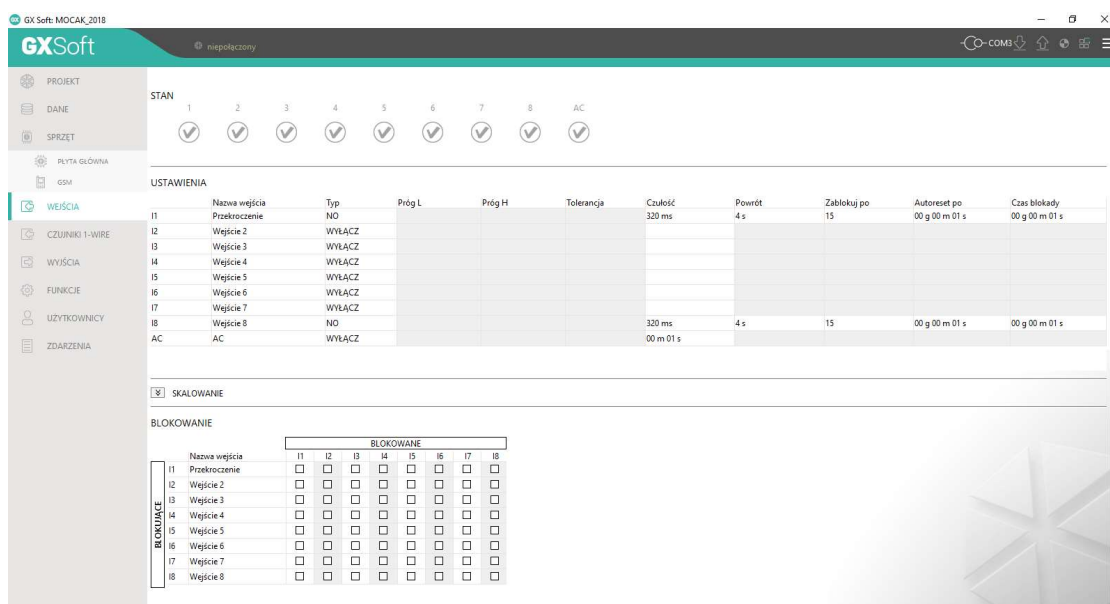


Rys. 3.9 Widok okna oprogramowania e!COCKPIT³⁰

Oprogramowanie e!COCKPIT wykorzystano w wersji 1.4.0.29 do wytworzenia algorytmów sterownia rozdzielnią główną Muzeum Sztuki Współczesnej w Krakowie MOC AK oraz do integracji z automatyką budynkową w technologii LON. Warto dodać, że oprogramowanie bazuje na systemie CODESYS 3.5.

3.2.2. GX Soft

Aplikacja GX Soft służy do programowania modułów komunikacyjnych GSM-X LTE, GSM-X, GPRS-A LTE i GPRS-A firmy Satel. W zależności od modułu komunikacyjnego dostępne są różne możliwości komunikacji. Do wyboru mamy media przodowe (mini-USB, RS-232 (TTL)), zdalną transmisją poprzez sieć Internet. W przypadku połączenia zdalnego aplikacja za pomocą kilku kliknięć nawiązuje bezpieczne połączenie z serwerem SATEL. Narzędzie współpracuje z systemem operacyjnym Windows VISTA/7/8/10. Dodatkowo aplikacja udostępnia wizualizację stanu modułu oraz pamięć zarejestrowanych zdarzeń, przykład na Rys. 3.10.



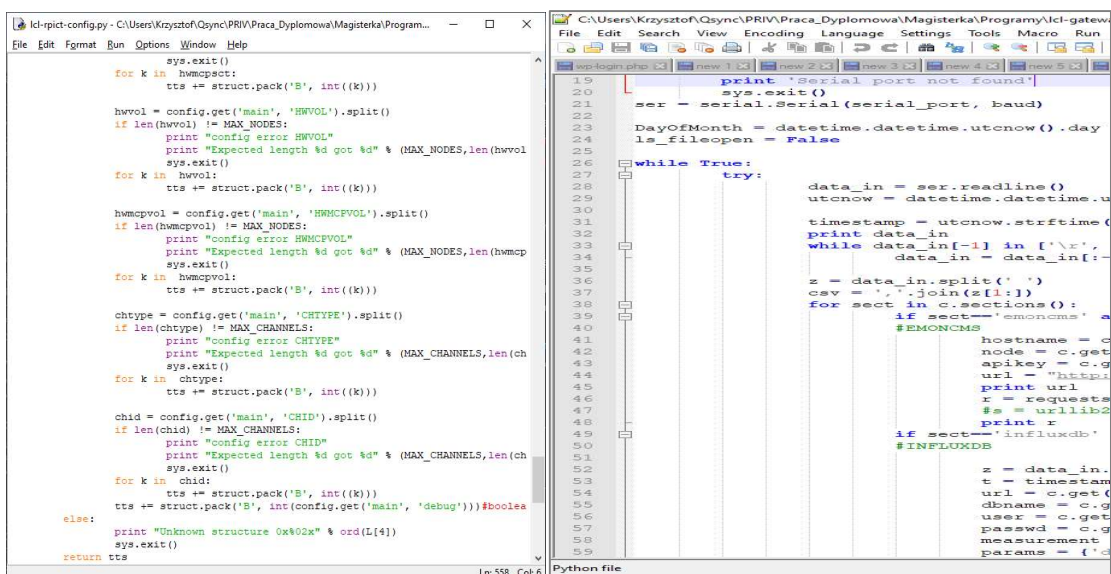
Rys. 3.10 Okno aplikacji aplikacji GX Soft³¹

3.2.3. Interpreter Python

Do realizacji prototypu posłużył natywny interpreter Python-a. Środowisko IDLE (*ang. Integrated DeveLopment Environment* lub *Integrated Development and Learning Environment*) jest zintegrowanym interpreterem w 100% napisanym językiem Python przy wykorzystaniu graficznego zestawu narzędzi Thinker.

Okno dialogowe powłoki udostępnia narzędzia kolorowania składni, danych wejściowych, wyjściowych i błędów. Obsługuje również wcięcia podpowiedzi, automatyczne uzupełniania, przeszukiwanie wielu okien. Istotnym dla twórcy oprymowania jest wbudowany debugger obsługujący kroki i trwałe przerwania. W przypadku długotrwałego korzystania z oprogramowania wymaga ono doinstalowania dodatkowych narzędzi.

³¹ Źródło: opracowanie własne



a)

b)

Rys. 3.11 Okna środowiska wykorzystanych do kodu Python:
a) IDLE, b) Notepad++³²

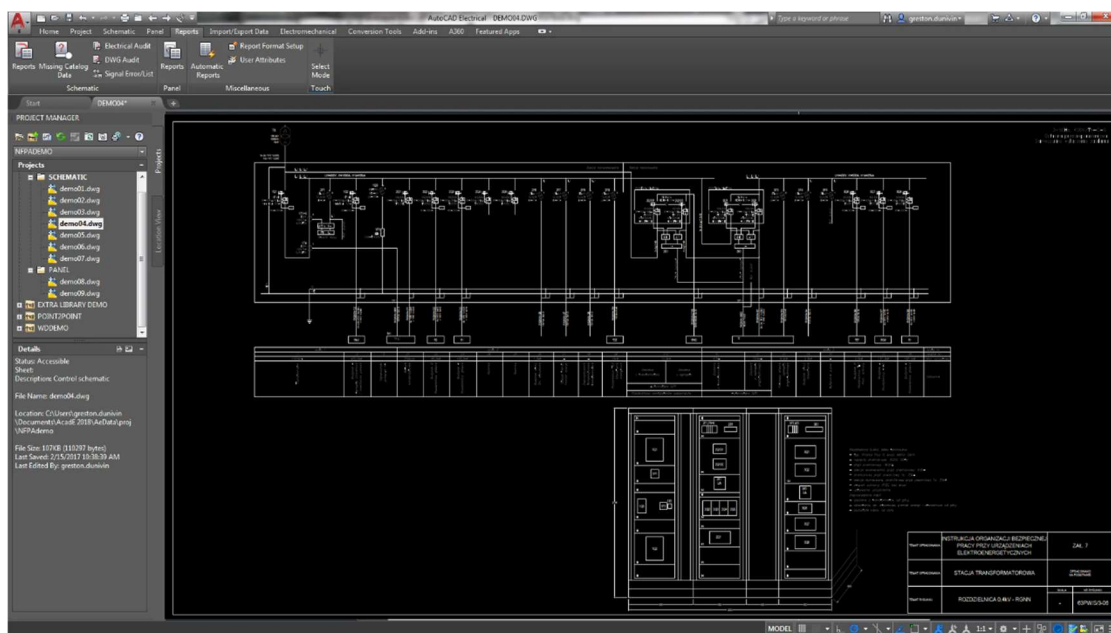
Dodatkowo do analizy składni kodu i poszukiwania błędów skorzystano z zaawansowanego edytora tekstu Notepad++ (Rys. 3.11, b)). Program został wykorzystany z uwagi na jego uniwersalność i możliwość podświetlenia składni wielu języków programowania m.in. C, C++, C#, XML, HTML, TeX, makefile, ASCII-Art, również skryptów powłok uniksowych BAT, SQL, Objective-C, Python, Lua, Ruby, VHDL oraz wielu innych³³.

Przyczyną ostatecznego wyboru środowisk przedstawionych powyżej była konieczność częstego przenoszenia programów pomiędzy maszyną inżynierską, a prototyp. Oprogramowanie musiało cechować się możliwością uruchomienia na odmiennych systemach operacyjnych oraz ograniczonym zużyciem zasobów sprzętowych (Raspberry Pi).

³² Źródło: opracowanie własne
³³

3.2.4. AutoCAD LT

AutoCAD LT® jest typowym oprogramowaniem CAD (*ang. Computer Aided Design*) służącym do projektowania komputerowego wykorzystywanego przez inżynierów, architektów oraz specjalistów z branży budowlanej. Ma zastosowanie głównie przy tworzeniu dokumentacji i rysunków 2D. W niniejszej pracy program został wykorzystany do przygotowania nowej dokumentacji technicznej dla rozdzielni MOCAK, po zainstalowaniu opracowanego narzędzia SM.



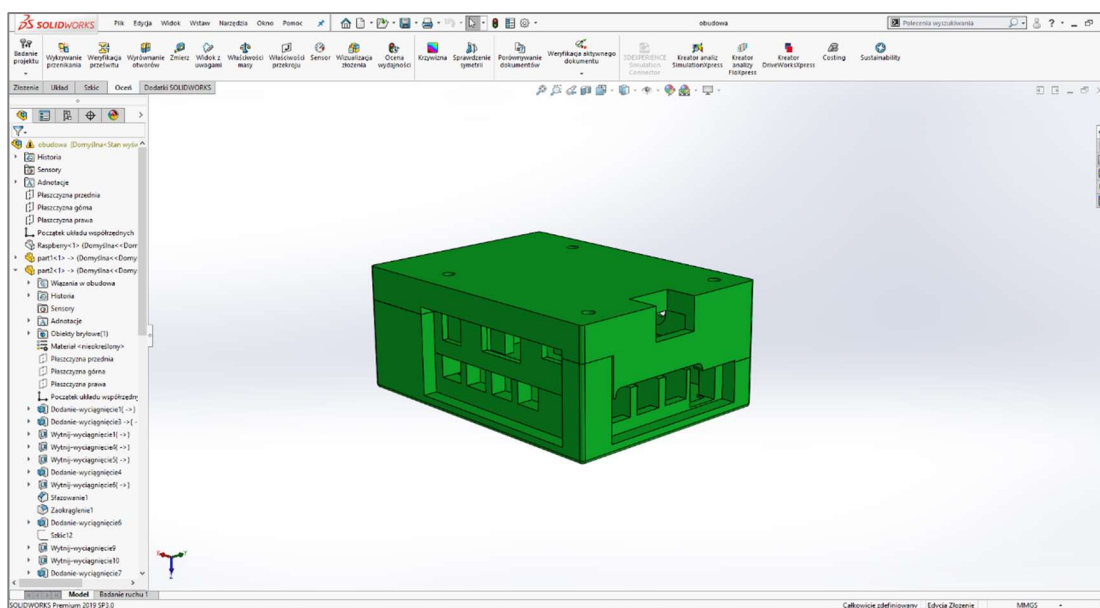
Rys. 3.12 Okno programu AutoCAD LT³⁴

Program został zastosowany, ponieważ dostępna licencja zawierała pakiet branży elektrycznej, co w znacznym stopniu skróciło czas tworzenia dokumentacji powykonawczej dla rozdzielni (Rys. 3.12).

3.2.5. SOLIDWORKS

SOLIDWORKS® Premium jest platformą o bardzo szerokim zakresie zastosowań do modelowania trójwymiarowego 3D. Oprócz modelowania brył, czy konstrukcji, umożliwia złożenia wszystkich elementów w jeden obiekt. Pakiet premium zawiera między innymi narzędzia do symulacji ruchu, obliczeń wytrzymałościowych, liniową analizę statyczną części i złożeń. SOLIDWORKS dopuszcza również skorzystanie z szerokiego asortymentu pakietów dodatkowych, dzięki którym można np. zaprojektować i dołączyć do projektu układy płytek drukowanych do modelu 3D (Rys. 3.13).

³⁴ Źródło: opracowanie własne



Rys. 3.13 Okno środowiska SOLIDWORKS Premium³⁵

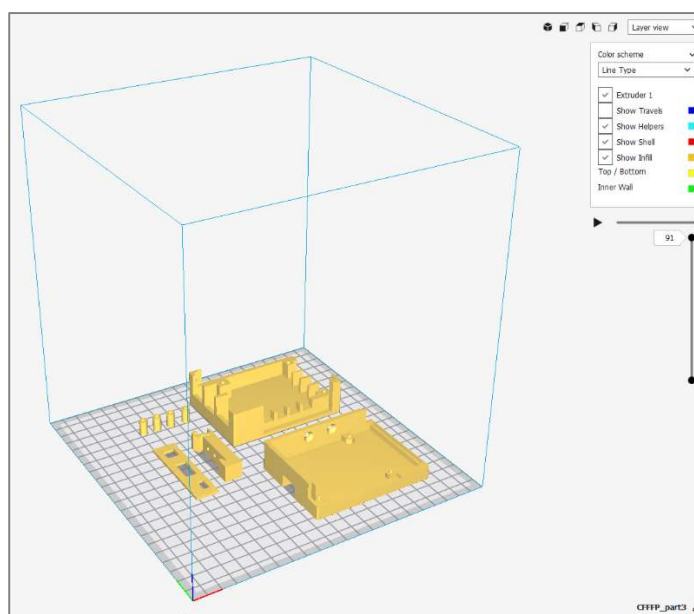
Złożoność i ilość narzędzi platformy SOLIDWORKS jest tak duża, że mogłaby stanowić przedmiot oddzielnego opracowania i nie sposób przedstawić jego funkcjonalności krótkim opisem.

3.2.6. Ultimaker Cura

Program służy do przygotowywania modelu do druku 3D. Jest to jeden z najpopularniejszych slicerów³⁶ dla drukarek 3D typu FDM. Obecnie program należy do firmy Ultimaker, lecz nadal jest rozwijany w postaci oprogramowania typu *open source*. Narzędzie obsługuje większość dostępnych na rynku drukarek niskobudżetowych i umożliwia definiowanie własnych parametrów i profili druku. W dostępnych ustawieniach możemy wybrać np. typ podpór, grubości drukowanych warstw, wielkości podstawek, stopień wypełnienia ścianek, typ struktury. Ponadto parametry te mogą być ustawione niezależnie dla powierzchni drukowanej. Oprogramowanie daje możliwość naprawy nieznacznych błędów przy projektowaniu oraz przeprowadzanie symulacji druku przykładowe okno z wykorzystania programu przedstawia Rys. 3.14.

³⁵ Źródło: opracowanie własne

³⁶ Slicer jest programem służącym do dzielenia modelu na warstwy i generowania kodu (G-code) zrozumiałego przez sterownik drukarki.



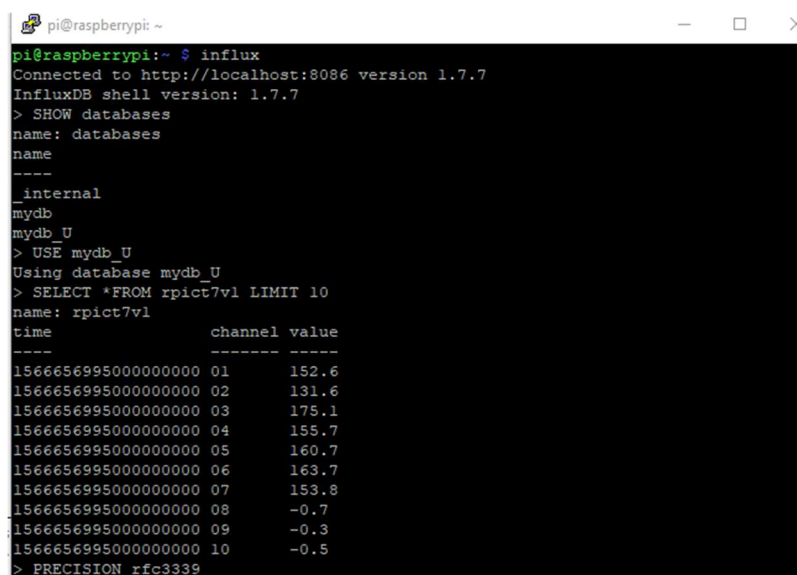
Rys. 3.14 Okno programu Ultimaker Cura³⁷

3.2.7. InfluxDB

Jako repozytorium danych pomiarowych wykorzystano opracowaną przez Influxdata® bazę danych dla szeregów czasowych o nazwie InfluxDB. Baza danych jest względnie nowa, ponieważ została wydana 6 lat temu (24 września 2013 r.). Baza umożliwia przetwarzanie z wysoką wydajnością, zapewnia kompresję danych pomiarowych oraz zapytania w czasie rzeczywistym. InfluxDB w całości jest napisany w języku Go³⁸, a jego kompilacja odbywa się do jednego pliku binarnego. Posiada wbudowane API HTTP oraz wiersz poleceń co zapewnia możliwość zapisu i zapytań. Zapytania do bazy wykonuje się językiem InfluxQL, składnia jest on zbliżony do popularnego SQL. Twórcy zadbali o to, aby język był intuicyjny, a także umożliwił szybkie wdrożenie dla znających SQL. InfluxQL obsługuje wyrażenia regularne, wyrażenia arytmetyczne i funkcje specyficzne dla szeregów czasowych w celu przyspieszenia przetwarzania danych. W założeniu twórców baza jest dedykowana do milionów punktów pomiarowych, dlatego zapewnia automatyczne agregowanie danych w celu uniknięcia problemów z pamięcią masową. Oprogramowanie umożliwia nam zmniejszenie próbek danych tak, aby nie przechowywać w pamięci całości surowych danych, a czas przetrzymywania danych zagregowanych może zostać skonfigurowany. InfluxDB ma zaimplementowane zapytania typu *Continuous Queries* i *Retention Policies* (zapytania ciągłe i zasady przechowywania).

³⁷ Źródło: opracowanie własne

³⁸ Język programowania zaprezentowany w 2009 r. przez firmę Google, jest on wieloparadygmatawy i wieloplatformowy, łączy w sobie łatwość pisania oraz wydajność.



```

pi@raspberrypi:~$ influx
Connected to http://localhost:8086 version 1.7.7
InfluxDB shell version: 1.7.7
> SHOW databases
name: databases
name
----
 _internal
mydb
mydb U
> USE mydb U
Using database mydb_U
> SELECT *FROM rpict7v1 LIMIT 10
name: rpict7v1
time                channel value
----                -
1566656995000000000 01      152.6
1566656995000000000 02      131.6
1566656995000000000 03      175.1
1566656995000000000 04      155.7
1566656995000000000 05      160.7
1566656995000000000 06      163.7
1566656995000000000 07      153.8
1566656995000000000 08      -0.7
1566656995000000000 09      -0.3
1566656995000000000 10      -0.5
> PRECISION rfc3339

```

Rys. 3.15 Okno wiersza poleceń InfluxDB³⁹

Wybrano bazę danych typu *time series* z uwagi na możliwość automatyzacji agregatów przechowywanych próbek pomiarowych, co znacząco zmniejsza wolumen przechowywanych danych. Co więcej InfluxDB jest rozwiązaniem *open source*, więc dopuszcza możliwość modyfikacji pod dedykowane rozwiązanie. W projekcie rozważono dwie wersje instalacji na serwerze oraz lokalnie na urządzeniu SM. Na Rys. 3.15 przedstawiono przykładowe zapytania do bazy danych poprzez terminal i SSH.

3.2.8. Grafana

Grafana podobnie jak InfluxDB została napisana w Go oraz należy do zbioru otwartego oprogramowania typu *open source*. Oprogramowanie dedykowane jest do analizy i wizualizacji danych pomiarowych. Narzędzie posiada wbudowane drivery do ponad 30 źródeł danych to m.in.: AWS CloudWatch, Azure Monitor, Elasticsearch, Google Stackdriver, Graphite, InfluxDB, Loki, Microsoft SQL Server (MSSQL), Mixed, MySQL, OpenTSDB, PostgreSQL, Prometheus, Testdata. Ilość obsługiwanych baz świadczy o dużym potencjale tego rozwiązania. Dodatkowo od wersji 3.0 istnieje możliwość instalowania dodatkowych plugin-ów, a ich ilość jest ogromna i w tym przypadku określenie to nie jest przesadne. W narzędziach Grafany możemy tworzyć szybkie i elastyczne wykresy do analityki, począwszy od histogramów, map cieplnych do map geodezyjnych. Stworzone kokpity i możemy poddać dynamice, zmiana na jednym z wykresów jest aktualizowana na pozostałych. Do analizy można wykorzystać również zapytania typu ad-hoc. Operator ma dostęp do dzienników zdarzeń oraz alarmowania.

³⁹ Źródło: opracowanie własne



Rys. 3.16 Okno Grafana w przeglądarce Internetowej⁴⁰

w narzędziu do bilansowania prosumenta skorzystano z Grafana przede wszystkim z uwagi na: otwartość rozwiązania, możliwość integracji danych pochodzących z różnych źródeł, wprowadzenie poziomu uprawnień, wymianę opracowanych pulpitów z innymi użytkownikami oraz prosty export np. do Excel-a. Przykładowy pulpit wizualizacyjny z wykresem przedstawia Rys. 3.16.

⁴⁰ Źródło: opracowanie własne

4. Projekt systemu

Rozdział przedstawia opis prac związanych z wykonaniem projektu urządzeń typu *smart meter*. W trakcie realizacji niniejszej pracy dyplomowej zakładano kilka różnych koncepcji opracowania urządzenia. Jednym z podstawowych założeń przyjętych przez autora przytoczonego opracowania było praktycznie zastosowanie i wdrożenie opracowanego urządzenia. Pierwszy koncept to inteligentne narzędzie do przeprowadzania audytów energetycznych wraz możliwością kontroli odbiorów i magazynów oraz prowadzenia bilansów na dużych obiektach przemysłowych. Jednakowoż w trakcie analiz literaturowych i uwarunkowań u partnera przemysłowego okazało się, że urządzenie musi posiadać odpowiednie atesty aby mogło być ulokowane wewnątrz rozdzielni. Dodatkowo pojawił się problem ruchowy funkcjonalności bilansowania, ponieważ wymagałby on modyfikacji struktury SEE na obiekcie MOCÁK. z uwagi na fakt, że tradycyjny SEE zakłada przepływ mocy w jednym kierunku zarówno w KSEE jak i lokalnie na obiektach. Obostrzenia eksploatacyjne i ruchowe narzucone na MOCÁK nie umożliwiały realizacji założonych w pracy celów. Partner przemysłowy nadal był zainteresowany współpracą, ponieważ ponosił on duże koszty za przekroczenia mocy.

Równolegle autor przedmiotowej pracy magisterskiej brał udział w tworzeniu strategii i analiz dla Klastrow Energii. Opracowane przez twórcę strategie zostały wyróżnione w konkursach na Certyfikaty Pilotażowych Klastrow Energii, które były ogłaszane przez Ministerstwo Energii na. W trakcie prac związanych ze wsparciem KE wielokrotnie przejawiał się problem pomiarów rozproszonych odnawialnych źródeł energii. Po przeanalizowaniu dostępnych na rynku urządzeń stwierdzono możliwości techniczne przygotowania rozproszonego systemu pomiarowego, lecz przyjęte na rynku rozwiązania miały kilka wad z punktu widzenia potencjalnych użytkowników i łatwości wdrożenia. Pierwszą poważną wadą była zamknięta struktura systemów dostępnych na rynku, co determinowało uzależnienie się od jednego dostawcy i utrudniało lub uniemożliwiało integrację z już istniejącymi pomiarami. Rozbudowane systemy do analizy wymagały szkolenia personelu przez ekspertów oraz uniemożliwiały bezpośredni dostęp do danych prosumentom. Część z KE jest zlokalizowana na terenach przemysłowych gdzie pojawiał się problem z przesyłem danych poprzez PLC⁴¹ (*ang. Power line communication*), a część na terenach wiejskich gdzie zawodziła pakietowa transmisja danych poprzez ciecic UMTS i LTE. Mieszkańcy zrzeszeni w ramach KE zgłaszali chęć udostępnienia swoich łączy do transferu danych. Prosumenci wykazywali zainteresowanie rozwiązaniem, które oprócz informacji dla koordynatora klastra będzie w stanie udostępnić do ich wglądu takie dane jak: wolumen produkowanej energii, zużycia i bilans, ale takie urządzenie, które nie wymagałoby poważnych ingerencji w instalację elektryczną budynków.

⁴¹ W pracy skrót PLC jest używany w kontekście dwóch odmiennych znaczeń: 1 – sterownik programowalny, 2 – komunikacja przez sieć elektroenergetyczną. Znaczenie jest interpretowane w kontekście wypowiedzi

Stanowisko uczestników klastrów zostało potwierdzone w trakcie badań prowadzonych w projekcie GOSPOSTRATEG KlastER.

Mając na uwadze chęć realizacji pracy z praktycznym wdrożeniem i współpracą z MOCAK oraz dostrzeżenie potrzeb prosumentów. Zdecydowano się zrealizować oba przedsięwzięcia. Taki stan rzeczy oznaczał budowę dwóch niezależnych „inteligentnych narzędzi”. Dodatkowo w trakcie prac nad urządzeniem dla MOCAK, rozpoczęto działania nad narzędziem do zarządzania rozproszonymi źródłami rezerwowymi do zastosowań ekonomicznych u MŚP w ramach projektu RELFlex.

Rozpoczęcie współpracy w wyżej wymienionych projektach dodatkowo zmotywowało autora pracy dyplomowej do opracowania dwóch niezależnych urządzeń.

- 1) Smart Power Guard – urządzenie dedykowane do dużych odbiorów energii elektrycznej.
- 2) Smart EMS – Smart Energy Control System – narzędzie dla prosumentów oraz audytu energetycznego.

4.1. Smart Power Guard

Urządzenie Smart Power Guard (Smart PG) było dedykowane do pomiaru parametrów wielkości elektrycznych w obiekcie Rozdzielni Głównej Muzeum Sztuki Współczesnej w Krakowie (MOCAK). Zastosowane rozwiązanie miało umożliwiać wizualizację chwilowych parametrów energii elektrycznej zarówno poprzez panel HMI zamontowany w rozdzielni głównej, jak i za pomocą sieci LAN poprzez web server. Ponadto urządzenie miało posiadać możliwość zapisu mierzonych parametrów na nośniku fizycznym w postaci karty pamięci. Lecz w przyszłości dodatkowo mogło się komunikować z zewnętrznym serwerem bazodanowym. Uwzględniając potrzeby partnera MOCAK oraz możliwość przemysłowego zastosowania opracowanego narzędzia w bardzo krótkim czasie. Zdecydowano, że do budowy pierwszego urządzenia będą wykorzystane gotowe dostępne na rynku moduły atestowane do pracy w środowisku przemysłowym. Koncepcja systemu została przedstawiona na Rys. 4.1.

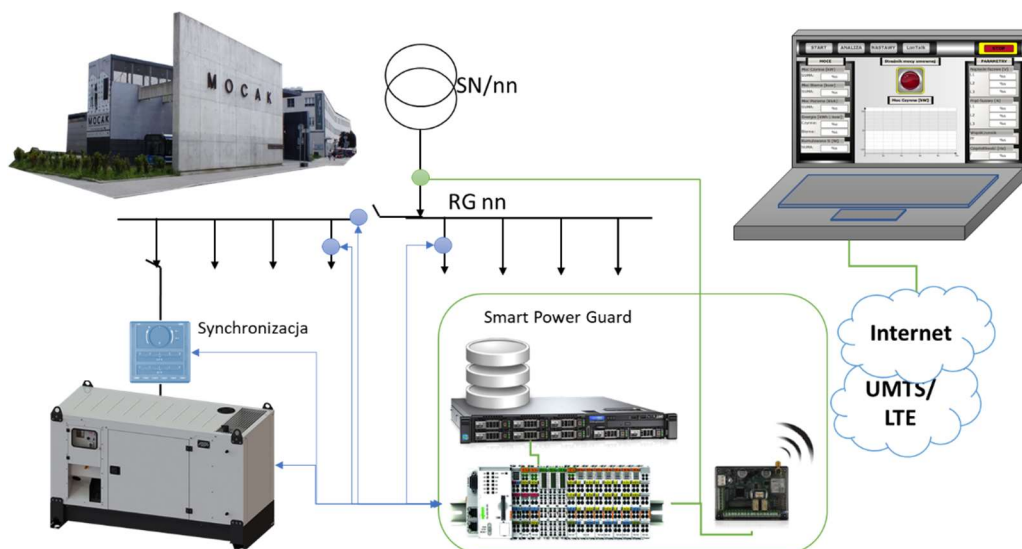
Ponadto projekt zakładał następujące cechy i funkcjonalności:

- praca z instalacją trójfazową czteroprzewodowa o napięciu 400 V.
- kontrolowany odbiór o mocy nominalnej od 200 do 350 kW,
- maksymalna wartość mocy która ma być zachowana 300 kW.
- autonomiczna praca,
- powiadomienia SMS o możliwym przekroczeniu mocy,
- zabezpieczanie przekroczenia mocy chwilowej poprzez:
 - wyłączenie odbiorów,
 - wydzielenie części odbiorów do pracy na agregat,
 - ograniczenie mocy wykorzystując system automatyki budynkowej,

- predykcja przekroczenia mocy, w oknie 15 minutowym,
- obsługa z poziomu panelu operatora oraz zdalna z wykorzystaniem LAN,
- układ przystosowany do pomiarów okresowych rozdzielni,
- ograniczona modyfikacja istniejącej infrastruktury,

Wymagania stawiane przez partnera przemysłowego:

- funkcja analizatora parametrów energii elektrycznej,
- pomiar wartości chwilowych oraz rejestracja przebiegów w czasowych następujących parametrów niezależnie dla każdej fazy:
 - o prąd,
 - o napięcie,
 - o moc czynna,
 - o moc bierna,
 - o moc pozorna,
 - o współczynnik $\cos\phi$,
 - o częstotliwość,
 - o harmoniczne do 40,
 - o dostęp do danych archiwalnych z kilku lat za pomocą sieci LAN i komputera klasy PC,
 - o informowanie o zmianie minimum dwóch wybranych parametrów pod wprowadzony wcześniej zakres.
 - o wywołanie alarmów dla wybranego parametru.
 - o możliwość komunikacji z automatyką budynku przy użyciu LON®



Rys. 4.1 Projekt funkcjonowania Smart PG⁴²

Jako system pomiarowy wykorzystano sterownik PLC WAGO przedstawiony na Rys. 3.1 wraz z modułami kontrolno-pomiarowymi. Wybrano sterownik WAGO z uwagi na dobry stosunek ceny do możliwości oraz możliwość budowania algorytmów łącząc

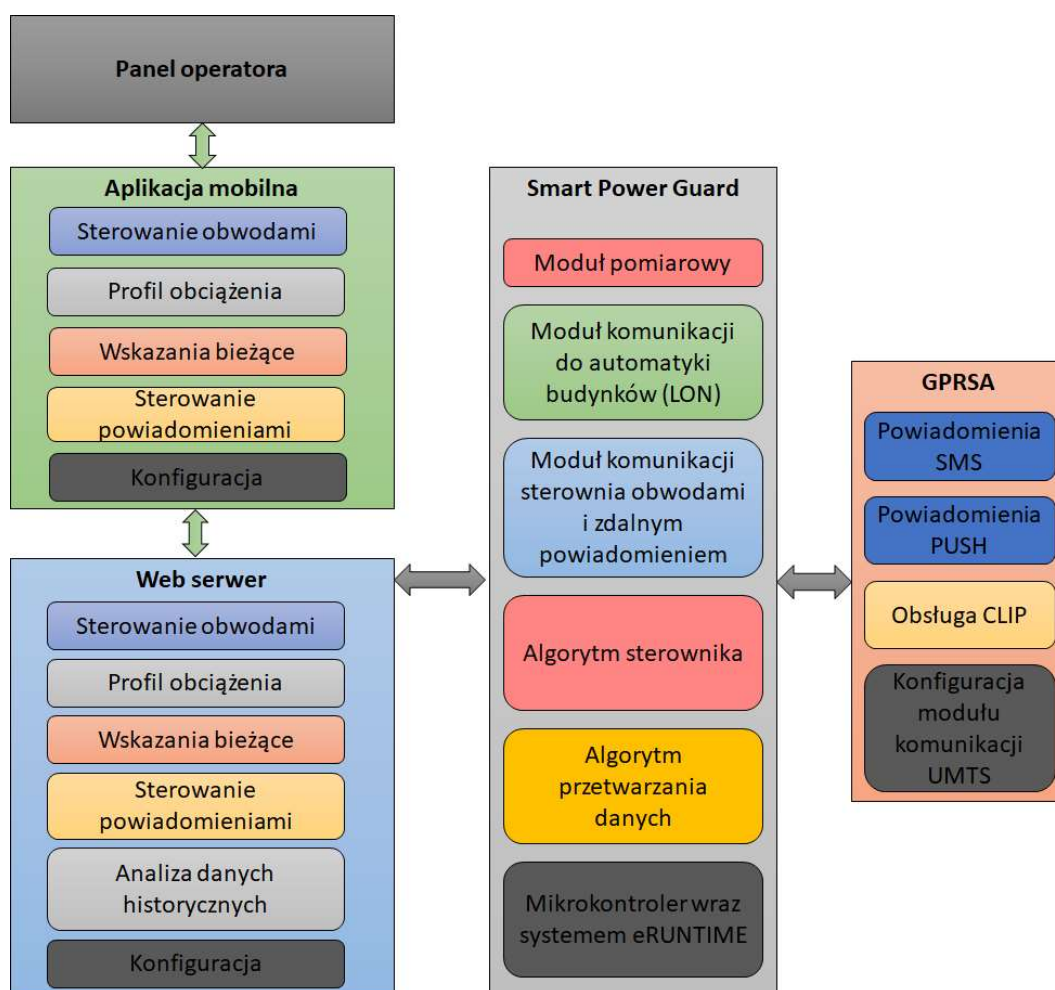
⁴² Źródło: opracowanie własne

różne języki IEC. W tabeli Tab. 4.1 przedstawiono zakładane wartości mierzone. Kolejną funkcją udostępnioną w Sterowniku jest web serwer, co znacząco ułatwiło wykonanie dostępu ze zdalnego komputera PC. Dostępna jest również aplikacja na urządzenia mobilne, co w przypadku operatora nie ogranicza jego pobytu do rozdzielni.

Tab. 4.1 Wykaz mierzonych parametrów elektrycznych

L.p.	Parametr
1	całkowita moc czynna, bierna, pozorna
2	współczynnik mocy PF
3	całkowita energia czynna, bierna, pozorna
4	prąd L1, L2, L3, N
5	napięcie L1-N, L2-N, L3-N
6	moc czynna L1, L2, L3
7	moc bierna L1, L2, L3
8	moc pozorna L1, L2, L3
9	częstotliwość L1, L2, L3
10	cos phi L1, L2, L3
11	współczynnik PF L1, L2, L3
12	kwadranty mocy L1, L2, L3
13	średni prąd L1, L2, L3
14	minimalny prąd L1, L2, L3
15	maksymalny prąd L1, L2, L3
16	wartość szczytowa prądu
17	średnie napięcie L1, L2, L3
18	minimalne napięcie L1, L2, L3
19	maksymalne napięcie L1, L2, L3
20	wartość szczytowa napięcia
21	napięcie wejścia dodatkowego x3
22	minimalna moc czynna L1, L2, L3
23	maksymalna moc czynna L1, L2, L3
24	współczynnik mocy LF L1, L2, L3
25	energia czynna L1, L2, L3
26	energia bierna L1, L2, L3
27	energia pozorna L1, L2, L3
28	kąt fazowy L1, L2, L3
29	maksymalna częstotliwość L1, L2, L3
30	minimalna częstotliwość L1, L2, L3
31	całkowita energia dostarczona
32	całkowita energia oddana

Schemat funkcjonalny przedstawiono na Rys. 4.2. Poszczególne bloki prezentują możliwości danego narzędzia. Jako panel operatora założono, że będzie tablet o przekątnej 10 cali z system operacyjnym Android.8.0. takie rozwiązanie zwiększa mobilność całego systemu.



Rys. 4.2 Schemat aplikacyjny wraz z podziałem na bloki funkcjonalne⁴³

Zastosowanie web serwera oraz aplikacji mobilnej znacząco rozszerzyło możliwości obsługi systemu. Założenie zewnętrznego urządzenia do komunikowania miało na celu uniezależnienie komunikatów alarmowych od poprawnego działania sterownika, zdecydowano się na zastosowanie modułu GPRSA, który oprócz standardowych funkcji powiadomień posiada wykrycie braku zasilania, po czym przesyła informację przed wyłączeniem się.

Schemat blokowy narzędzia Smart PG współpracującego z agregatami rezerwowymi i automatyką BMS przedstawiono na Rys. 4.3.



Rys. 4.3 Schemat blokowy konstrukcji Smart PG⁴⁴

⁴³ Źródło: opracowanie własne

⁴⁴ Źródło: opracowanie własne

4.1.1. Oprogramowanie sterownika

Na prośbę partnera przesyłowego oprogramowanie na sterownik główny rozpoczęto w jednym z języków IEC61131-3, a mianowicie w LD dla lepszego zrozumienia przez zespół techników, którym dysponował. Po przystąpieniu do realizacji złożonych funkcjonalności od razu uwidoczniły się problemy związane z brakiem technicznych możliwości realizacji założonych funkcjonalności przy pomocy zestawu dostępnych bloków funkcyjnych. W celu rozwiązania problemu skorzystano z własności oprogramowania e!COCKPIT bazującego na CODESYS 3.5, a mianowicie możliwości integracji części programu napisanych w różnych językach. Do realizacji niedostępnych za pomocą standardowych funkcji i narzędzi dostępnych w PLC wykorzystano język ST. Ciekawą właściwością programu jest możliwość bezpośredniej implementacji programu do bloku funkcyjnego, dlatego zmieszano 3 języki programowania LD, FBD i ST w realizacji niemiejszego narzędzia.

W ten sposób dokonano podziału algorytmów na tzw. część siłową, która odzwierciedlała narzędzia techniczne oraz część programową, odpowiedzialną za funkcje „smart”..

Kolejno zamieszczono spis zmiennych startowych blokad i definicji bloków.

```
PROGRAM PLC_PRG
VAR
//Smart Power Guard v2.4
//-----
// program startup variables
    xSTOP:BOOL;
    xBlokada1:BOOL:=FALSE;
    xBlokada2:BOOL:=FALSE;
    xBlokada3:BOOL:=TRUE;
    xBlokada1Kon:BOOL;
    xBlokada2Kon:BOOL;
    xBlokada3Kon:BOOL;

// Startup buttons
    xSTOPAwaria:BOOL;
    xTRENDY:BOOL;
    xNASTAWY:BOOL;
    xSTART:BOOL;
    xANALIZA:BOOL;
    xHARMONICZNE: BOOL;
    xHPomiar: BOOL;
    xZapisDanych: BOOL;

// Locks
// Datalogger
//IF xInit THEN
//    xInit:= FALSE;
//    typConfigParameters.bDatalogger_type := 3; //Datalogger format
//    typConfigParameters.sPath := '/media/sd'; // Path to file
//    typConfigParameters.sFilename := 'MyLogfile.csv'; //Filename

//    typConfigParameters.xCyclicLoggin := TRUE; // Cyclic loggin
//    typConfigParameters.bInterval := 2; // Interwal time
```



```

//     typConfigParameters.uiIntervalFactor := 1; //Write every 15 min
// Initialize chanel
//     typConfigParameters.atypChannelConfig[1].xChannelExists :=
TRUE;
//     typConfigParameters.atypChannelConfig[1].sChannelName :=
'Cycle';
//END_IF
//aValues[1]:= TO_REAL(1);

//END_IF
//     FbDatalogger_0: WagoAppDatalogger.FbDatalogger;
//     xActive: BOOL;
//     aValues: ARRAY [1..MAX_CHANNELS] OF REAL;
//     xEvent: BOOL;
//     typConfigParameters:=typConfigParameters,

//-----
// program variables settings
    rMocUmowna:REAL:=280;
    rStopien1:REAL:=150;
    rStopien2:REAL:=250;
    rMocUsredniona15:REAL;
    rlicznikenergii:REAL;

    xPrzycisk:BOOL;
    xLampka:BOOL;
    rF:REAL;
    rP:REAL;
//-----
// part responsible for measuring power and harmonics in the program

    fb_Compact1: FbAC_Compact_495;
    _3_PHASE_POM_690VAC_1A_Compact:
WagoAppPowerMeasurement.WagoTypesModule_75x_49x.I_Module_75x_495;
    //FbAC_Compact_495_0: WagoAppPowerMeasurement.FbAC_Compact_495;
    FbHarmonicValues_495_0:
WagoAppPowerMeasurement.FbHarmonicValues_495;
    FbHarmonicValues_495_1:
WagoAppPowerMeasurement.FbHarmonicValues_495;
    //Config1:
WagoAppPowerMeasurement.FbConfigurationAndStatus_495;
    //typConfiguration:typ495Config;

    rVHarmonic_5: REAL;
    rVHarmonic_7: REAL;
    rVHarmonic_11: REAL;

    rIHarmonic_5: REAL;
    rIHarmonic_7: REAL;
    rIHarmonic_11: REAL;

// -----
//DataLogger
Channell: REAL;
Config: typConfigDatalogger;
FbDatalogger_0: WagoAppDatalogger.FbDatalogger;
aValues: ARRAY [1..MAX_CHANNELS] OF REAL;
xEvent: BOOL;
xInit: BOOL:= TRUE;
//i: DWORD:=0;

```

```
// -----
// Delayed engagement of locks
TON_0: Standard.TON;
TON_1: Standard.TON;
TON_2: Standard.TON;
TON_3: Standard.TON;
TOF_0: Standard.TOF;

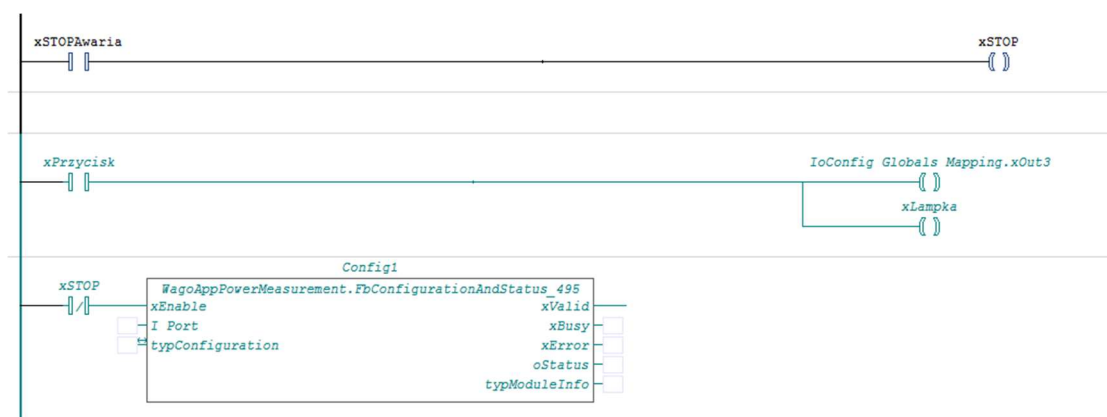
//Comparison of exceedance with the set value
Przekroczenie_3: porownanie;
Przekroczenie_2: porownanie;
Przekroczenie_1: porownanie;

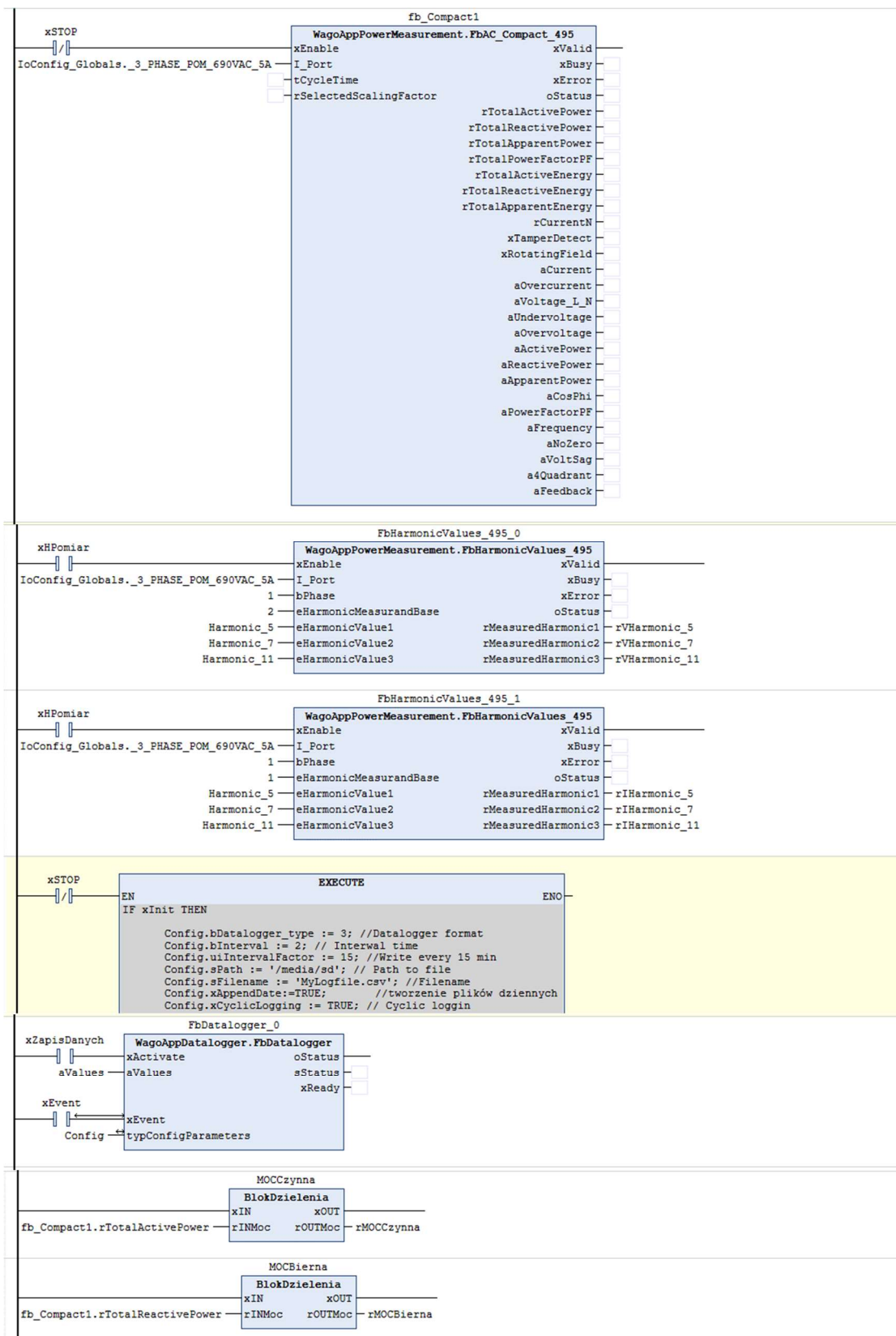
// -----
// Conversion and division values
MOCCzynna: BlokDzielenia;
rMOCCzynna: REAL;
MOCBierna: BlokDzielenia;
rMOCBierna: REAL;
MOCPozorna: BlokDzielenia;
rMOCPozorna: REAL;
ENERGIACzynna: BlokDzielenia;
rENERGIACzynna: REAL;
ENERGIABierna: BlokDzielenia;
rENERGIABierna: REAL;

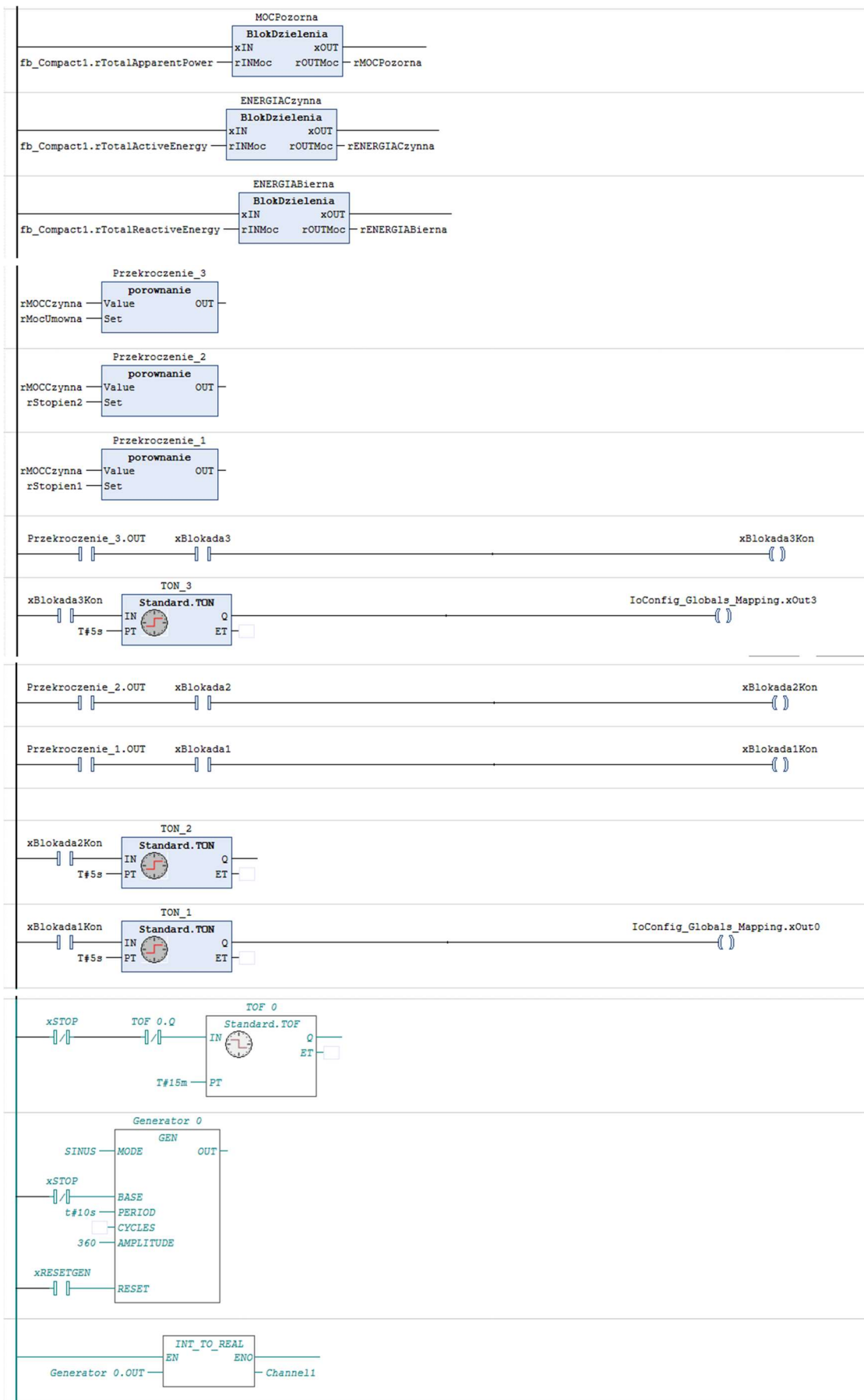
//-----
// Block - verification
// Auxiliary generator for checking measured values
Generator_0: GEN;
SINUS: GEN_MODE;
xRESETGEN: BOOL;

END_VAR
```

Następnie zamieszczono część z bloków łączoną LD i FBD, blok execute o długim zapisie jest domyślnie skraccany. Kod bloku został zamieszczony poniżej programu PLC_PRG.







Kod konfiguracji dla modułu pomiaru mocy.

```
IF xInit THEN

    Config.bDatalogger_type := 3; //Datalogger format
    Config.bInterval := 3; // Interwał time
    Config.uiIntervalFactor := 3; //Write every 15 min
    Config.sPath := '/media/sd'; // Path to file
    Config.sFilename := 'MyLogfile.csv'; //Filename
    Config.xAppendDate:=TRUE; //tworzenie plików dziennych
    Config.xCyclicLogging := TRUE; // Cyclic logging

    Config.uiIntervalFactor := 1; //Write every 15 min
// Initialize chanel
    Config.atypChannelConfig[1].xChannelExists := TRUE;
    Config.atypChannelConfig[1].bDecimalPlaces:=2;
    Config.atypChannelConfig[1].sUnit:='kW';
    Config.atypChannelConfig[1].sChannelName := 'P';

    Config.atypChannelConfig[2].xChannelExists := TRUE;
    Config.atypChannelConfig[2].bDecimalPlaces:=2;
    Config.atypChannelConfig[2].sUnit:='kvar';
    Config.atypChannelConfig[2].sChannelName := 'Q';

    Config.atypChannelConfig[3].xChannelExists := TRUE;
    Config.atypChannelConfig[3].bDecimalPlaces:=2;
    Config.atypChannelConfig[3].sUnit:='kVA';
    Config.atypChannelConfig[3].sChannelName := 'S';

    Config.atypChannelConfig[4].xChannelExists := TRUE;
    Config.atypChannelConfig[4].bDecimalPlaces:=2;
    Config.atypChannelConfig[4].sUnit:='kWh';
    Config.atypChannelConfig[4].sChannelName := 'EP';

    Config.atypChannelConfig[5].xChannelExists := TRUE;
    Config.atypChannelConfig[5].bDecimalPlaces:=2;
    Config.atypChannelConfig[5].sUnit:='kvarh';
    Config.atypChannelConfig[5].sChannelName := 'EQ';

    Config.atypChannelConfig[6].xChannelExists := TRUE;
    Config.atypChannelConfig[6].bDecimalPlaces:=2;
    Config.atypChannelConfig[6].sUnit:='V';
    Config.atypChannelConfig[6].sChannelName := 'L1';

    Config.atypChannelConfig[7].xChannelExists := TRUE;
    Config.atypChannelConfig[7].bDecimalPlaces:=2;
    Config.atypChannelConfig[7].sUnit:='V';
    Config.atypChannelConfig[7].sChannelName := 'L2';

    Config.atypChannelConfig[7].xChannelExists := TRUE;
    Config.atypChannelConfig[7].bDecimalPlaces:=2;
    Config.atypChannelConfig[7].sUnit:='V';
    Config.atypChannelConfig[7].sChannelName := 'L3';

    Config.atypChannelConfig[8].xChannelExists := TRUE;
    Config.atypChannelConfig[8].bDecimalPlaces:=2;
    Config.atypChannelConfig[8].sUnit:='I';
    Config.atypChannelConfig[8].sChannelName := 'L1';

    Config.atypChannelConfig[9].xChannelExists := TRUE;
    Config.atypChannelConfig[9].bDecimalPlaces:=2;
```

```
Config.atypChannelConfig[9].sUnit:='I';
Config.atypChannelConfig[9].sChannelName := 'L2';

Config.atypChannelConfig[10].xChannelExists := TRUE;
Config.atypChannelConfig[10].bDecimalPlaces:=2;
Config.atypChannelConfig[10].sUnit:='I';
Config.atypChannelConfig[10].sChannelName := 'L3';

Config.atypChannelConfig[11].xChannelExists := TRUE;
Config.atypChannelConfig[11].bDecimalPlaces:=2;
Config.atypChannelConfig[11].sChannelName := 'PF';

Config.atypChannelConfig[12].xChannelExists := TRUE;
Config.atypChannelConfig[12].bDecimalPlaces:=2;
Config.atypChannelConfig[12].sUnit:='I';
Config.atypChannelConfig[12].sChannelName := 'L3';

Config.atypChannelConfig[13].xChannelExists := TRUE;
Config.atypChannelConfig[13].bDecimalPlaces:=2;
Config.atypChannelConfig[13].sUnit:='INT';
Config.atypChannelConfig[13].sChannelName := 'i';

xInit:= FALSE;
END_IF

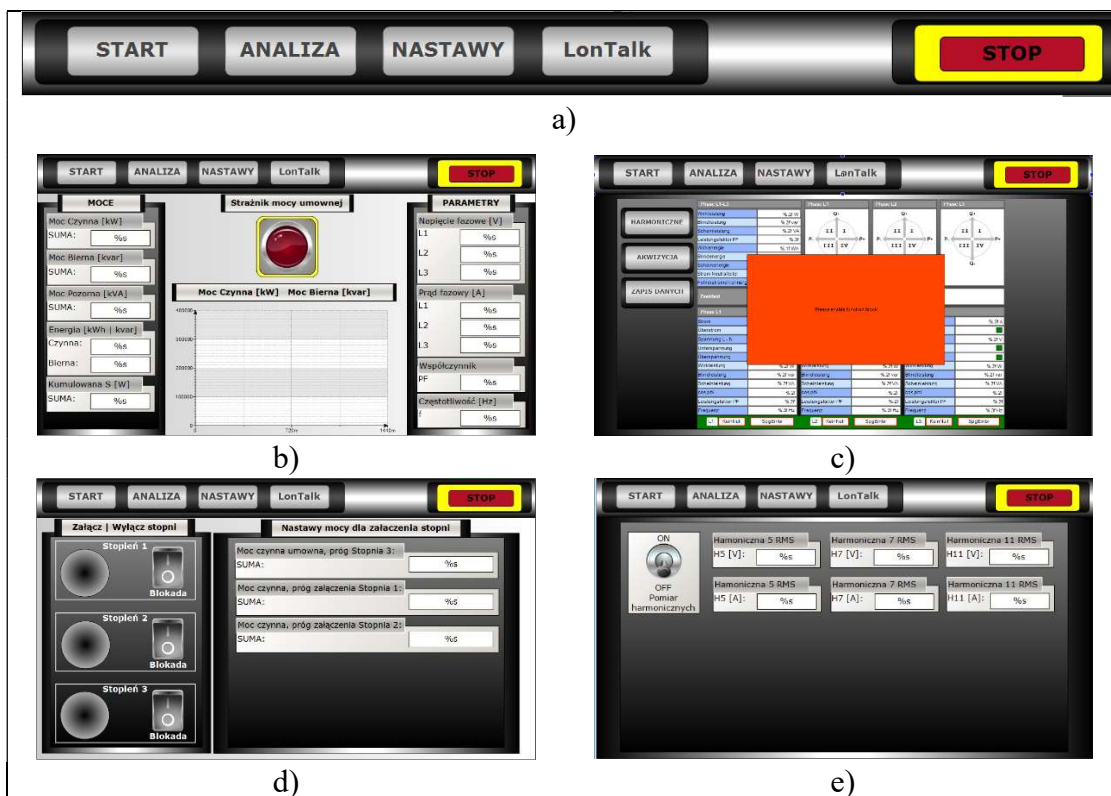
i:=i+1;
//aValues[1]:= TO_REAL(i);
aValues[1] := rMOCCzynna;
aValues[2] := rMOCBierna;
aValues[3] := rMOCPozorna;
aValues[4] := rENERGIACzynna;
aValues[5] := rENERGIABierna;
aValues[6] := PLC_PRG.fb_Compact1.rTotalPowerFactorPF;
aValues[7] := PLC_PRG.fb_Compact1.aVoltage_L_N[1];
aValues[8] := PLC_PRG.fb_Compact1.aVoltage_L_N[2];
aValues[9] := PLC_PRG.fb_Compact1.aVoltage_L_N[3];
aValues[10] := PLC_PRG.fb_Compact1.aCurrent[1];
aValues[11] := PLC_PRG.fb_Compact1.aCurrent[2];
aValues[12] := PLC_PRG.fb_Compact1.aCurrent[3];
aValues[13] := TO_REAL(i);
```

Po analizie działania kodu głównego przystąpiono do tworzenia wizualizacji oraz adaptacji narzędzia do analizy danych historycznych.

Menu narzędzia podzielono 4 części:

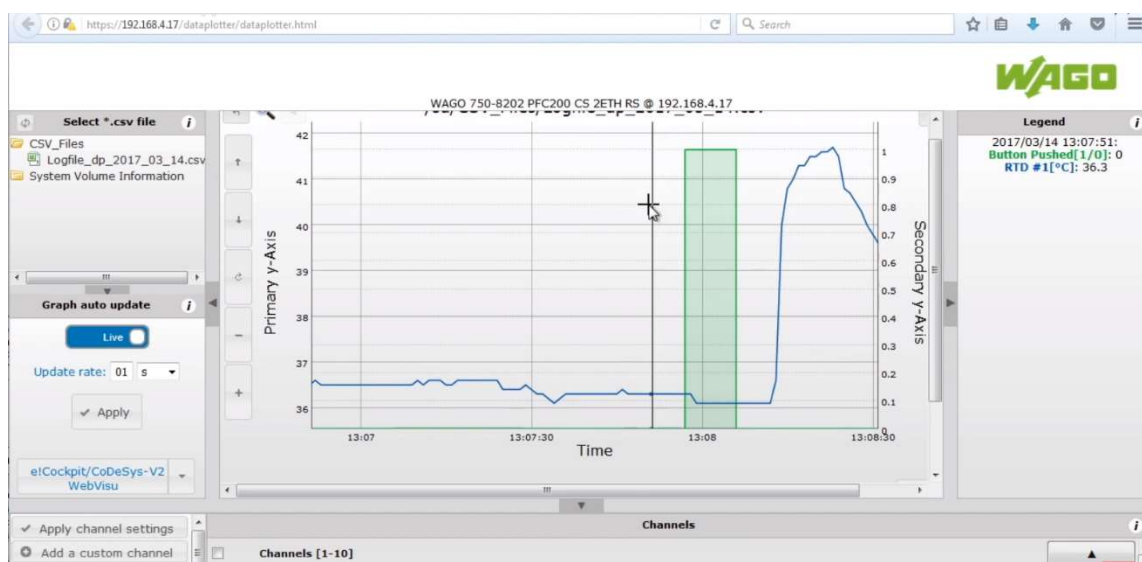
- START,
- ANALIZA,
- NASTAWY,
- LonTalk,

Oraz przycisk awaryjny zatrzymujący działanie wszystkich modułów sprzętowych sterownika. Kolejno na rysunku Rys. 4.4 przedstawiono poszczególne okna aplikacji.



Rys. 4.4 Wybrane okna wizualizacji projektowanej wizualizacji:
 a) MENU, b) START, c) ANALIZA,
 d) NASTAWY, e) wybór zapisu wybranych harmonicznych⁴⁵

Opracowane ekrany wizualizacji zostały zaakceptowane, a następnie zaimplementowano je do rozwiązania docelowego. Do bardziej szczegółowej analizy danych historycznych wykorzystano narzędzie Data Plotter od WAGO, przykładowy panel przedstawia Rys. 4.5.



Rys. 4.5 Okno aplikacji Data Plotter WAGO⁴⁶

⁴⁵ Źródło: opracowanie własne

⁴⁶ Źródło: opracowanie własne

4.2. Smart EMS

Projekt kompaktowego systemu Smart Energy Management System (SEMS) zakładał konstrukcję narzędzia do audytowania odbiorców energii elektrycznej o stosunkowo rozbudowanej sieci zasilającej. Jedną z najważniejszych cech miało być automatyczne generowanie raportu dotyczącego bilansu energetycznego. Niemniej jednak w trakcie opracowywania założeń autor został członkiem zespołów badawczych w projektach RELFlex i KlastER. Aktywność związana z działalnością na rzecz KE przerodziła się w chęć opracowania brakującego na rynku prostego systemu, który będzie użyteczny zarówno z punktu widzenia koordynatora klastra jak i członka będącego prosumentem lub odbiorcą końcowym. Tym razem zdecydowano się na zastosowanie w jak największym stopniu osprzętu i narzędzi otwartych. Takie założenie przyjęto, ponieważ część istniejących klastrów dysponowała już różnymi komercyjnymi rozwiązaniami, lecz w 95% przypadków było ono zamknięte, a wszelkie formy integracji wymagały rozmów z dostawcami. Zdarzały się rozwiązania udostępniające ogólne protokoły transmisji stosowane w „inteligentnym” opomiarowaniu jak np. LonWorks®, natomiast rozwiązania te ograniczały listę możliwych do zastosowania rozwiązań oraz uniemożliwiały ingerencję w ich dalszy rozwój lub modyfikację.

Z uwagi na powyższe fakty postanowiono zrobić „ukłon” w stronę narzędzi *open source*. Jedną z najważniejszych zalet otwartego oprogramowania jest możliwość użycia go dowolnie modyfikując i rozpowszechniając. Należy jednak pamiętać, że istnieje kilka licencji otwartego oprogramowania, a w szczególności licencje zgodne z *Free Software Foundation* takie jak GNU GPL mają charakter wręcz jak wirusy tzn. każdy program, który wykorzystuje część kodu innego programu musi być udostępniony na tych samych zasadach licencjonowania. Przykładem łagodniejszej formy licencji na otwarte oprogramowanie jest Lesser GPL, który dopuszcza wykorzystanie bibliotek GNU GPL przez rozwiązania o zamkniętej licencji. Oczywiście istnieje kilka modeli licencji oprogramowania otwartego, dlatego w przypadku *open source* koniecznym jest zgłębienie zasad licencjonowania. Zgodnie z *Oen Source Initiative* (OSI) licencja winna być neutralna i nie podlegać zastrzeżeniu patentowym, niemniej jednak nie wyklucza to czerpania korzyści finansowych z tworzenia oprogramowania. Organizacja OSI wręcz zachęca do inwestowania w rozwój otwartego oprogramowania, ponieważ tylko taki sposób tworzenia zapewnia najwyższy poziom bezpieczeństwa, niezawodność i jakość.

Wielu z pośród dzisiejszych użytkowników urządzeń elektronicznych kojarzy narzędzia *open source* z systemem Linux i pasjonatami informatyki, a także nie do końca w pełni funkcjonalnym systemem. Warto przywołać kilka faktów: 74,45% rynku systemów operacyjnych to Android, którego jądro pochodzi w 75% z Linux, 70% serwerów pracuje na systemach Unix w tym aż 37% jest stwierdzone, że pracują na systemie Linux (usługi typu „chmury”) dla dużych koncernów i korporacji. Ostateczna dominacja Linux przejawia się na rynku superkomputerów gdzie z pośród 500 aż 498 pracuje na Linux.

W przypadku twórców oprogramowania w minionych latach bardzo często spotyka się modele, gdzie całość środowiska produkcyjnego jest oparta o narzędzia *open source*. Począwszy od środowiska programistycznych np. *Eclipse*, kompilatorów, np. *gcc*, aż po system wersjonowania np. *svn*. W przypadku innych dziedzin inżynierii częstym rozwiązaniem jest dostarczanie pakietów komercyjnych w wersjach studenckich do uczelni takich jak, *Matlab*, *LabVIEW*, *Ansys*, *SOLIDWORKS*, *Altium* etc. Ma to na celu powiązanie danego produktu z przyszłym rynkiem. Jednakowoż należy zaznaczyć, że takie postępowanie winno być nagradzane aplauzem oraz powinno być naśladowane. z innej jednak strony może być przyczyną się protekcjonalizmu inżynierów do innych rozwiązań.

Rozwinięty wstęp podrozdziału ma na celu ukazanie, że w trakcie studiów mimo zachodzących zmian nadal zbyt mało uwagi poświęca się otwartym narzędziom. Oczywiście uczelnie szkolą kadrę inżynierów oraz dbają o ich przyszłe miejsca pracy, a w przypadku dużych koncernów takich jak *automotive* czy telekomunikacja niekwestionowany jest wybór profesjonalnych narzędzi wraz z systemem wsparcia. W przypadku *start-up* częstym problemem jest brak możliwości finansowych na zakup profesjonalnego rozwiązania wówczas z pomocą przychodzą narzędzia otwarte.

To właśnie SEMS jest narzędziem mogącym być podstawą *start-up-u*. W przypadku rzeczowego narzędzia nakłady związane z wykorzystaniem komercyjnym narzędzi zamkniętych znacząco podwyższyłyby koszty związane z wdrożeniem narzędzia, a wręcz stanowiły barierę zaporową wdrożenia na rynek. Przy koncepcji tworzenia systemu założono wykorzystanie w jak największym stopniu środowisk otwartych, ponieważ znacząco redukowały koszty urządzeń oraz nie zamykały się na możliwości późniejszej modyfikacji i integracji z innymi systemami. Koncepcję rozwiązania zaprezentowano na Rys. 4.6.

Koncepcja budowy systemu Smart EMS zakładała korzystanie istniejących na rynku komponentów, które nie ograniczały dalszego rozwoju lub przeniesienia narzędzia na inną platformę. W projekcie przyjęto następujące główne założenia:

- współpraca ze środowiskiem przemysłowym,
- narzędzie dla odbiorców indywidualnych i prosumentów,
- brak konieczności ingerencji w instalacje użytkowników,

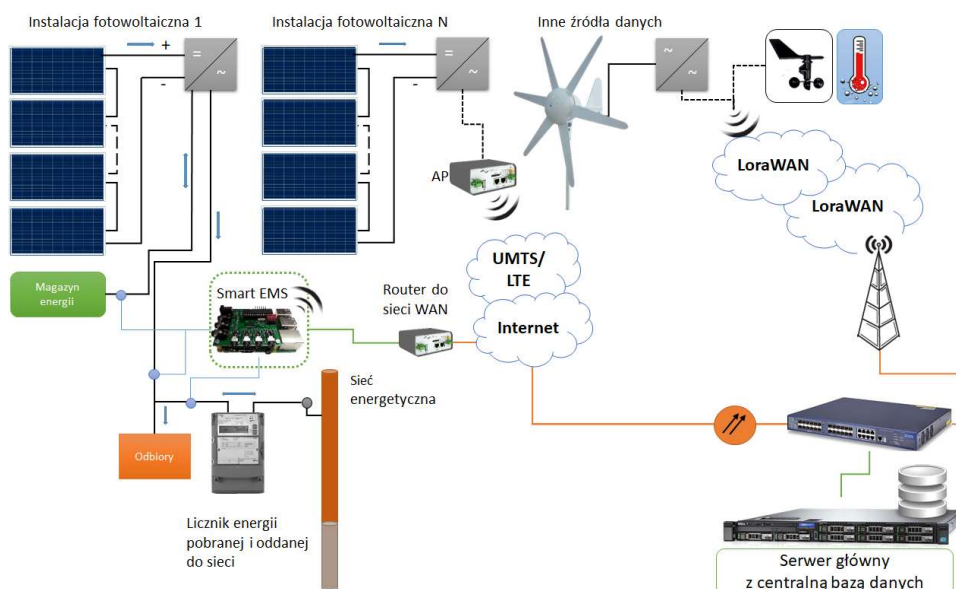
Jako podstawowe funkcjonalności zakładano:

- zarządzanie energią,
- monitorowanie zużycia,
- monitorowanie produkcji,
- informacja o bilansie energetycznym,
- pomiar podstawowych wartości elektrycznych i nieelektrycznych,
- integracja z systemami BMS,
- monitorowanie i telemetria,

- rynek przedsiębiorstw wchodzących w przemysł 4.0,
- integracja z rozwiązaniami IoT,
- możliwość kontroli i sterownia magazynem energii według wiedzy eksperckiej,
- integracja „chmurowym” z repozytorium danych,
- lokalna baza danych,
- otwartość i możliwość rozwoju o dodatkowe protokoły i funkcjonalności.
- dostęp zdalny z sieci WAN i lokalny z urządzeń mobilnych.

Wymagania stawiane przez koordynatorów KE:

- funkcja analizy parametrów energii elektrycznej,
- pomiar wartości chwilowych oraz rejestracja przebiegów w czasowych następujących parametrów niezależnie dla każdej fazy:
 - o prąd,
 - o napięcie,
 - o moc czynna,
 - o moc bierna,
 - o moc pozorna,
 - o energia czynna,
 - o energia bierna,
 - o bilans energetyczny obiektu,
 - o bilansowanie międzyfazowe.
 - o akwizycja danych pomiarowych na zdalnym serwerze,
 - o dostęp do danych archiwalnych z kilku lat za pomocą sieci Internet,
 - o alarmowanie stanów (niezdefiniowano),
 - o bezinwazyjna instalacja urządzenia w obwody prądowe.

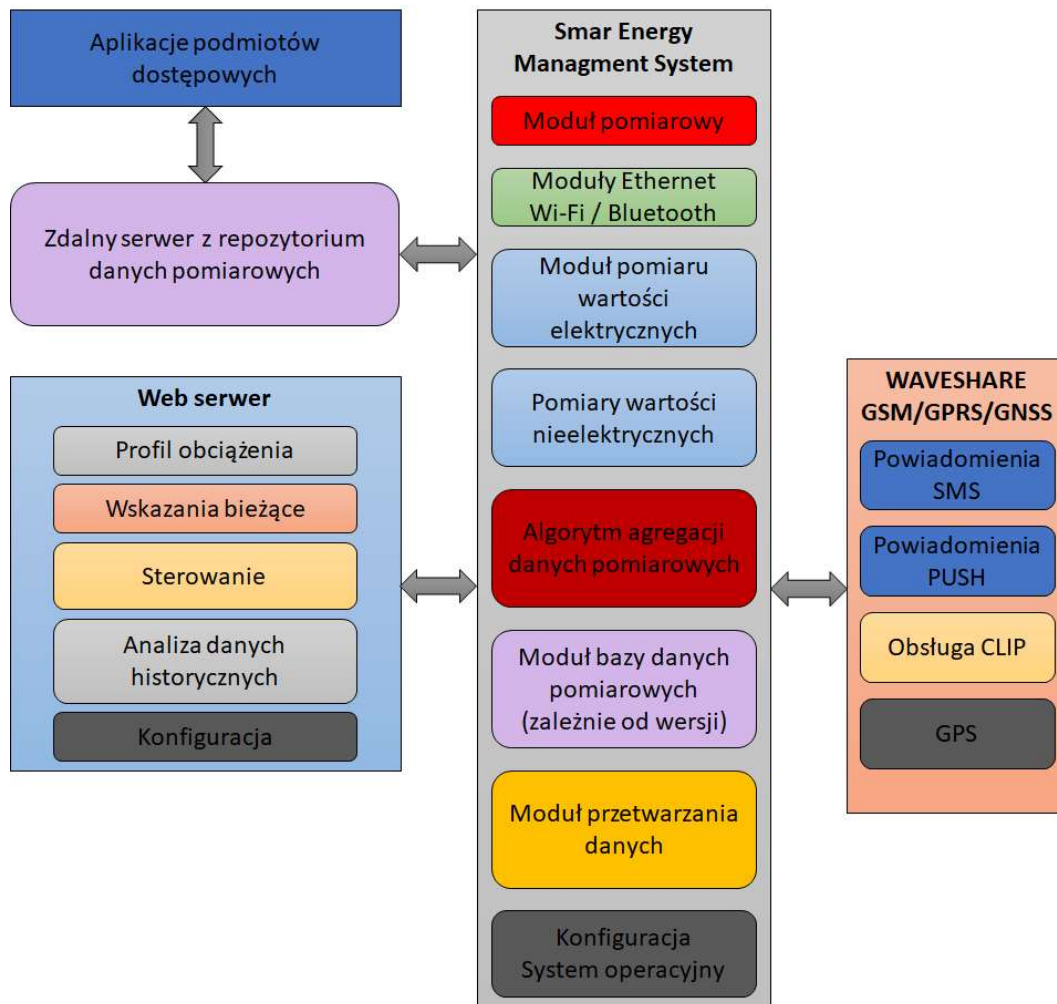


Rys. 4.6 Koncepcja funkcjonowania SEMS⁴⁷

⁴⁷ Źródło: opracowanie własne

Jako jednostkę centralną sytemu wykorzystano komputer jednopłytkowy Raspberry Pi przedstawiony na Rys. 3.6 wraz z zaprojektowaną modyfikacją układu do pomiaru wartości elektrycznych.

Schemat funkcjonalny projektowanego sytemu przedstawia na Rys. 4.7.

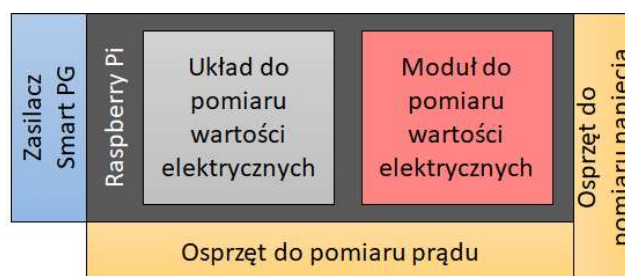


Rys. 4.7 Schemat aplikacyjny wraz z podziałem na bloki funkcjonalne⁴⁸

Zastosowanie Raspberry Pi jako bazy dla systemu ma wiele korzyści, począwszy od dostępu do szerokiej społeczności wsparcia technicznego i programowego jak i komplementarność układu. Na płycie mamy złącza USB, port Gigabit Ethernet, wbudowane Wi-Fi 2,4/5 GHz (IEEE 802.11.b/g/n/ac), a także Bluetooth 4.2, które wykorzystano do obsługi i codziennej komunikacji opracowanego narzędzia. Jako opcję dodatkową planowano zastosować układ telefonu komórkowego wraz z GPS (Waveshare GSM/GPRS/GNSS/Bluetooth). Założenia obejmowały również zastosowanie małego magazynu energii do podtrzymania zasilania w przypadku zaniku napięcia.

Analogiczny do Smart PG schemat blokowy narzędzia Smart EMS do zastosowań w instalacjach prosumenckich przedstawiono na Rys. 4.8.

⁴⁸ Źródło: opracowanie własne

Rys. 4.8 Schemat blokowy konstrukcji Smart EMS⁴⁹

4.2.1. Oprogramowanie układu mikroprocesorowego

W odróżnieniu od poprzedniego narzędzia Smart EMS był programowany w kilku środowiskach jednocześnie. Począwszy od powłoki bash w systemie Raspbian poprzez IDLE, konfiguracji bazy danych, aż do ustawień pulpitów nawigacyjnych (wizualizacji) w Grafana. Równoległe z oprogramowaniem prowadzono proces projektowania modyfikacji układu scalonego do pomiarów wielkości elektrycznych oraz prowadzono dobór narzędzi do pomiarów prądowych.

Obliczanie wartości skutecznej prądu i napięcia:

$$I_{RMS} = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} i_k^2} \quad (1)$$

$$U_{RMS} = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} U_k^2} \quad (2)$$

gdzie:

- i_k – wartość próbkowania prądu,
- I_{RMS} – wartość skuteczna prądu,
- U_k – Wartość próbkowania napięcia
- U_{RMS} – wartość skuteczna napięcia,
- N – ilość próbek pomiarowych

Program układu hardware odpowiedzialny za przeprowadzenie obliczeń i pomiarów

```
#!/usr/bin/python
#
# VERSION 2,0,1

import ConfigParser, sys, struct, serial
import os,path
import optparse

if os,path,exists('/dev/ttyAMA0'):
    serial_port = '/dev/ttyAMA0'
elif os,path,exists('/dev/ttyS0'):
    serial_port = '/dev/ttyS0'
else:
```

⁴⁹ Źródło: opracowanie własne

```

    print 'Could not detect any supported serial port on device'
    sys.exit()

ser = serial.Serial(serial_port, 38400)
ser.flushInput()
ser.flushOutput()

key = 'C-|\x19'
#key = '\x43\x2D\x7C\x19'
MAX_NODES = 0
MAX_CHANNELS = 0

def extr_float(lL, idx):
    out = ""
    for i in range(4):
        out += lL[idx+i]
    return out

def extr_byte(lL, idx):
    out = lL[idx]+'\\x00'+ '\\x00'+ '\\x00'
    return out

def extr_int(lL, idx):
    out = lL[idx]+ lL[idx+1]+ '\\x00'+ '\\x00'
    return out

def wait_and_read(options):
    # Function waiting to receive the id key
    # Once id key detected the structure is read (stored in L[4])
    global MAX_NODES
    global MAX_CHANNELS
    fileoutname = "/tmp/rpict,conf"
    f = open(fileoutname, 'w')
    f.write("[main]\n")

    L = ["", "", "", ""]
    read_length = 5
    while len(L) < read_length:
        r = ser.read(1)
        #if options, debug: print "%d/%d" %
(len(L), read_length)
        if "", join(L[:4]) == key:
            L.append(r)
            if len(L) == 5:
                if options, debug: print "\n# Received
magic key ok,"

                if L[4] == '\\xa0':
                    read_length = 21
                elif L[4] == '\\xa1':
                    read_length = 49
                elif L[4] == '\\xa2':
                    read_length = 52
                elif L[4] == '\\xb0':
                    N_SENSORS = 8
                    MAX_NODES = 25 # for 0xB0

only
                    MAX_CHANNELS = 64 # for 0xB0

only
                    read_length =
47+4*MAX_NODES+2*MAX_CHANNELS
                elif L[4] == '\\xb1':

```

```

N_MAX_BOARD = 5
N_SENSORS = 8
MAX_NODES = 25 # for 0xB1

only
MAX_CHANNELS = 64 # for 0xB1

only
read_length =
47+128+4*MAX_NODES+2*MAX_CHANNELS
elif L[4]=='\xb2':
N_MAX_BOARD = 5
N_SENSORS = 8
MAX_NODES = 28 # for 0xB2

only
MAX_CHANNELS = 64 # for 0xB2

only
read_length =
47+128+4*MAX_NODES+2*MAX_CHANNELS
elif L[4]=='\xb3':
N_MAX_BOARD = 5
N_SENSORS = 8
MAX_NODES = 28 # for 0xB2

only
MAX_CHANNELS = 64 # for 0xB2

only
read_length =
48+128+4*MAX_NODES+2*MAX_CHANNELS
elif L[4]=='\xb4':
N_MAX_BOARD = 5
N_SENSORS = 8
MAX_NODES = 28 # for 0xB2

only
MAX_CHANNELS = 64 # for 0xB2

only
read_length =
51+128+4*MAX_NODES+2*MAX_CHANNELS
elif L[4]=='\xc0':
N_MAX_BOARD = 2
N_SENSORS = 8
MAX_NODES = 16 #
MAX_CHANNELS = 64 #
read_length =
79+4*MAX_NODES+2*MAX_CHANNELS
elif L[4]=='\xc1':
N_MAX_BOARD = 2
N_SENSORS = 8
MAX_NODES = 16 #
MAX_CHANNELS = 64 #
read_length =
80+4*MAX_NODES+2*MAX_CHANNELS
else:
print "Unknown structure

0x%02x" % ord(L[4])

sys,exit()

else:
if options,debug: print ",",
L,append(r)
L,pop(0)

print "# "
print "# Configuration in memory:"
print "# "
print "# Structure: 0x%02x" % ord(L[4])

```

```

idx = 5
format_ = struct,unpack('I', extr_byte(L, idx))
idx += 1
print "# Format: %d" % format_
f,write("format = %d\n" % format_)
nodeid = struct,unpack('I', extr_byte(L, idx))
idx += 1
print "# NodeId: %d" % nodeid
f,write("nodeid = %d\n" % nodeid)
polling = struct,unpack('I', extr_int(L, idx))
idx += 2
print "# Polling: %d" % polling[0]
f,write("polling = %d\n" % polling[0])

if L[4] in ['\xb0', '\xb1', '\xb2', '\xb3', '\xc0', '\xc1']: #
RPICT7V1 & RPICT4V3 version 2 + RPICT8
    print "# KCAL: ",
    f,write("kcal =")
    if L[4]=='\xb0': nkcalread = N_SENSORS
    elif L[4] in ['\xb1', '\xb2', '\xb3', '\xc0', '\xc1']:
nkcalread = N_SENSORS * N_MAX_BOARD
    for u in range(nkcalread):
        sup = struct,unpack('f', extr_float(L,
idx))[0],
        f,write(" %s" % sup)
        print " %s" % sup,
        idx += 4
    f,write("\n")
    print

if L[4] in ['\xb3', '\xc1']:
idx)
    PHASECAL = struct,unpack('I', extr_byte(L,
    idx += 1
    print "# PHASECAL: %d" % PHASECAL
    f,write("phasecal = %d\n" % PHASECAL)

vest = struct,unpack('f', extr_float(L,idx))
idx += 4
print "# VEST: %f" % vest
f,write("vest = %f\n" % vest)

Nnode = struct,unpack('I', extr_byte(L, idx))
idx += 1
print "# Nnode: %d" % Nnode
f,write("Nnode = %d\n" % Nnode)

Nchan = struct,unpack('I', extr_byte(L, idx))
idx += 1
print "# Nchan: %d" % Nchan
f,write("Nchan = %d\n" % Nchan)

f,write("HWSCT =")
for u in range(MAX_NODES):
    sup = struct,unpack('I', extr_byte(L,
idx))[0]
    f,write(" %s" % sup)
    idx += 1
    f,write("\n")

```

```

f,write("HWMCPST =")
for u in range(MAX_NODES):
    sup = struct,unpack('I', extr_byte(L,
idx))[0]
    f,write(" %s" % sup)
    idx += 1
f,write("\n")

f,write("HWVOL =")
for u in range(MAX_NODES):
    sup = struct,unpack('I', extr_byte(L,
idx))[0]
    f,write(" %s" % sup)
    idx += 1
f,write("\n")

f,write("HWMCPVOL =")
for u in range(MAX_NODES):
    sup = struct,unpack('I', extr_byte(L,
idx))[0]
    f,write(" %s" % sup)
    idx += 1
f,write("\n")

f,write("CHTYPE =")
for u in range(MAX_CHANNELS):
    sup = struct,unpack('I', extr_byte(L,
idx))[0]
    f,write(" %s" % sup)
    idx += 1
f,write("\n")

f,write("CHID =")
for u in range(MAX_CHANNELS):
    sup = struct,unpack('I', extr_byte(L,
idx))[0]
    f,write(" %s" % sup)
    idx += 1
f,write("\n")
print "# Complete configuration in %s" % fileoutname
elif L[4]=='\xb4': # RPICT7V1 & RPICT4V3 version 2 + RPICT8
print "# KCAL: ",
f,write("kcal =")
nkcalread = N_SENSORS * N_MAX_BOARD
for u in range(nkcalread):
    sup = struct,unpack('f', extr_float(L,
idx))[0],
    f,write(" %s" % sup)
    print " %s" % sup,
    idx += 4
f,write("\n")
print

PHASECAL = struct,unpack('b', L[idx])#extr_byte(L,
idx))
idx += 1
print "# PHASECAL: %d" % PHASECAL
f,write("phasecal = %d\n" % PHASECAL)

```



```

vest = struct,unpack('f', extr_float(L,idx))
idx += 4
print "# VEST: %f" % vest
f,write("vest = %f\n" % vest)

xpFREQ = struct,unpack('I', extr_byte(L, idx))
idx += 1
print "# xpFREQ: %d" % xpFREQ
f,write("xpFREQ = %d\n" % xpFREQ)

Ncycle = struct,unpack('I', extr_byte(L, idx))
idx += 1
print "# Ncycle: %d" % Ncycle
f,write("Ncycle = %d\n" % Ncycle)

Nnode = struct,unpack('I', extr_byte(L, idx))
idx += 1
print "# Nnode: %d" % Nnode
f,write("Nnode = %d\n" % Nnode)

Nchan = struct,unpack('I', extr_byte(L, idx))
idx += 1
print "# Nchan: %d" % Nchan
f,write("Nchan = %d\n" % Nchan)

f,write("HWSCT =")
for u in range(MAX_NODES):
    sup = struct,unpack('I', extr_byte(L,
idx)) [0]
    f,write(" %s" % sup)
    idx += 1
f,write("\n")

f,write("HWMCPST =")
for u in range(MAX_NODES):
    sup = struct,unpack('I', extr_byte(L,
idx)) [0]
    f,write(" %s" % sup)
    idx += 1
f,write("\n")

f,write("HWVOL =")
for u in range(MAX_NODES):
    sup = struct,unpack('I', extr_byte(L,
idx)) [0]
    f,write(" %s" % sup)
    idx += 1
f,write("\n")

f,write("HWMCPVOL =")
for u in range(MAX_NODES):
    sup = struct,unpack('I', extr_byte(L,
idx)) [0]
    f,write(" %s" % sup)
    idx += 1
f,write("\n")

f,write("CHTYPE =")
for u in range(MAX_CHANNELS):

```

```

        sup = struct,unpack('I', extr_byte(L,
idx))[0]
        f,write(" %s" % sup)
        idx += 1
    f,write("\n")

    f,write("CHID =")
    for u in range(MAX_CHANNELS):
        sup = struct,unpack('I', extr_byte(L,
idx))[0]
        f,write(" %s" % sup)
        idx += 1
    f,write("\n")

    debug = struct,unpack('I', extr_byte(L, idx))
    idx += 1
    print "# debug: %d" % debug
    f,write("debug = %d\n" % debug)

    print "# Complete configuration in %s" % fileoutname
print "# "

f,close()

return L

def config_serialise(config, structure):

    tts = key
    tts += structure
    tts += chr(int(config,get('main', 'format')))
    tts += struct,pack('B', int(config,get('main',
'nodeid')))#nodeid
    tts += struct,pack('H', int(config,get('main',
'polling')))#polling
    if structure=='\xa0':
        tts += struct,pack('f', float(config,get('main',
'ical')))#ical
        tts += struct,pack('f', float(config,get('main',
'vcal')))#vcal
        tts += struct,pack('f', float(config,get('main',
'vest')))#vest
    elif structure=='\xa1':
        kcal = config,get('main', 'kcal'),split()
        for k in kcal:
            tts += struct,pack('f', float(k))
            tts += struct,pack('f', float(config,get('main',
'phasecal')))#phasecal
            tts += struct,pack('f', float(config,get('main',
'vest')))#vest
    elif structure=='\xa2':
        kcal = config,get('main', 'kcal'),split()
        for k in kcal:
            tts += struct,pack('f', float(k))
            tts += struct,pack('B', int(config,get('main',
'phasecal')))#phasecal
            tts += struct,pack('f', float(config,get('main',
'vest')))#vest
            tts += struct,pack('B', int(config,get('main',
'xFREQ')))# xPFREQ

```

```

        tts += struct,pack('B', int(config,get('main',
'Ncycle')))# Ncycle
        tts += struct,pack('B', int(config,get('main',
'debug')))# debug
        elif structure in
['\xb0', '\xb1', '\xb2', '\xb3', '\xc0', '\xc1']:
            kcal = config,get('main', 'kcal'),split()
            for k in kcal:
                tts += struct,pack('f', float(k))
            if structure in ['\xb3', '\xc1']: tts +=
struct,pack('c', chr(int(config,get('main', 'phasecal')))#phasecal
            tts += struct,pack('f', float(config,get('main',
'vest')))#vest
            tts += struct,pack('c', chr(int(config,get('main',
'Nnode')))#Nnode
            tts += struct,pack('c', chr(int(config,get('main',
'Nchan')))#Nchan

            hwsct = config,get('main', 'HWSCT'),split()
            if len(hwsct) != MAX_NODES:
                print "config error HWSCT"
                print "Expected length %d got %d" %
(MAX_NODES,len(hwsct))
                sys,exit()
            for k in hwsct:
                tts += struct,pack('c', chr(int((k))))

            hwmcpsct = config,get('main', 'HWMCPST'),split()
            if len(hwmcpsct) != MAX_NODES:
                print "config error HWMCPST"
                print "Expected length %d got %d" %
(MAX_NODES,len(hwmcpsct))
                sys,exit()
            for k in hwmcpsct:
                tts += struct,pack('c', chr(int((k))))

            hwvol = config,get('main', 'HWVOL'),split()
            if len(hwvol) != MAX_NODES:
                print "config error HWVOL"
                print "Expected length %d got %d" %
(MAX_NODES,len(hwvol))
                sys,exit()
            for k in hwvol:
                tts += struct,pack('c', chr(int((k))))

            hwmcpvol = config,get('main', 'HWMCPVOL'),split()
            if len(hwmcpvol) != MAX_NODES:
                print "config error HWMCPVOL"
                print "Expected length %d got %d" %
(MAX_NODES,len(hwmcpvol))
                sys,exit()
            for k in hwmcpvol:
                tts += struct,pack('c', chr(int((k))))

            chtype = config,get('main', 'CHTYPE'),split()
            if len(chtype) != MAX_CHANNELS:
                print "config error CHTYPE"
                print "Expected length %d got %d" %
(MAX_CHANNELS,len(chtype))
                sys,exit()
            for k in chtype:

```

```

        tts += struct,pack('c', chr(int((k))))

    chid = config,get('main', 'CHID'),split()
    if len(chid) != MAX_CHANNELS:
        print "config error CHID"
        print "Expected length %d got %d" %
(MAX_CHANNELS,len(chid))
        sys,exit()
    for k in chid:
        tts += struct,pack('c', chr(int((k))))
    elif structure == '\xb4':
        kcal = config,get('main', 'kcal'),split()
        for k in kcal:
            tts += struct,pack('f', float(k))
        tts += struct,pack('b', int(config,get('main',
'phasecal')))#phasecal
        tts += struct,pack('f', float(config,get('main',
'vest')))#vest
        tts += struct,pack('B', int(config,get('main',
'xpFREQ')))#phasecal
        tts += struct,pack('B', int(config,get('main',
'Ncycle')))#phasecal
        tts += struct,pack('B', int(config,get('main',
'Nnode')))#Nnode
        tts += struct,pack('B', int(config,get('main',
'Nchan')))#Nchan

    hwsct = config,get('main', 'HWSCT'),split()
    if len(hwsct) != MAX_NODES:
        print "config error HWSCT"
        print "Expected length %d got %d" %
(MAX_NODES,len(hwsct))
        sys,exit()
    for k in hwsct:
        tts += struct,pack('B', int((k)))

    hwmcpsct = config,get('main', 'HWMCPST'),split()
    if len(hwmcpsct) != MAX_NODES:
        print "config error HWMCPST"
        print "Expected length %d got %d" %
(MAX_NODES,len(hwmcpsct))
        sys,exit()
    for k in hwmcpsct:
        tts += struct,pack('B', int((k)))

    hwvol = config,get('main', 'HWVOL'),split()
    if len(hwvol) != MAX_NODES:
        print "config error HWVOL"
        print "Expected length %d got %d" %
(MAX_NODES,len(hwvol))
        sys,exit()
    for k in hwvol:
        tts += struct,pack('B', int((k)))

    hwmcpvol = config,get('main', 'HWMCPVOL'),split()
    if len(hwmcpvol) != MAX_NODES:
        print "config error HWMCPVOL"
        print "Expected length %d got %d" %
(MAX_NODES,len(hwmcpvol))
        sys,exit()
    for k in hwmcpvol:

```

```

        tts += struct,pack('B', int((k)))

    chtype = config,get('main', 'CHTYPE'),split()
    if len(chtype) != MAX_CHANNELS:
        print "config error CHTYPE"
        print "Expected length %d got %d" %
(MAX_CHANNELS,len(chtype))
        sys,exit()
    for k in chtype:
        tts += struct,pack('B', int((k)))

    chid = config,get('main', 'CHID'),split()
    if len(chid) != MAX_CHANNELS:
        print "config error CHID"
        print "Expected length %d got %d" %
(MAX_CHANNELS,len(chid))
        sys,exit()
    for k in chid:
        tts += struct,pack('B', int((k)))
    tts += struct,pack('B', int(config,get('main',
'debug')))#booleans
    else:
        print "Unknown structure 0x%02x" % ord(L[4])
        sys,exit()
    return tts

if __name__ == "__main__":
# If just reading then the config is displayed
# If writing then the config file is sent over serial
    # Some option parsing first
    config = ConfigParser,ConfigParser()

    parser = optparse,OptionParser(usage="%prog [-w] filename [--
version]", version="%prog 1,9,0")
    parser,add_option('-w', '--write',
                        dest='filename',
                        default="",
                    )
    parser,add_option('-d', '--debug',
                        dest='debug',
                        default=False,
                        action='store_true',
                    )

    options, remainder = parser,parse_args()

    write_config = False
    if os,path,isfile(options,filename):
        config,read(options,filename)
        write_config = True
    else:
        if options,filename != "":
            print "Can not read file %s" %
options,filename
            sys,exit()

    print "# RPICT Configuration Utility"

```

```

        if write_config:
            print "# Configuration will be overwritten (Ctrl C to
cancel)"
        else:
            print "# Read only"
            print "# Now reset RPICT hardware"
            L = wait_and_read(options)

            if write_config:
                print "# Writing configuration with file %s" %
options,filename
                tts = config_serialise(config, L[4])

                ser,write(tts)

                wait_and_read(options)

            if options,debug:
                while True:
                    try:
                        print ser,readline(),
                        #tobe tested:
                        #print ser,read(),
                    except KeyboardInterrupt:
                        ser,close()
                        break

```

Program bramki komunikacyjnej do bazy danych

```

#!/usr/bin/python
#
# GATEWAY
# VERSION 1,8.4

import ConfigParser
import serial
import os, sys
import requests
import datetime

c = ConfigParser,ConfigParser()
c,read("gateway,conf")

baud = c,getint('system','baud')
serial_port = c,get('system', 'port')
if not os,path,exists(serial_port):
    print 'Serial port not found'
    sys,exit()
ser = serial,Serial(serial_port, baud)

DayOfMonth = datetime,datetime,utcnow(),day
ls_fileopen = False

while True:
    try:
        data_in = ser,readline()
        utcnow = datetime,datetime,utcnow()

        timestamp = utcnow,strftime("%s")
        print data_in
        while data_in[-1] in ['\r', '\n']:

```

```

        data_in = data_in[:-1]

        z = data_in,split(' ')
        csv = ', ',join(z[1:])
        for sect in c,sections():
            if sect=='influxdb' and
c,getboolean(sect,'enabled'):
                #INFLUXDB
                z = data_in,split(' ')
                t = timestamp + '000000000'
                url = c,get(sect, 'url')
                dbname = c,get(sect, 'dbname')
                user = c,get(sect, 'user')
                passwd = c,get(sect, 'passwd')
                measurement = c,get(sect,
'measurement')

                params = {'db': dbname, 'u': user,
'p': passwd}

                i = 0
                payload = ""
                for zz in z[1:]:
                    i += 1
                    payload += "%s,channel=%02d
value=%s %s\
n" % (measurement, i, zz,t)
                print payload
                r = requests,post(url, params=params,
data=paylo
ad)
                print r,text

            if sect=='localsave' and
c,getboolean(sect,'enabled'):
                # LOCALSAVE
                ls_dir = c,get(sect, 'directory')
                filename =
ls_dir+'/'+'+timestamp+',csv'

                if not ls_fileopen:
                    f = open(filename,'w')
                    ls_fileopen = True
                if DayOfMonth != utcnow,day:
                    DayOfMonth = utcnow,day
                    if ls_fileopen: f,close()
                    f = open(filename,'w')
                    ls_fileopen = True

                f,write(timestamp+', '+csv+'\n')

    except KeyboardInterrupt:
        if ls_fileopen: f,close()
        print "Terminating,"
        break

```

Konfiguracja programu bazy danych w przypadku lokalnego punktu pomiarowego

Pobranie i instalacja InfluxDB

```
curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add
-sudo apt install apt-transport-https
sudo apt-get update
sudo apt-get install influxdb
```

Edycja konfiguracji bazy danych

```
sudo nano /etc/influxdb/influxdb.conf
```

Zamiana ustawień zapytań http

```
[http]
# Determines whether HTTP endpoint is enabled.
enabled = true

# The bind address used by the HTTP service.
bind-address = ":8086"

# Determines whether user authentication is enabled over
HTTP/HTTPS.
auth-enabled = false
```

Restart bazy danych

```
sudo service influxdb restart
```

Utworzenie bazy danych

```
sudo influx
CREATE DATABASE mydb_U
SHOW DATABASE
```

Konfiguracja programu dla bramy bazy danych gateway.conf

```
[system]
port=/dev/ttyAMA0
baud=38400

[influxdb]
enabled = True
url = http://localhost:8086/write
dbname = mydb
user = pi
passwd = raspberry
measurement = rpict7v1_noV

[localsave]
enabled = true
directory = /tmp/
```



```

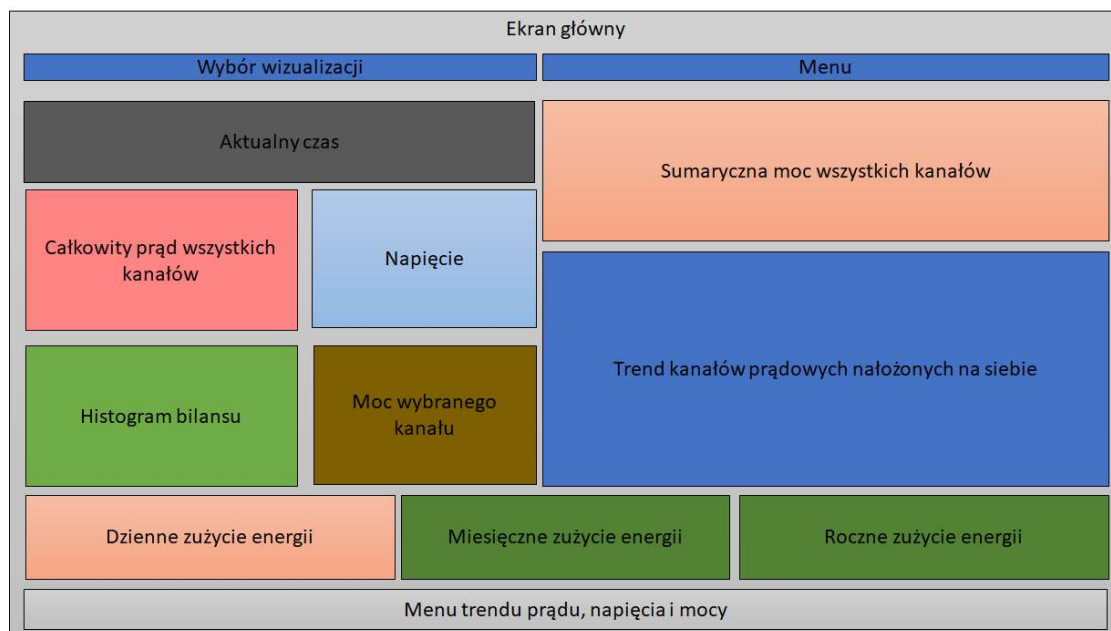
HWSCT = 7 6 5 4 3 2 1 7 6 5 4 3 2 1 255 255 255 255 255 255 255 255
255 255 255 255 255 255
HWMCPST = 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 255 255 255 255
255 255 255 255 255 255 255 255 255 255
HWVOL = 127 127 127 127 127 127 127 127 127 127 127 127 127 127 127 255
255 255 255 255 255 255 255 255 255 255 255 255
HWMCPVOL = 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 255 255 255 255
255 255 255 255 255 255 255 255 255 255
CHTYPE = 4 4 4 4 4 4 4 5 5 5 5 5 5 5 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255
CHID = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255
debug = 0
    
```

Konfiguracja ustawień Grafana

```

curl https://bintray.com/
user/downloadSubjectPublicKey?username=bintray | sudo apt-key add -
sudo apt-get update
sudo apt-get install grafana
sudo service grafana-server start
# sudo update-rc.d grafana-server defaults
sudo systemctl enable grafana-server
    
```

Podczas prac nad koncepcją narzędzia Smart EMS wypracowano założenia dla pulpitu wizualizacyjnego, które następnie realizowano w trakcie pracy (Rys. 4.9).

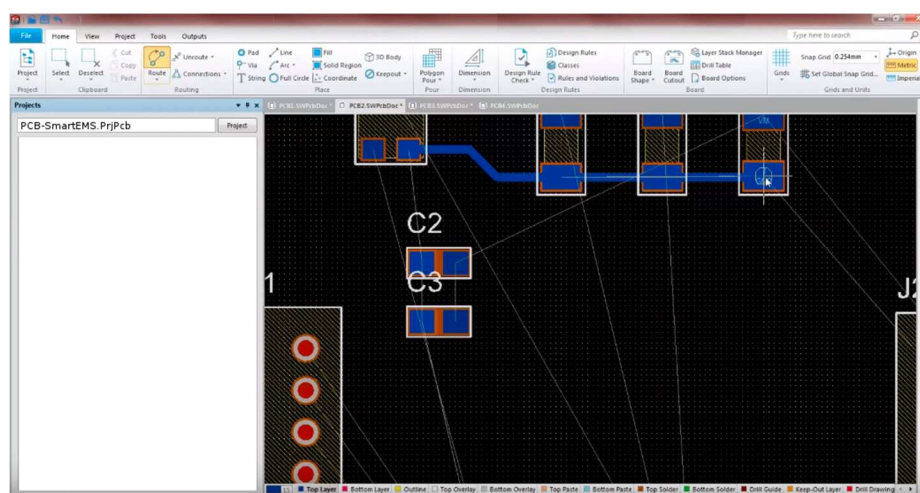


Rys. 4.9 Koncepcja pulpitu wizualizacji Smart EMS⁵¹

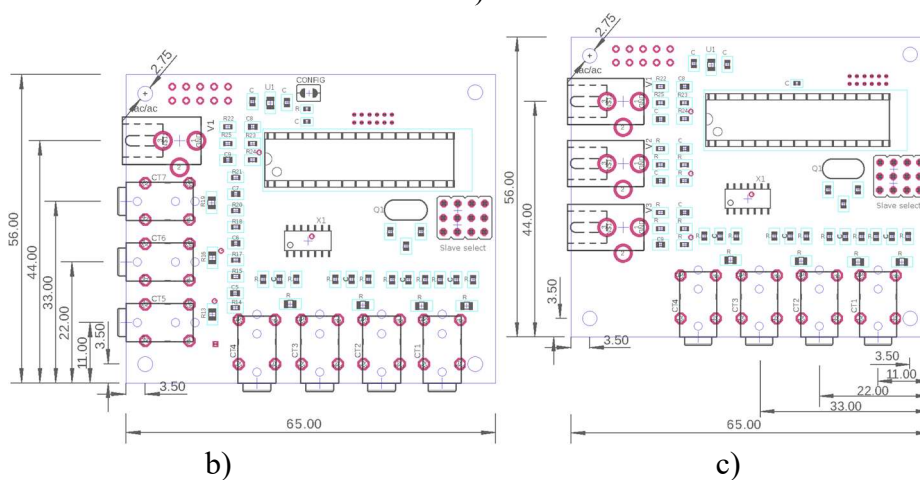
⁵¹ Źródło: opracowanie własne

4.2.1. Projekt układu scalonego i obudowy

Elementy wykorzystane do budowy systemu Smart EMS wymagały oprogramowania oraz modyfikacji układu pomiarowego, a także zaprojektowania obudowy do prowadzenia testów w warunkach rozdzielnic domowych. Narzędzie, z którego skorzystano do opracowania układu oraz projektu obudowy wchodziły w skład SOLIDWORKS Premium lub zostały udostępnione do testów (SOLIDWORKS PCB) Rys. 4.10 a).



a)

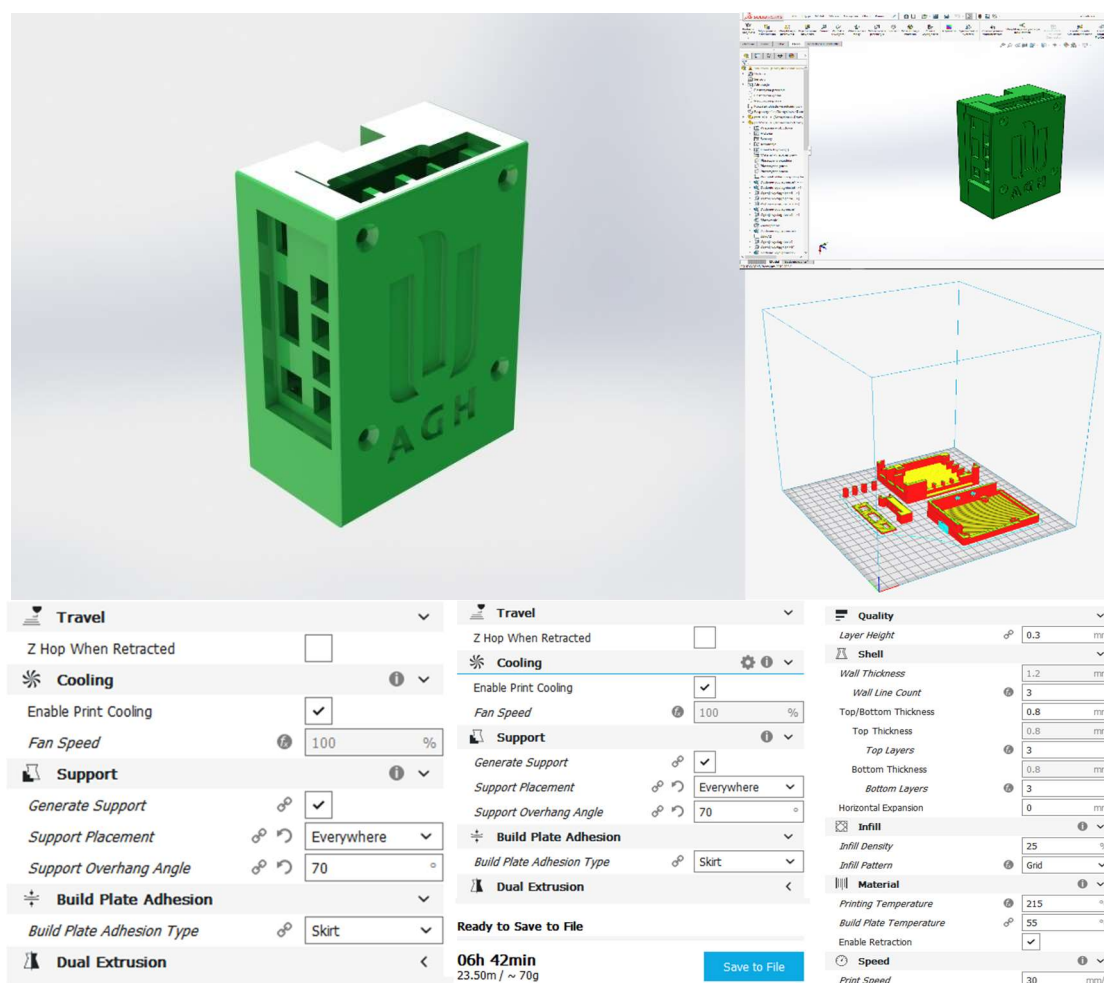


b)

c)

Rys. 4.10 Układ scalony modułu pomiaru wartości elektrycznych:
 a) proces projektowania SOLIDWORKS PCB,
 b) układ dedykowany do audytów rozdzielni,
 c) układ do bilansowania międzyfazowego

Po osiągnięciu docelowego kształtu układu do pomiaru wartości elektrycznych przystąpiono do zaprojektowania obudowy urządzenia pomiarowego. Zaprojektowano kilka wersji obudowy w zależności od ilości modułów podlegających stokowaniu, a także ilości faz napięcia pomiarowego. Zaprojektowano obudowę, a następnie przygotowano pliki z g-kodami dla drukarki 3D. Efekt przedstawiony na Rys. 4.11.



Rys. 4.11 Model obudowy dla Smart EMS⁵²

Osprzęt do pomiaru wartości elektrycznych

Po opracowaniu układu pomiarowego i zaprogramowaniu kolejnym etapem był dobór osprzętu do pomiaru prądów i napięć. W założeniach na początku projektu ustalono, że pomiar będzie się odbywał bez ingerencji lub ograniczał ją do minimum. Zdecydowano się zatem na zastosowanie przekładników prądowych oraz napięciowych (transformator) Rys. 4.12.



Rys. 4.12 Osprzęt do pomiarów:
a) transformator obniżający napięcie, b) przekładnik prądowy⁵³

⁵² Źródło: opracowanie własne

⁵³ Źródło: opracowanie własne

5. Praktyczna realizacja systemów pomiarowych

Rozdział przedstawia realizację założeń zaprezentowanych w rozdziale 4. Zarówno w przypadku Smart PG bazującego w 100% na komercyjnym rozwiązaniu jak i SEMS, które w większości zostało oparte o rozwiązania typu *open source* prace były realizowane według tego samego schematu. Oba systemy do momentu zbudowania działających urządzeń były realizowane w modelu kaskadowym tworzenia oprogramowania. Niemniej jednak urządzenie SEMS, dalej rozwijano i zdecydowano się na model ewolucyjny. Prace były wykonane według następującego planu: w pierwszej kolejności wykonano koncepcję systemów, następnie przystąpiono do przeglądu literaturowego oraz dostępnych na rynku rozwiązań, które umożliwiały wykonanie opracowanych idei. Stopniowo przystąpiono do konfiguracji niezbędnego oprogramowania biorącego udział w procesie tworzenia oprogramowania. Proces tworzenia algorytmów i oprogramowania przebiegał następująco. Na początku przystąpiono do tworzenia specyfikacji dla funkcjonalności i identyfikowano ograniczenia. Kolejno rozpoczęto projektowanie i implementowanie założonych algorytmów, po czym rozpoczęto proces testowania oraz zatwierdzenia (dla Smart PG). Sam proces testowania przebiegał następująco: po każdym ukończonym module był on testowany i funkcjonalność modułu była sprawdzana oraz integrowana z głównym programem. Po czym sukcesywnie testowano i lokalizowano błędy wynikające z integracji poszczególnych podsystemów. Działające systemy sukcesywnie modyfikowano pod dodatkowe zastosowanie.

Należy jednak zaznaczyć, że pomiędzy procesem budowy pierwszego i drugiego rozwiązania istniały zasadnicze różnice. Otóż algorytmy i wizualizacja Smart PG była tworzona od podstaw. W kwestii SEMS z racji otwartości zastosowanych rozwiązań, w wielu przypadkach umożliwiło to skorzystanie z już istniejących rozwiązań po nieznacznej ich modyfikacji.

Druga różnica wynikała z zastosowania docelowego dla narzędzia Smart PG, ponieważ po zakończeniu testów dodatkowo wymagano zatwierdzenia opracowanych funkcjonalności z założeniami. Wynikało to wprost z instalacji sprzętu w obiekcie użyteczności publicznej o podwyższonym standardzie utrzymania warunków klimatycznych.

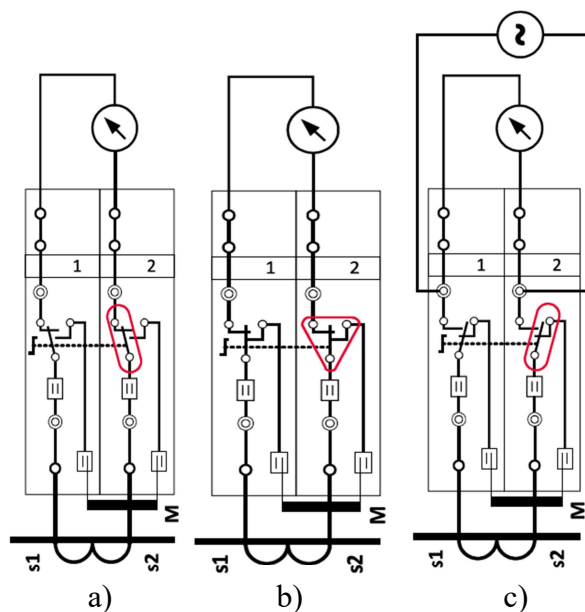
5.1. Smart Power Guard

Zrealizowane prace polegały na zaprojektowaniu i zaprogramowaniu oraz montażu narzędzia Smart PG umożliwiającego rejestrację podstawowych parametrów elektrycznych w zadanych odstępach czasu, a w szczególności prąd, napięcie każdej z faz, mocy czynnej, mocy biernej, mocy pozornej, współczynnika mocy oraz częstotliwości i jej wielokrotności w postaci harmonicznych. Wizualizacja mierzonych parametrów odbywa się poprzez panel HMI znajdujący się w rozdzielni głównej. Panel HMI umożliwia wizualizację danych chwilowych, dodatkowo istnieje możliwość analizy danych archiwalnych za pośrednictwem sieci LAN i komputera klasy PC.

Na podstawie opracowywanych założeń przystąpiono do realizacji praktycznej części pracy dotyczącej Smart PG. Po zrealizowaniu założonych funkcjonalności w środowisku programistycznym przystąpiono do kompletowania części sprzętowej do testów laboratoryjnych. Jako element łączący część niskoprądową (sterownik z algorytmem) z sieciowym napięciem trójfazowym wykorzystano listwę kontrolno-pomiarową. Kolejno zaprezentowano obsługę listwy. Była to istotna kwestia aby nie uszkodzić Sterownika. Skompletowane urządzenie przedstawiono na rys Rys. 5.3.

Obsługa obwodu prądowego ze złączką 282-870

W dalszej części zaprezentowano metody łączenia listwy kontrolno-pomiarowej z rysunku Rys. 3.5. Połączenia na listwie mają istotne znaczenie z uwagi na podłączenie części do zasilania sieci elektrycznej o napięciu 400 V. Natomiast w przypadku torów prądowych listwa zabezpiecza przed uszkodzeniem przekładników. Na Rys. 5.1 przedstawiono położenia łączników roboczych w listwie wraz z opisem poszczególnych położzeń.

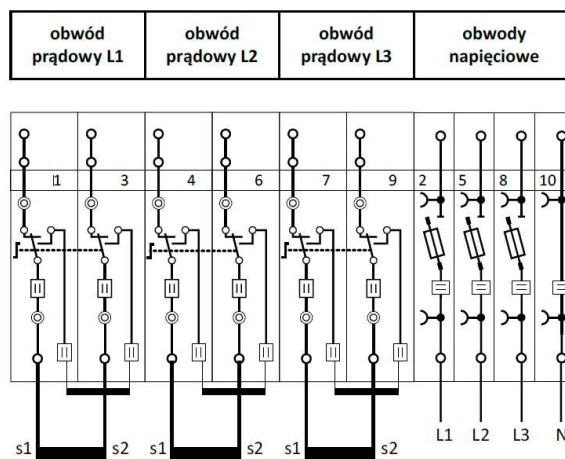


Rys. 5.1 Położenie łączników toru prądowego 282-870: a) robocze, b) położenie pośrednie – wstępne zwarcie strony wtórnej przekładnika, bez przerywania układu pomiarowego, c) położenie testowe⁵⁴

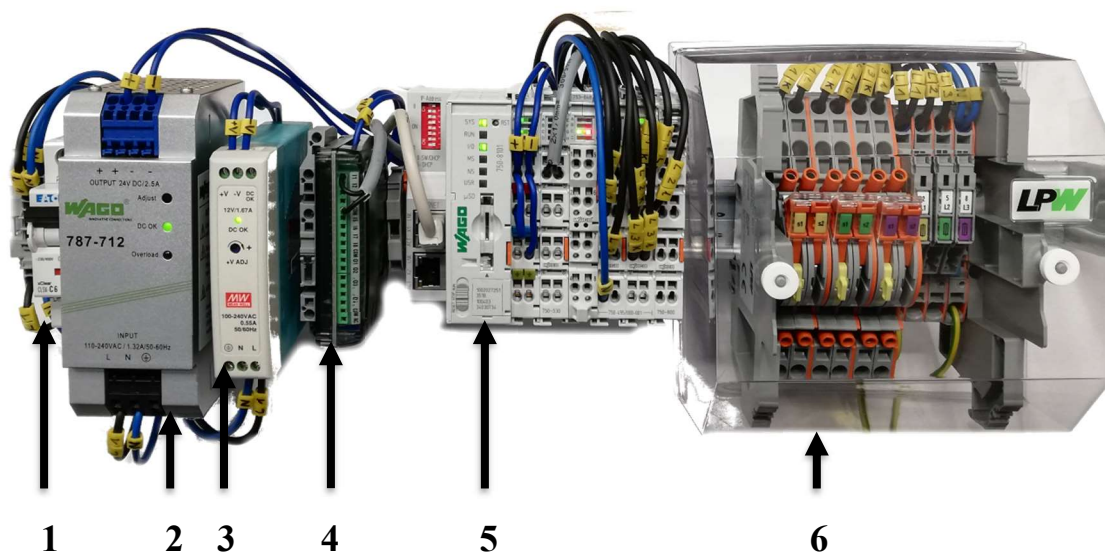
Zgodnie z przedstawionymi położeniami nożyca łącznika prądowego w pozycji roboczej powinna znajdować się w pozycji górnej dla każdej z faz. Kolejno odstawiany jest układ pomiarowy (należy wyjąć żółtą blokadę), następnie przeciągnąć łączniki do pozycji dolnej - następuje zwarcie strony wtórnej przekładnika prądowego, układ pomiarowy rozarty. Wówczas mamy możliwość podłączenia wymuszania prądowego bez konieczności wypinania układu pomiarowego (pozycja c) z Rys. 5.1).

⁵⁴ Źródło: nota katalogowa

Uwaga: pomarańczowy mostek „M” jest warunkiem prawidłowego zwierania strony wtórnej przekładnika. Wszelkie manipulacje przy mostku podczas czynności pomiarowo-kontrolnych są zabronione. Zwarcie bezpieczników napięcia powoduje podanie napięcia niebezpiecznego dla zdrowi i życia na urządzenie pomiarowe (400 V).



Rys. 5.2 Schemat wewnętrzny listwy 847-988



Rys. 5.3 Widok frontu opracowanego urządzenia Smart PG⁵⁵

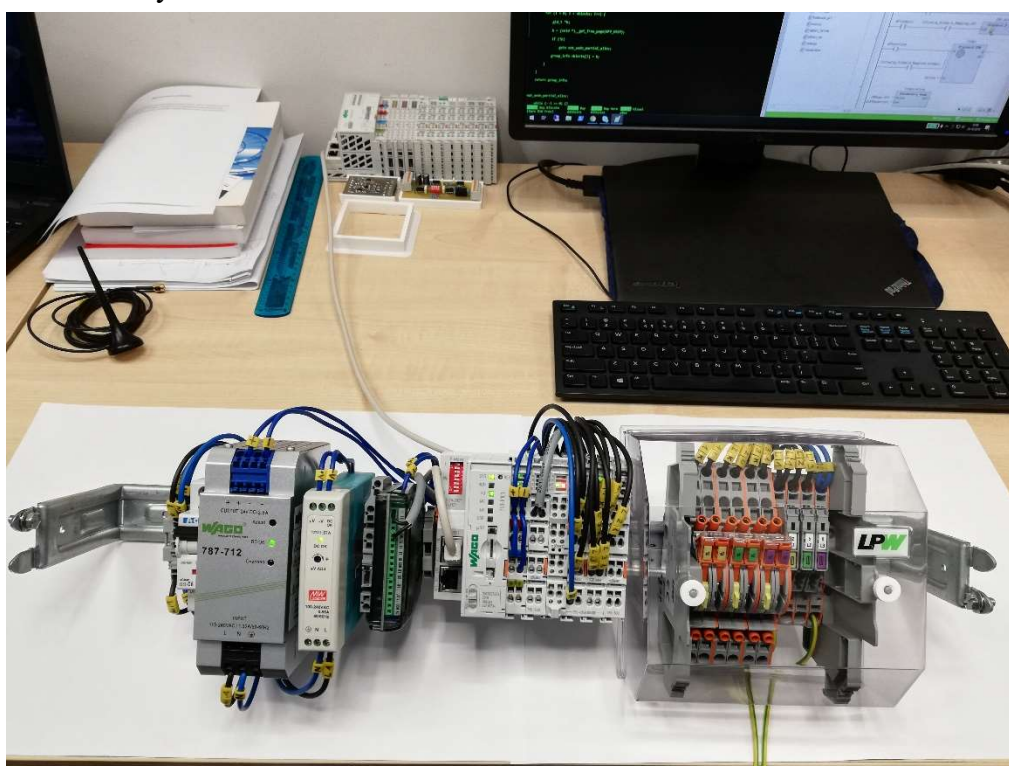
Wykaz oznaczeń:

- 1) Wyłącznik nadprądowy – C6.
- 2) Zasilacz impulsowy po stronie pierwotnej – 24 V.
- 3) Przemysłowy zasilacz na szynę DIN – 12 V.
- 4) Moduł powiadomień GSM/GPRS.
- 5) Sterownik PLC z modułami kontrolno-pomiarowymi.
- 6) Listwa kontrolno-pomiarowa.

⁵⁵ Źródło: opracowanie własne

Urządzenie pomiarowe zostało zabezpieczone poprzez wyłącznik nadprądowy o charakterystyce C 6 a (230/400 V AC), a tor pomiarowy napięcia szybkimi wkładkami topikowymi 3,15 a (500 V AC) Szczegółowe informacje dotyczące toru pomiarowego zostały przedstawione w dalszej części opracowania. Zdjęcie z testów w laboratorium przedstawiono Rys. 5.4. Wówczas zweryfikowano działanie układu pomiarowego dla prądów i napięć. Przetestowano także działanie autorskiego algorytmu kontrolowania wolumenu mocy i energii pobieranej przez urządzenia.

W pierwszej fazie napotkano szereg problemów z poprawnym działaniem algorytmu do estymacji parametrów tak aby Smart PG wykonywał predykcję zużycia energii w ciągu okien 15 minutowych. Implementacja założeń okazała się niewystarczająca co pociągało za sobą konieczność modyfikacji bloku odpowiedzialnego za wczesne wykrywanie przekroczeń mocy.



Rys. 5.4 Testy praktyczne Smart PG⁵⁶

W tym miejscu należy podkreślić wagę bloku predykcji obciążenia w odniesieniu do dalszych działań podejmowanych autonomicznie przez urządzenie. W zależności od ustawień progów dla nastaw urządzenie może podjąć następujące działania:

- tylko powiadamiać o zmianach mocy i przekroczeniach,
- wyłączyć wybrane obwody,
- przekazać komunikat do automatyki budynkowej o konieczności redukcji mocy klimatyzacji,
 - wariant pierwszy – redukcja mocy klimatyzacji,

⁵⁶ Źródło: opracowanie własne

- wariant drugi – klimatyzacja już osiągnęła graniczny minimalny poziom chłodzenia, wówczas wyłączenie wybranych obwodów.
- przełączyć wybrane odbiory na agregat prądowrczy (wariant nie sprawdzany w praktyce, konieczna modyfikacja infrastruktury rozdzielni po stronie MOCAK).

Z uwagi na ograniczone możliwości sprzętowe zastosowanego sterownika należało zmodyfikować ilość i gradację danych, ponieważ powodowały one latencję w działaniu pozostałych bloków obliczeniowych (sterownik PLC pracuje w pętli opisanej w paragrafie 3.1.1).

Instalacja Smart PG w rozdzielni głównej MOCAK

Pozytywne testy w warunkach laboratoryjnych zaowocowały zatwierdzeniem funkcjonalności i możliwością instalacji opracowanego systemu w rozdzielni głównej MOCAK Muzeum Sztuki Współczesnej w Krakowie. Na rysunkach Rys. 5.5, Rys. 5.6 przedstawiono rzeczywistą realizację Smart PG.



Rys. 5.5 Zdalny panel operatora⁵⁷

Jak przedstawiono w paragrafie 3.1.4 jako panel operatorski Smart PG wykorzystano tablet z mobilną aplikacją do komunikacji ze sterownikiem. Zastosowanie

⁵⁷ Źródło: opracowanie własne

rozwiązanie montażowe panelu umożliwia jego bezinwazyjny demontaż w przypadku prac remontowych w rozdzielni, aby zabezpieczyć go przed uszkodzeniem. Ponadto operator w trakcie prac pomiarowych w rozdzielni może zabrać tablet w dowolne miejsce (w obrębie rozdzielni) i na bieżąco weryfikować pomiary lub dokonać wyłączenia.

Do komunikacji panelu operatorskiego HMI z jednostką centralną wykorzystano bezprzewodową sieć Wi-Fi. Od routera stanowiącego punkt dostępowy do sieci LAN doprowadzono przewód Ethernet do sterownika PLC. Następnie skonfigurowano odpowiednią sieć VLAN dla działu technicznego.

Smart PG jest narzędziem niezależnym od układu pomiarowego dostawcy energii. W rozdzielni pomiar odbywa się pół-pośrednio (pomiar prądów realizowany poprzez przekładniki prądowe 1500 A / 5 A, a pomiar napięć bezpośrednio). W torze pomiarowym pomiędzy przekładnikami prądowymi i zaciskami napięciowymi znajduje się dobrana listwa kontrolna-pomiarowa. Zastosowanie listy umożliwia wymontowanie układu pomiarowego oraz sterownika bez konieczności wyłączenia całości instalacji. Opracowane urządzenie zainstalowane w docelowej lokalizacji przedstawiono na Rys. 5.6.

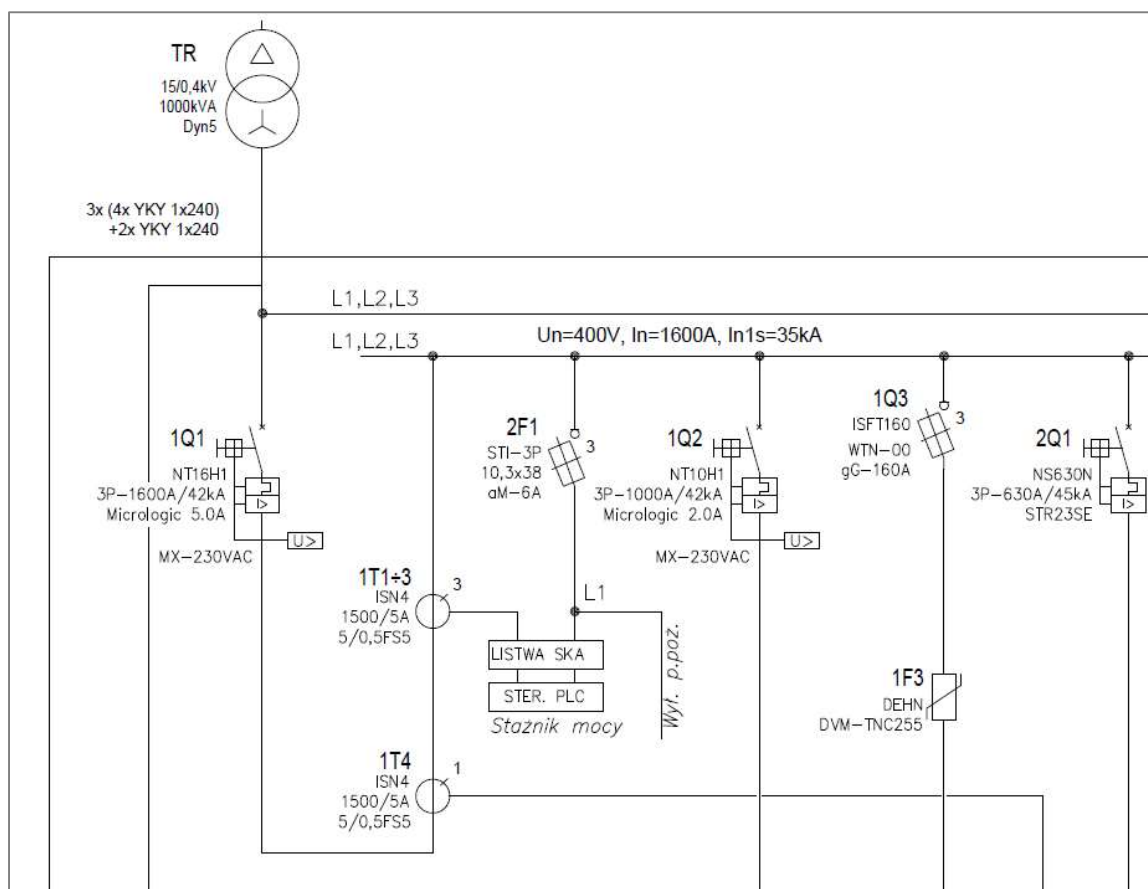


Rys. 5.6 Smart PG zainstalowany w rozdzielni MOCAR⁵⁸

Z powodu braku zgody na ujawnianie danych pomiarowych, praca nie zawiera prezentacji analizy danych podlegających akwizycji. Zarejestrowane zdarzenia i alarmy

⁵⁸ Źródło: opracowanie własne

oraz dane pomiarowe są objęte tajemnicą danych i nie mogą zostać ujawnione podmiotą trzecim.



Rys. 5.7 Wycinek schematu sieci zasilającej po zainstalowaniu Smart PG⁵⁹

Na Rys. 5.7 zaprezentowano wycinek opracowanego schematu rozdzielni w docelowej lokalizacji części sprzętowej opracowanego systemu aktywnej kontroli mocy. Całość schematu umieszczono w załączniku nr 12.

⁵⁹ Źródło: opracowanie własne

5.2. Smart EMS

Poniżej przedstawiono czynności wykonane w procesie zrealizowania założeń z podrozdziału 4.2. Proces tworzenia Smart EMS nie był obostrzony ustaleniami z podmiotami trzecimi i nie musiał być realizowany „stricte” według przyjętych założeń. Niemniej jednak realizacja głównych założeń przebiegała zgodnie wytycznymi z 4.2. Urządzenie może pracować w środowisku przemysłowym i stanowi narzędziem zarówno dla odbiorów przemysłowych jak oraz użytkowników indywidualnych i prosumentów. Smart EMS służy głównie do rejestracji parametrów elektrycznych, lecz w porównaniu do Smart PG w łatwy sposób może zostać rozbudowane o pomiar wartości nieelektrycznych t.j. temperatura, wilgotność etc. Wyposażenie układu zastosowanego komputera jednopłytkowego w 40 złącz GPIO, a także w magistralę 1-Wire, pozwalana prostą implementację szeregu dostępnych czujników na rynku np. DS18B20 (temperatura), DHT22 (temperatura i wilgotność). Opracowane rozwiązanie spełnia oczekiwania przeciętnego prosumenta, ponieważ udostępnia dane takiej wolumen, energii pobranej oddanej, aktualną informację o mocy generowanej i pobieranej, oraz bilans energii elektrycznej. Podgląd wartości odbywa się w intuicyjny sposób poprzez przeglądarki internetowe po wpisaniu odpowiedniego adresu.

W tym miejscu należy zaznaczyć, że w zależności od zastosowania utworzono trzy wersje sprzętowe oraz dwie pod względem programowym. Opis poszczególnych wersji zaprezentowano w dalszej części podrozdziału.

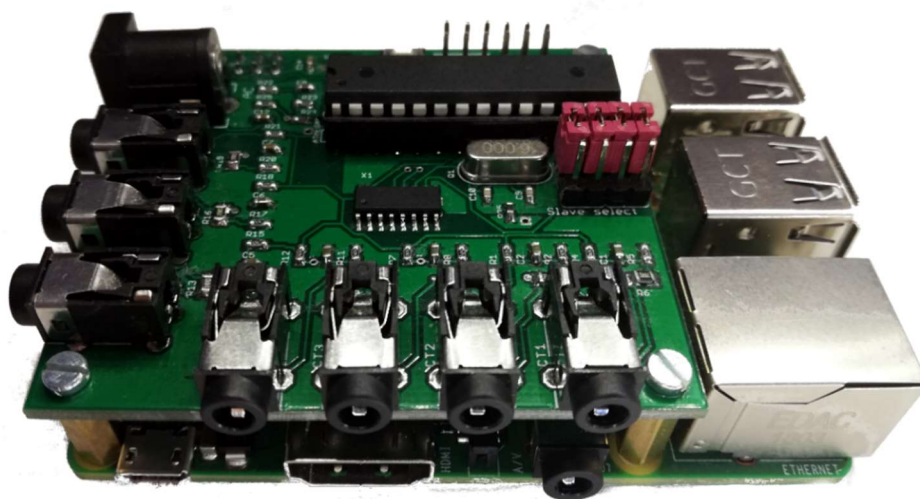
Początek realizacji urządzenia przebiegał analogicznie jak w Smart PG, Na początku przystąpiono do instalacji i konfiguracji narzędzi programistycznych i kompletowania części sprzętowej do testów laboratoryjnych. Natomiast dalszy ciąg prac miał inny przebieg. Przed przystąpieniem do testów laboratoryjnych należało skompletować i wykonać część do pomiarów elektrycznych. Do tego celu wykorzystano układy scalone przedstawione podrozdziale 4.2. Do połączenia układów z siecią zasilającą w przypadku części prądowej użyto gotowe dostępne na rynku przekładniki prądowe, natomiast do pomiaru napięcia wykorzystano transformator obniżający, który został wlutowany do dostępnych obudów uniwersalnych. W kolejnym etapie układy były testowane w wersji kombinowanej, gdzie stokowano wersje jedno i trój fazowe w celu sprawdzenia poprawności działania układu i komunikacji (Rys. 5.8). Kolejno przystąpiono do wykonania obudów dla różnych wersji sprzętowych Rys. 5.10.

Po zweryfikowaniu poprawnego działania układu testowego przystąpiono do konfiguracji lokalnej bazy danych Smart EMS, i wówczas skonfigurowano narzędzie Grafana i wypracowano pulpity wizualizacyjne.

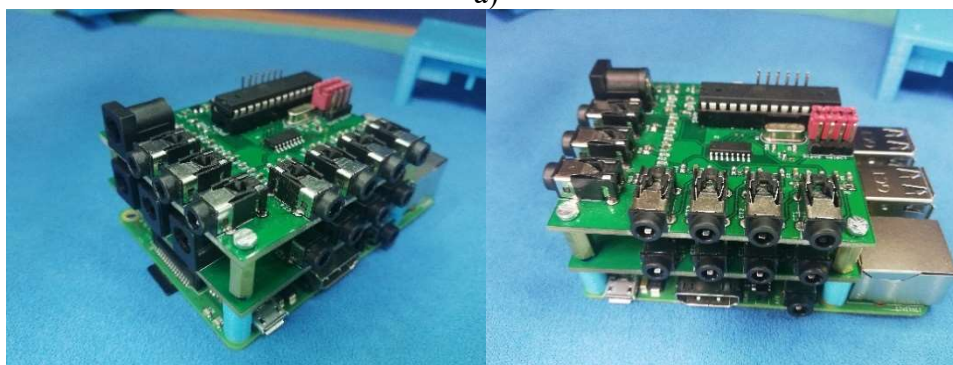
W narzędziu ustawiono poziomy dostęp:

- dostęp tylko do przeglądania – wyświetlanie stanu danych, logowanie się na koncie i generowanie raportów,

- dostęp standardowy – do wyświetlania danych połączonych z kontem, edytować skale wykresów etc. Logowanie na połączone konta i generowanie raportów, otrzymywanie skonfigurowanych powiadomień o przekroczeniach,
- dostęp dla koordynatora klastra – dostęp do modyfikacji ustawień łącz danych, możliwość tworzenia własnych pulpitów, możliwość dodania nowego użytkownika,
- dostęp administracyjny – dostęp do wszystkich funkcjonalności systemu



a)

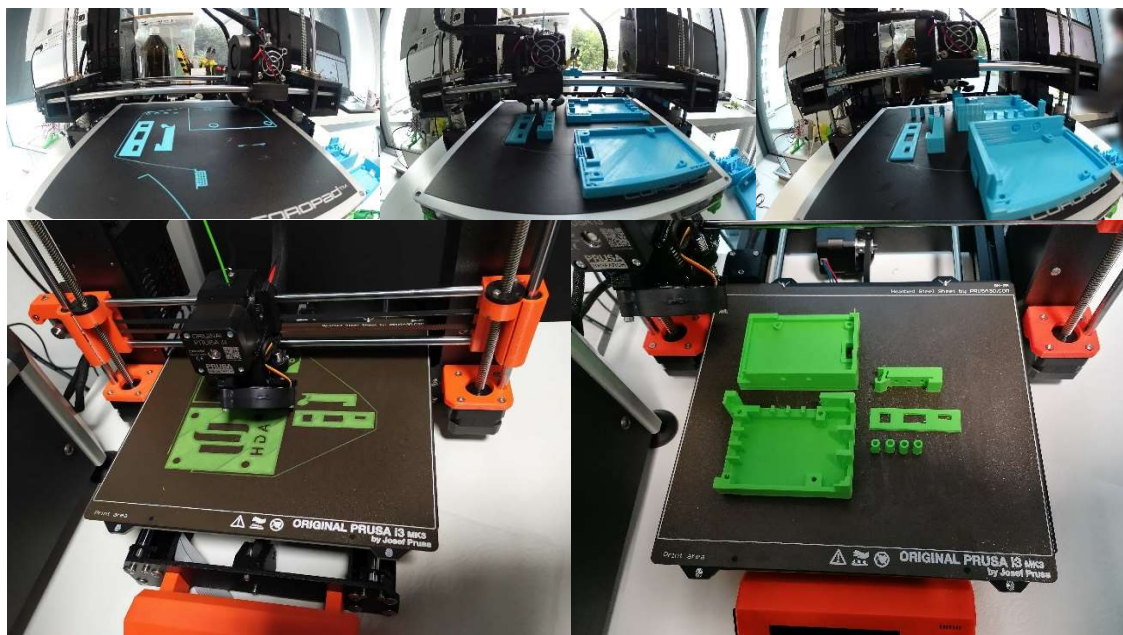


b)

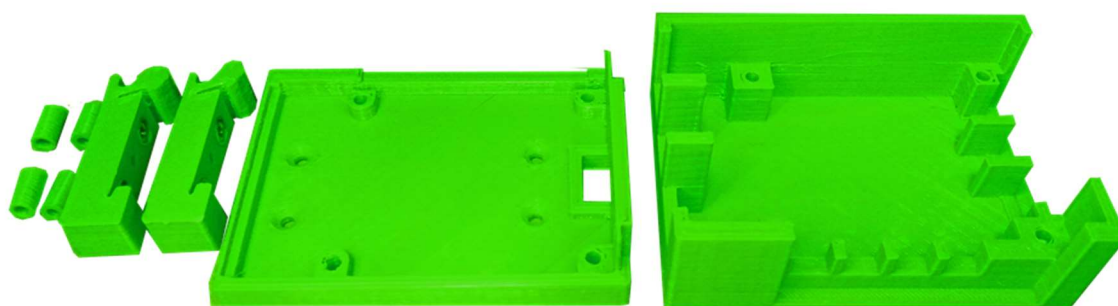
Rys. 5.8 Stokowanie układów jednofazowych i trójfazowych: a) układ do pomiaru prądów jednofazowy, b) układ do pomiarów trójfazowych 4-ro przewodowych⁶⁰

W dalszej kolejności wykonano model zaprojektowanej obudowy. Obudowa podlega ciągłym zmianą ewolucyjnym, ponieważ cały czas pojawiają się nowe koncepcje rozszerzeń. Do wykonania obudów wykorzystano drukarki 3D przedstawione w paragrafie 3.1.7, a proces druku prezentuje Rys. 5.9 Proces druku obudów Rys. 5.9, a jedną z wersji obudów Rys. 5.10. Prototyp samego urządzenia w dwóch wersjach sprzętowych przedstawiono na Rys. 5.11.

⁶⁰ Źródło: opracowanie własne



Rys. 5.9 Proces druku obudów⁶¹



Rys. 5.10 Obudowa urządzenie SEMS⁶²



a)

b)

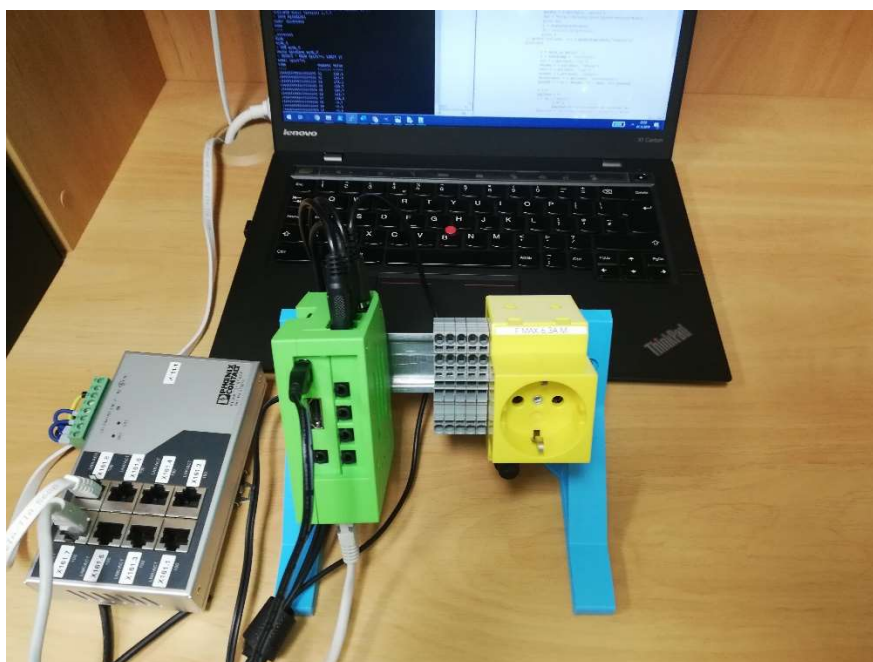
Rys. 5.11 Prototyp urządzenia Smart EMS
a) wersja 1-fazowa pomiar odpływów, b) wersja 3-fazowa⁶³

⁶¹ Źródło: opracowanie własne

⁶² Źródło: opracowanie własne

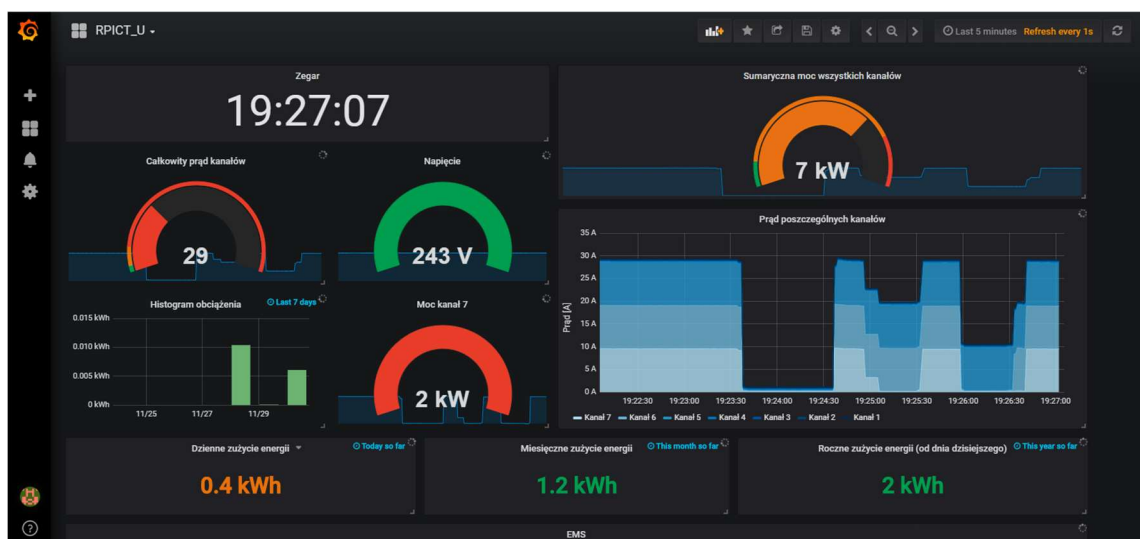
⁶³ Źródło: opracowanie własne

Docelowy model Smart EMS poddano szeregowi testów, po zweryfikowaniu działającego prototypu (Rys. 5.12) rozpoczęto prace w laboratorium z użyciem narzędzi do generowania mocy.



Rys. 5.12 Układ do badania algorytmów⁶⁴

Po uruchomieniu i sprawdzeniu właściwości pomiarowych, przystąpiono do tworzenia paneli wizualizacyjnych w oprogramowaniu Grafana. Na podstawie specyfikacji przedstawionej na końcu paragrafu 4.2.1 opracowano kilka wersji pulpitów. Pulpity są podzielne w zależności od wersji sprzętowej oraz od użytkownika (odbiorca, prosument, koordynator). Przykład opracowanego pulpitu przedstawia Rys. 5.13.



Rys. 5.13 Pulpit wizualizacyjny Smart EMS⁶⁵

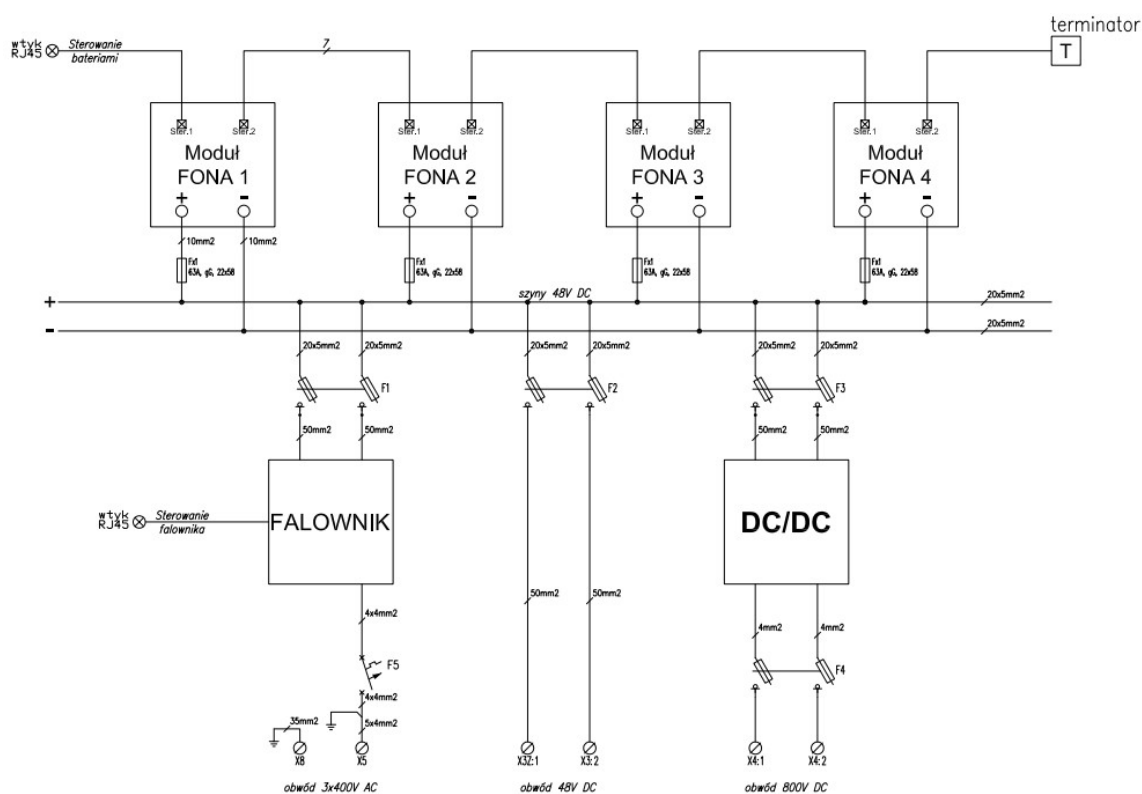
⁶⁴ Źródło: opracowanie własne

⁶⁵ Źródło: opracowanie własne

W odróżnieniu od Smart PG tym razem przeprowadzono testy współpracy urządzenia z zasobnikiem energii. Do tego celu wykorzystano magazyn bazujący na bateriach L-ion. Magazyn do testów przedstawia Rys. 5.14.



Rys. 5.14 Magazyn energii IMPACT⁶⁶



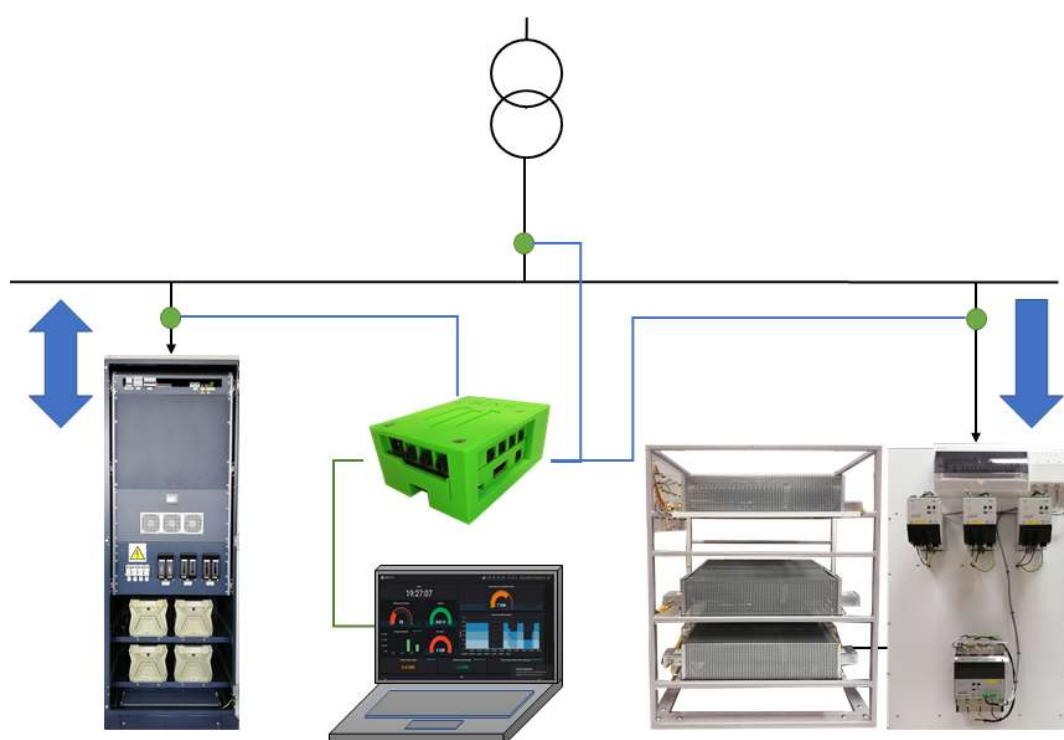
Rys. 5.15 Schemat połączeń wewnętrznych zasobnika FONALION

⁶⁶ Źródło: opracowanie własne

Magazyn energii FONALION jest przeznaczony do pracy w prosumenckich systemach energetycznych połączonych liniami 3x400 V, 50 Hz. Zasobnik obecnie ma pojemność nominalną 180 Ah i 10 kWh z możliwością rozszerzenia wolumenu gromadzonej energii do celów projektowych. Prototyp magazynu został zaprojektowany tak, aby mógł pracować z obciążeniami o stałym napięciu.

Magazyn energii składa się z następujących komponentów:

- czterech modułów akumulatorowych połączonych równolegle,
- czterokwadrantowego konwerter AC / DC 10 kW, 3-fazowy.
- komory i szyny dla przetwornicy DC / DC 10 kW, 800 V,
- zabezpieczeń szyn prądu stałego i przemiennego,
- magistrali komunikacyjnej do nadrzędnego systemu monitorowania i sterowania.

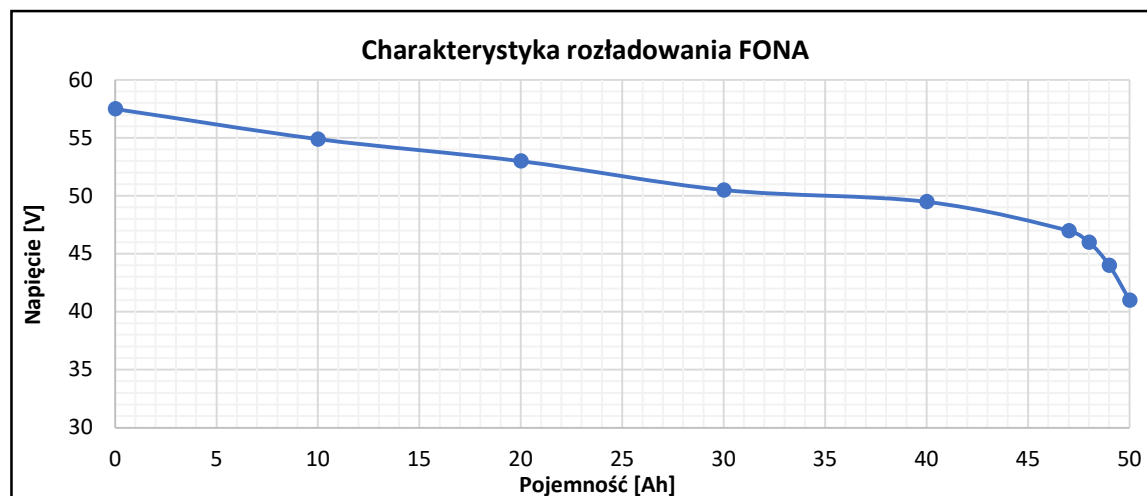


Rys. 5.16 Schemat układu pomiarowego z zasobnikiem energii elektrycznej⁶⁷

Schemat układu pomiarowego Smart EMS wraz z odbiorem i magazynem energii przedstawiono na Rys. 5.16. W przeprowadzonych badaniach jako odbiornik wykorzystano opornice podłączone do sterowników mocy (tyrystorowych). Pomiary przeprowadzono dla różnych scenariuszy działania odbiornika. Układ pracował na 3-fazowy układ symetryczny (sterownik RP3) oraz asymetryczny gdzie każda faza była sterowana niezależnie. Niestety w istniejącej konfiguracji zasobnik nie dawał możliwości przetestowania pracy „wyspowej”. Przeprowadzone pomiary umożliwiły potwierdzenie

⁶⁷ Źródło: opracowanie własne

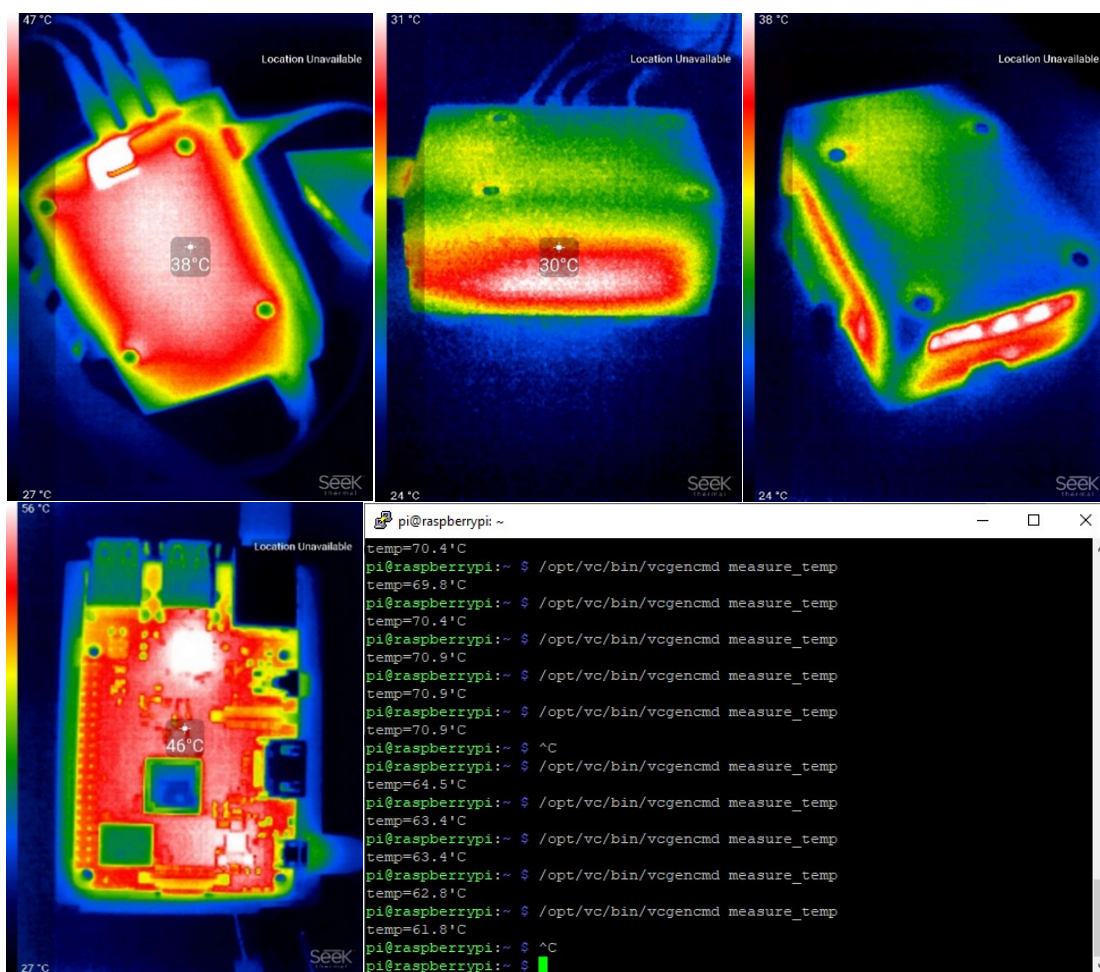
zarówno parametrów pracy modułów FONA jak i poprawnego funkcjonowania opracowane urządzenia Rys. 5.17.



Rys. 5.17 Schemat układu pomiarowego z zasobnikiem energii elektrycznej⁶⁸

Kolejno przed przystąpieniem do instalacji prototypowych urządzeń u odbiorców końcowych zdecydowano się na przeprowadzenie testów temperatury pracy układu pomiarowego. Smart EMS podczas intensywnego obciążenia procesora i dużej ilości przetwarzanych danych rozgrzewał się na powierzchni do 56 °C, natomiast wbudowany w układ scalony monitor temperatury wskazywał maksymalnie 80 °C. Dlatego przeprowadzono test nagrzewania zaprojektowanej obudowy kamerą termowizyjną. Podstawą do przeprowadzenia badania była otrzymana informacja, że urządzenie może być instalowane miejscach, gdzie inne urządzenia mają wskazaną temperaturę pracy wynoszącą do 40 °C. Wyniki testów przedstawiono na Rys. 5.18.

⁶⁸ Źródło: opracowanie własne



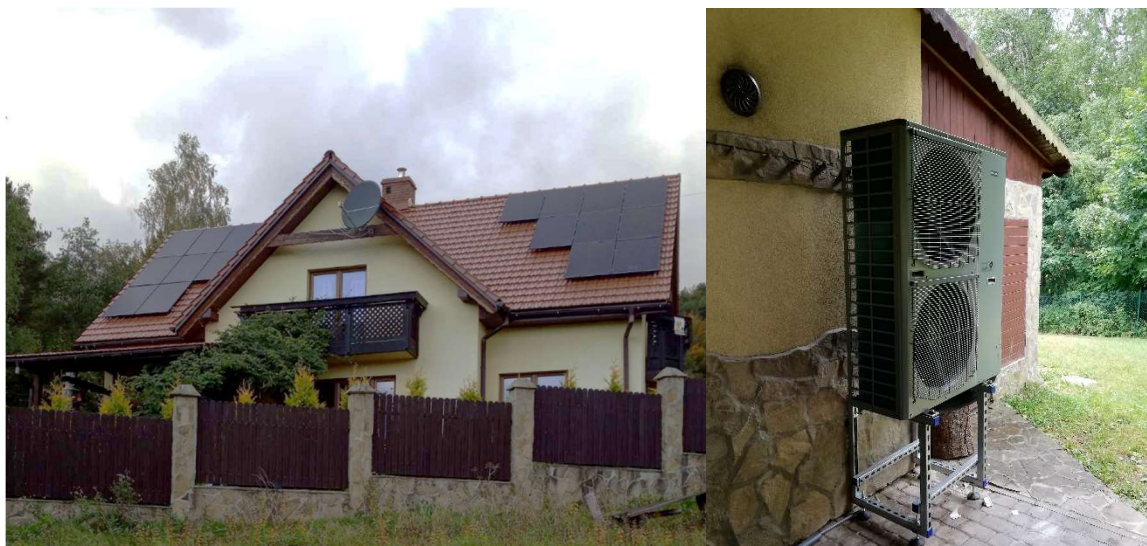
Rys. 5.18 Badania termowizyjne urządzenia Smart EMS⁶⁹

Instalacja Smart EMS u prosumentów

Pozytywne testy w warunkach laboratoryjnych skłoniły autora aby przetestować urządzenie w warunkach środowiska docelowego. Urządzenie zostało zainstalowane u dwóch prosumentów. Obaj dysponowali nowymi instalacjami fotowoltaicznymi oraz pompami ciepła. Kolejno zaprezentowano przedmiotowe instalacje. Wizerunek jest udostępniony za zgodą obu użytkowników.

- Prosument 1 – posiada nową instalację fotowoltaiczną 6 kWp oraz pompę ciepła 16 kWc (Rys. 5.19).
- Prosument 2 – nowa instalacja fotowoltaiczna o mocy 9,9 kWp i pompa ciepła 12 kWc (Rys. 5.20).

⁶⁹ Źródło: opracowanie własne



Rys. 5.19 Instalacja prosumenta I⁷⁰

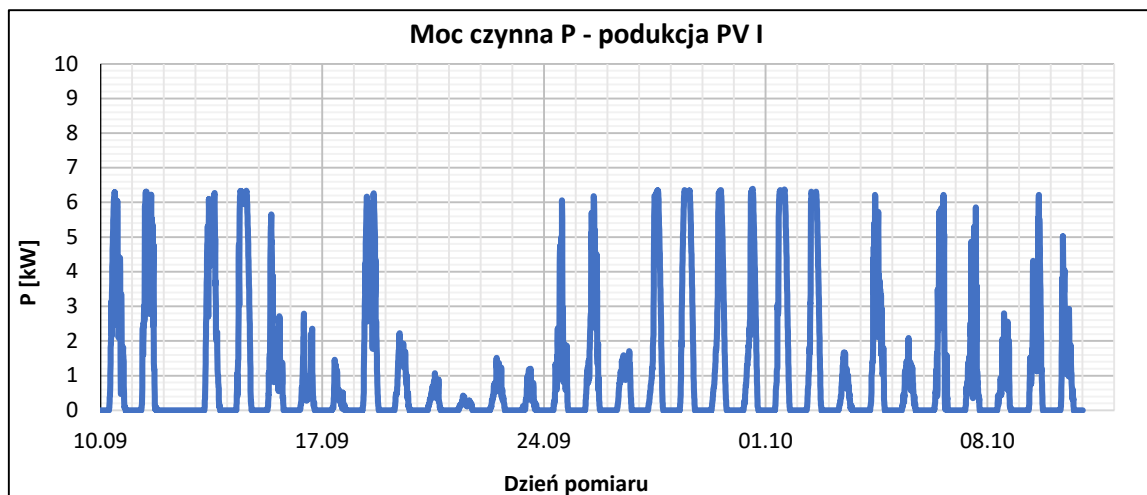


Rys. 5.20 Instalacja prosumenta II⁷¹

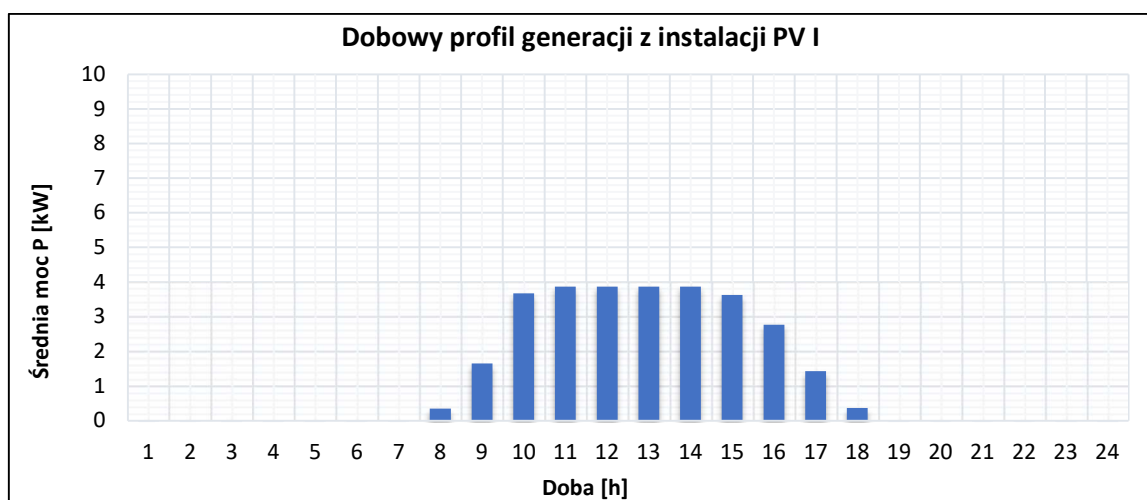
Niestety z powodu krótkiego okresu pomiarowego (nowo wykonane instalacje fotowoltaiczne oraz pompy ciepła), nie udało się zgromadzić wystarczającej ilości danych pomiarowych do analiz doboru zasobnika energii oraz zaprezentować całkowity bilans energetyczny badanych obiektów. W dalszej części zamieszczono przebiegi mocy generowanej z przedstawionych instalacji wraz z wolumenem energii oraz dobowym profilem generacji. Przebiegi zostały wyznaczone na podstawie danych zarejestrowanych przez SEMS.

⁷⁰ Źródło: opracowanie własne

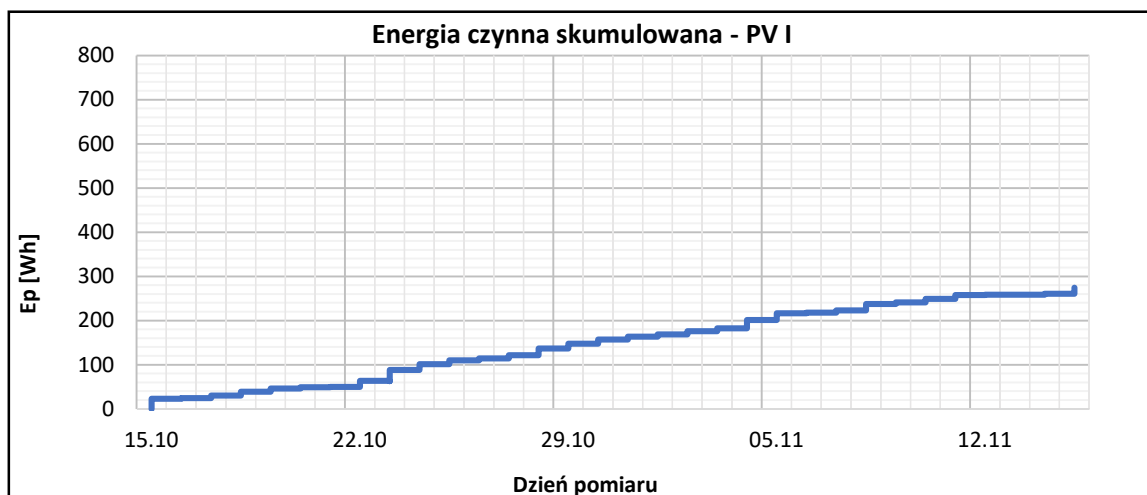
⁷¹ Źródło: opracowanie własne



Rys. 5.21 Moc czynna w badanym miesiacu - Instalacja PV I⁷²



Rys. 5.22 Dobowy profil generacji wartości maksymalne ze średnich agregowanych PV I⁷³



Rys. 5.23 Energia skumulowana w badanym miesiacu, instalacja PV I⁷⁴

⁷² Źródło: opracowanie własne

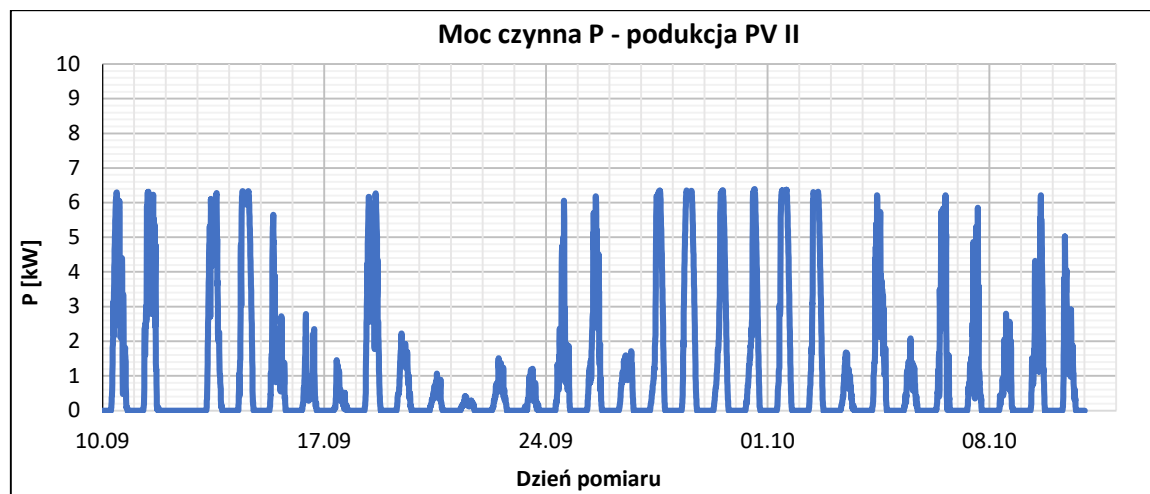
⁷³ Źródło: opracowanie własne

⁷⁴ Źródło: opracowanie własne

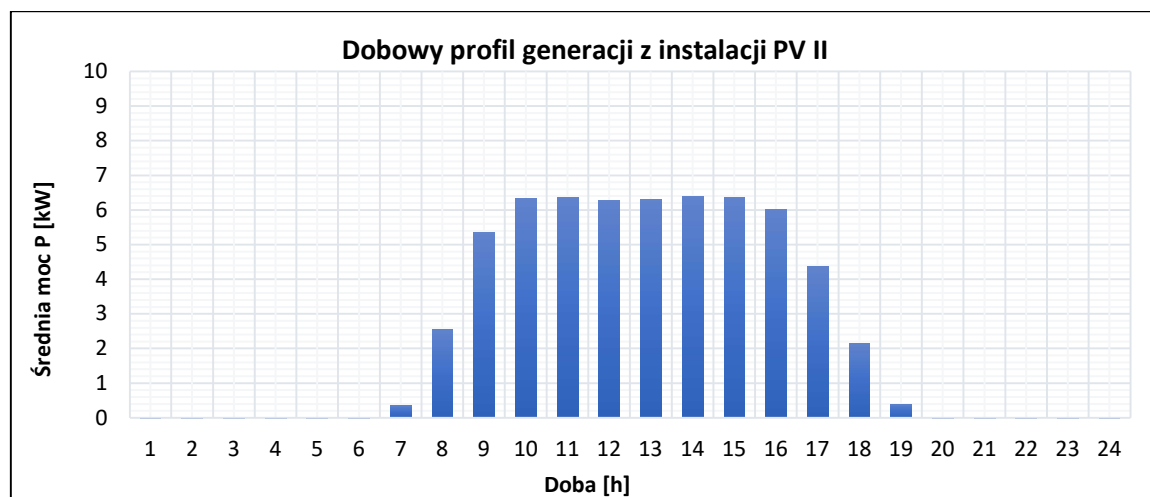
Maksymalną moc zarejestrowaną przez Smart EMS oraz miesięczną energię wyprodukowaną z instalacji PV I Przedstawiono w Tab. 5.1.

Tab. 5.1 Produkcja miesięczna instalacja PV I

Zestawienie wyników generacji dla instalacji PV I	
P_{\max} [kW]	ΣE_p [kWh]/miesiąc
3,871	274,894



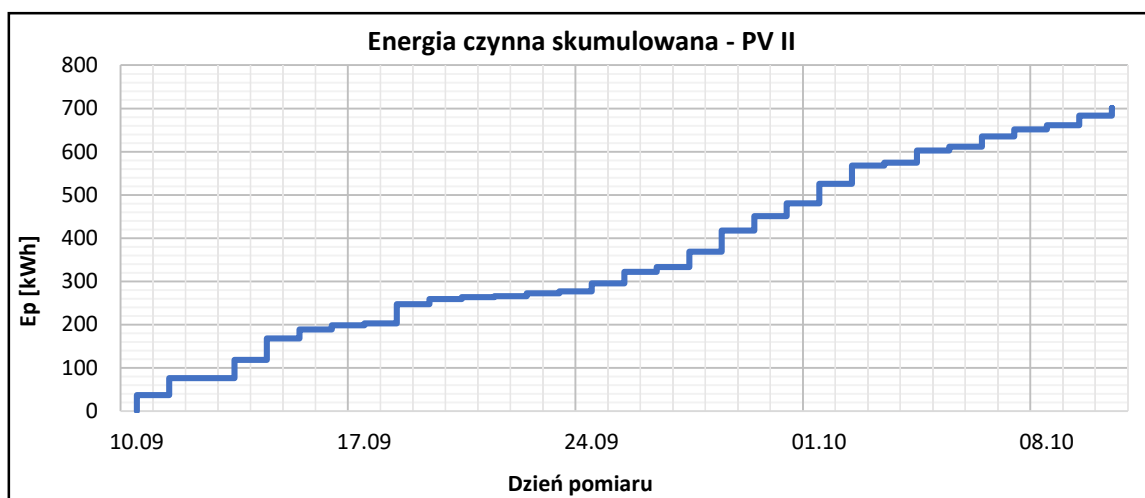
Rys. 5.24 Moc czynna w badanym miesiącu - Instalacja PV II⁷⁵



Rys. 5.25 Dobowy profil generacji, wartości maksymalne ze średnich agregowanych PV II⁷⁶

⁷⁵ Źródło: opracowanie własne

⁷⁶ Źródło: opracowanie własne



Rys. 5.26 Energia skumulowana w badanym miesiącu, instalacja PV II⁷⁷

Maksymalna moc zarejestrowana przez Smart EMS oraz miesięczna energia wyprodukowana z instalacji PV II została przedstawiona w Tab. 5.2.

Tab. 5.2 Produkcja miesięczna instalacja PV II

Zestawienie wyników generacji dla instalacji PV II	
P_{\max} [kW]	ΣE_p [kWh]/miesiąc
6,393	701,110

⁷⁷ Źródło: opracowanie własne

Podsumowanie

Sieci „inteligentne” w kontekście *Smart Grid* nieustannie się rozwijają i obejmują swoim zakresem coraz więcej dziedzin. Obecnie (2019) to nie tylko zagadnienia techniczne, ale także behawioralne i społeczne. Jedną z przyczyn eskalacji SG są rozproszone źródła energii, a konkretnie nieposkromiony rozwój mikroinstalacji OZE. Jak wspomniano na początku niniejszej pracy oprócz pozytywnych aspektów związanych z czystą energią pojawia się również wiele problemów. Duża ilość źródeł o niestabilnym charakterze produkcji jest przyczyną trudności w sterowaniu systemem elektroenergetycznym. Częstym zjawiskiem w przypadku energii elektrycznej zwłaszcza w okresie letnim jest przepływ mocy od sieci niskiego napięcia w wzwyż za przyczyną instalacji fotowoltaicznych.

W chwili obecnej poszukuje się rozwiązań jak rozprzestrzenić źródła niestabilne w sposób przemyślany i odpowiedzialny, który będzie korzystny dla wszystkich uczestników rynku. Jednym z proponowanych rozwiązań są Klastry Energii, których zadaniami nadrzędnymi są: poprawa bezpieczeństwa energetycznego, wzrost ilości nowych źródeł rozproszonych bazujących na OZE, a także wzrost świadomości społecznej w zakresie ekologii i efektywności energetycznej. Idee KE są słuszne niemniej jednak prace badawcze w projektach KlastER i RELflex wykazały brak uniwersalnych rozwiązań do zarządzania klastrami. Dostępne na rynku urządzenia i oprogramowanie każdorazowo wymagają dostosowania do indywidualnych potrzeb, a niejednokrotnie ograniczają późniejszy rozwój do konkretnych dostawców. Wyżej wspomniane projekty mają na celu wypracowanie metod i narzędzi współpracy klastrów oraz małych przedsiębiorstw KSEE.

W niniejszej pracy zrealizowano projekt budowy dwóch różnych instrumentarium typu *smart meter* do zarządzania rozproszonymi źródłami energii, jako rozwiązanie dla MŚP oraz prosumentów, a w szczególności koordynatorów klastrów. Na podstawie wiedzy eksperckiej opracowano koncept budowy obu systemów. Kolejno opracowano algorytmy do monitorowania rozproszonych źródeł i odbiorów oraz skonfigurowano bazy danych pomiarowych.

Jednym celów pracy było wykonanie aparatury do kontroli odbiorów dużej mocy. Cel zrealizowano i opracowano Smart Power Guard, który został zainstalowany w przemysłowym środowisku i współpracuje z odbiorami o mocy 300 kW energii elektrycznej. Podczas pracy napotkano na problem połączenia z siecią LAN, która nie została doprowadzona na dzień instalacji urządzenia. Problem rozwiązano poprzez uruchomienie własnego routera dostępowego i skonfigurowanie połączenia w sieci VLAN.

Realizacja założonych celów nie była by możliwa bez szerokiego przeglądu literatury na podstawie, której dokonanej analizy obecnego stanu założeń dla bilansowania OZE, a także organizacji i konfiguracji transmisji danych pomiarowych.

Drugi z opracowanych przyrządów cechuje kompaktowość, ponieważ aparatura pomiarowa, baza danych i wizualizacja do analizy mogą znajdować się na jednym małym

sprzęcie. Jednym z problemów w trakcie opracowania koncepcji było powiązanie interesów zarówno użytkowników końcowych (prosumentów) i koordynatorów. Dlatego Smart EMS może współpracować z zewnętrznym źródłem danych, ale ma możliwość akwizycji i gromadzenia ich lokalnie. Kolejnym problemem w zdalnym dostępie do urządzeń jest ich cyfrowe bezpieczeństwo. W przypadku modelu, gdzie komunikacja jest nawiązywana z zewnętrznymi jednostkami uruchomiono szyfrowane połączenie przy pomocy tunelu VPN.

Uzyskane wyniki badań zarówno w warunkach laboratoryjnych, jak i przemysłowych potwierdziły poprawne działanie założonych funkcjonalności. Niestety z powodu braku dostępności rzeczywistej instalacji współpracującej z dużym magazynem energii nie przeprowadzono testów w środowisku przemysłowym algorytmów do magazynu energii Smart EMS.

Pierwsze z urządzeń zrealizowano w 100% w komercyjnym licencjonowanym oprogramowaniu. Jako zaletę stwierdza się budowę systemu w jednym spójnym środowisku, a także możliwość skorzystania ze wsparcia technicznego. Aczkolwiek minusem jest znikoma możliwość integracji osprzętu od innych producentów oraz konieczność tworzenia całości oprogramowania praktycznie od zera. W następnej kolejności opracowano system Smart EMS, który był budowany w jak najszerszym stopniu o rozwiązania typu *open source*. Jako minus należy podać czasochłonność i brak regularnego wsparcia technicznego. Można również stwierdzić, że dla przeciętnego użytkownika wykorzystywane narzędzia były mniej „przyjazne” niż komercyjne rozwiązania. Jako wadę i zaletę należy wymienić otwartość kodu źródłowego wykorzystywanych rozwiązań. Z jednej strony oznaczać może szybką wykrywalność błędów i zagrożeń, natomiast z drugiej rozwiązania tego typu mogą być podatne na zero-day exploit⁷⁸

Na zakończenie warto powiedzieć, że prace nad urządzeniem Smart PG zostały zakończone sukcesem, lecz nie jest ono nadal rozwijane. Jednakże Smart EMS po wytworzeniu w pełni działającego prototypu jest rozwijany w modelu ewolucyjnym, na dzień zakończenia niniejszej pracy prowadzone są testy terenowe.

Opracowany system otrzymał pozytywne opinie potencjalnych użytkowników co zaowocowało zaproszeniem na rozmowę przez fundusz inwestycyjny na początku roku 2020. Autor obecnie planuje budowę modułów, plug&play do pomiarów nieelektrycznych, co dodatkowo rozszerzy funkcjonalność. Plany dotyczą również opracowania wydajnego systemu chłodzenia, z uwagą na tendencję temperatury w nowszych wersjach komputera jednopłytkowego.

⁷⁸ Jest to wydanie produktu z luką bezpieczeństwa, o której nie ma informacji producent i systemy antywirusowe.

Literatura

- [1] Przygodzki M.: Generacja rozproszona jako aktywny uczestnik zarządzania pracą KSE, *Logistyka* 2015, s. 4037-4044, 2.
- [2] Czapaj R., Rzepka P., Szablicki M.: Typowanie zmiennych objaśniających przy wykorzystaniu zautomatyzowanych metod statystycznych jako sposób optymalizacji wyboru metody estymacji szczytowego dobowego obciążenia KSE, *Konferencja Naukowa Prognozowanie w Elektroenergetyce, Podlesice 2016*.
- [3] Sołtysik M., Wilczyński A.: Warunki optymalnego doboru uczestników grup bilansujących, *Materiały VI Konferencji naukowo-technicznej Sieci Elektroenergetyczne w Przemśle i Energetyce*, wrzesień 2008, s. 259-266.
- [4] T. Szymusiak, *Prosumenci-Prosumpcja-Prosumeryzm. Ekonomiczne oraz społeczne korzyści prosumpcji na przykładzie Polski oraz Niemiec (podejście naukowe)*, Wydawnictwo Bezkresy Wiedzy, 2015.
- [5] Ustawa z dnia 20 lutego 2015 r. o odnawialnych źródłach energii (Dz. U. z 2015 r., poz. 478, 2365, z 2016 r. poz. 925, 1579).
- [6] Ustawa z dnia 19 lipca 2019 r. o zmianie ustawy o odnawialnych źródłach energii oraz niektórych innych ustaw (Dz. U. z 2019 r., poz. 1525).
- [7] Informacja (nr 70/2016) Prezesa Urzędu Regulacji Energetyki w sprawie średniej ceny sprzedaży energii elektrycznej na rynku konkurencyjnym w III kwartale 2016 roku, 21.12.2016 r.
- [8] <http://www.tge.pl>.
- [9] Informacja Prezesa Urzędu Regulacji Energetyki Nr 2/2017 z dnia 3.01.2017 r,
- [10] Informacja Prezesa Urzędu Regulacji Energetyki Nr 4/2017 z dnia 3.01.2017 r,
- [11] Informacja Prezesa Urzędu Regulacji Energetyki Nr 5/2017 z dnia 3.01.2017 r,
- [12] *Ceny energii elektrycznej dla odbiorców przemysłowych i indywidualnych. Etap I.* Agencja Rynku Energii SA. Warszawa, marzec 2016.
- [13] *Raporty miesięczne z RB i KSE* [Online, dostęp: 9.09.2017], dostęp online: http://www.pse.pl/index.php?modul=8&y=2016&m=12&id_rap=212
- [14] Ustawa z dnia 20 lutego 2015 r. o odnawialnych źródłach energii (Dz. U. poz. 478 i 2365)
- [15] *Komunikat Komisji do Parlamentu Europejskiego. Rady. Europejskiego Komitetu Ekonomiczno-Społecznego i Komitetu Regionów.* Bruksela. 13.10.2014 r.
- [16] *KONCEPCJA FUNKCJONOWANIA KLASTRÓW ENERGII w POLSCE*, Opracowanie, na zlecenie Skarbu Państwa – Ministra Energii, wykonało Konsorcjum w składzie: Krajowa Agencja Poszanowania Energii S.A, WiseEuropa – Fundacja Warszawski Instytut Studiów Ekonomicznych i Europejskich Atmoterm S.A, przy udziale Krajowy Instytut Energetyki Rozproszonej.
- [17] *Klaster energii - koncepcja funkcjonowania* [Online, dostęp 29.08.2019] dostęp online: <https://www.brewa.pl/strefa-wiedzy/klaster-energii-koncepcja-funkcjonowania.html>
- [18] Hodana M., Holtzer G., Kalandyk K., Szymańska A., Szymański B, Zymankowska-Kumon S., *Odnawialne źródła energii - Poradnik*, Kraków 2012.

- [19] Ustawa z dnia 10 kwietnia 1997 r. Prawo energetyczne. Dz. U. nr 54. poz, 348 z późniejszymi zmianami
- [20] Berent-Kowalska G., Kacprowska J., Moskal I., Jurgaś A., Energia ze źródeł odnawialnych w 2015r., Warszawa 2016.
- [21] Saboori H., Mohammadi M., Taghe R.: Virtual power plant (VPP), definition, concept, components and types, Power and Energy Engineering Conference (APPEEC), 2011 Asia-Pacific.
- [22] Elektrownia wirtualna [Online, dostęp 03,06,2019], dostęp online: https://pl.wikipedia.org/wiki/Elektrownia_wirtualna
- [23] Magulski R., Koncepcja funkcjonowania wirtualnej elektrowni integrującej rozproszone źródła energii jako dostawcy usług systemowych, AKTUALNE PROBLEMY w ELEKTROENERGETYCE APE'17, Jastrzębia Góra, 7-9 czerwca 2017.
- [24] Sahand Ghavidel, Li Li, Jamshid Aghaei, Tao Yu, Jianguo Zhu: a Review on the Virtual Power Plant: Components and Operation Systems, 2016 IEEE International Conference on Power System Technology (POWERCON), Wollongong, 28 Sept.-1 Oct. 2016.
- [25] Siewierski T., Szykowski M.: Zasobniki energii elektrycznej i bilansowanie odnawialnych źródeł energii, Elektroenergetyka nr 3-4, pp. 65–75, 2012
- [26] P. Gawrysiak, Cyfrowa rewolucja: rozwój cywilizacji informacyjnej, Wydawnictwo Naukowe PWN, 2008.
- [27] Hanzelka Z., Firlit A., Elektrownie ze źródłami odnawialnymi zagadnienia wybrane, Wydawnictwo AGH, Kraków 2015.
- [28] Bakari K. El., Myrzik J.M.A., Kling W.L., Prospects of a Virtual Power Plant to control a cluster of Distributed Generation and Renewable Energy Sources, Universities Power Engineering Conference (UPEC), 2009 Proceedings of the 44th International
- [29] Możliwości świadczenia usług regulacyjnych przez generację rozproszoną, J,G, Korpikiewicz, L, Bronk, T, Pakulski, 2013
- [30] Java DataBase Connectivity [Online, dostęp: 1.09.2019]
https://pl.wikipedia.org/wiki/Java_DataBase_Connectivity
- [31] ODBC [Online, dostęp: 1.09.2019], dostęp online: <https://pl.wikipedia.org/wiki/ODBC>
- [32] <http://www.powerhub.com/> [Online, dostęp: 04.06.2018]
- [33] Open Source Licenses - Open Source Initiative". [Online, dostęp 23.10.2019 r.]
<http://opensource.org/licenses>
- [34] K. Ziętek, „Wolny i Otwarty w świecie biznesu. Możliwości wykorzystania rozwiązań typu FOSS”. [Online, dostęp 23.10.2019 r.],
<http://kis.pwszchelm.pl/publikacje/IV/Zietek.pdf>
- [35] 111+ Linux Statistics and Facts – Linux Rocks!, [Online, dostęp 24.10.2019 r.],
<https://hostingtribunal.com/blog/linux-statistics/#gref>
- [36] iPhone vs Android - kto rządzi na rynkach? [Online, dostęp 24.10.2019 r.],
<https://www.pcworld.pl/news/iPhone-vs-Android-kto-rzadzi-na-rynkach,412131.html>

Spis tabel i rysunków

Rys. 2.1 Poglądowa koncepcja Smart Grid.....	14
Rys. 2.2 Schemat ogólny systemu Smart Meter na przykładzie systemu monitorowania jakości dostaw energii elektrycznej.....	16
Rys. 2.3 Strefy energetyczne wiatru na obszarze Polski,.....	21
Rys. 2.4 Średnioroczne nasłonecznienie dla obszaru Polski.....	22
Rys. 2.5 Podział magazynów w zależności od technologii.....	24
Rys. 3.1 Sterownik PLC PFC100.....	26
Rys. 3.2 Moduł pomiaru mocy 750-495	27
Rys. 3.3 Moduł komunikacji z siecią LonWorks 753-648.....	28
Rys. 3.4 Moduł komunikacji i powiadomień SMS, GPRS-A.....	29
Rys. 3.5 Listwa kontrolno-pomiarowa 847-998.....	30
Rys. 3.6 Raspberry Pi 3B+	31
Rys. 3.7 Drukarki 3D w technologii FDM: a) Prusa i3 MK3S, b) Anet A6.....	32
Rys. 3.8 Kamera Seek Thermal Pro	33
Rys. 3.9 Widok okna oprogramowania e!COCKPIT.....	34
Rys. 3.10 Okno aplikacji aplikacji GX Soft.....	35
Rys. 3.11 Okna środowiska wykorzystanych do kodu Python: a) IDLE, b)Notepad++.....	36
Rys. 3.12 Okno programu AutoCAD LT	37
Rys. 3.13 Okno środowiska SOLIDWORKS Premium.....	38
Rys. 3.14 Okno programu Ultimaker Cura	39
Rys. 3.15 Okno wiersza poleceń InfluxDB.....	40
Rys. 3.16 Okno Grafana w przeglądarce Internetowej	41
Rys. 4.1 Projekt funkcjonowania Smart PG.....	45
Rys. 4.2 Schemat aplikacyjny wraz z podziałem na bloki funkcjonalne	47
Rys. 4.3 Schemat blokowy konstrukcji Smart PG	47
Rys. 4.4 Wybrane okna wizualizacji projektowanej wizualizacji: a) MENU, b) START, c)ANALIZA, d) NASTAWY, e) wybór zapisu wybranych harmoniczných.....	55
Rys. 4.5 Okno aplikacji Data Plotter WAGO	55
Rys. 4.6 Koncepcja funkcjonowania SEMS	58
Rys. 4.7 Schemat aplikacyjny wraz z podziałem na bloki funkcjonalne	59
Rys. 4.8 Schemat blokowy konstrukcji Smart EMS	60
Rys. 4.9 Koncepcja pulpitu wizualizacji Smart EMS	74
Rys. 4.10 Układ scalony modułu pomiaru wartości elektrycznych: a) proces projektowania SOLIDWORKS PCB, b) układ dedykowany do audytów rozdzielni, c) układ do bilansowania międzyfazowego.....	75

Rys. 4.11 Model obudowy dla Smart EMS.....	76
Rys. 4.12 Osprzęt do pomiarów: a) transformator obniżający napięcie, b) przekładnik prądowy	76
Rys. 5.1 Położenie łączników toru prądowego 282-870: a) robocze, b) położenie pośrednie – wstępne zwarcie strony wtórnej przekładnika, bez przerywania układu pomiarowego, c) położenie testowe	78
Rys. 5.2 Schemat wewnętrzny listwy 847-988	79
Rys. 5.3 Widok frontu opracowanego urządzenia Smart PG.....	79
Rys. 5.4 Testy praktyczne Smart PG.....	80
Rys. 5.5 Zdalny panel operatora.....	81
Rys. 5.6 Smart PG zainstalowany w rozdzielni MOCÁK	82
Rys. 5.7 Wycinek schematu sieci zasilającej po zainstalowaniu Smart PG	83
Rys. 5.8 Stokowanie układów jednofazowych i trójfazowych: a) układ do pomiaru prądów jednofazowy, b) układ do pomiarów trójfazowych 4-ro przewodowych.....	85
Rys. 5.9 Proces druku obudów	86
Rys. 5.10 Obudowa urządzenie SEMS	86
Rys. 5.11 Prototyp urządzenia Smart EMS a) wersja 1-fazowa pomiar odpływów, b) wersja 3-fazowa	86
Rys. 5.12 Układ do badania algorytmów	87
Rys. 5.13 Pulpit wizualizacyjny Smart EMS	87
Rys. 5.14 Magazyn energii IMPACT.....	88
Rys. 5.15 Schemat połączeń wewnętrznych zasobnika FONALION	88
Rys. 5.16 Schemat układu pomiarowego z zasobnikiem energii elektrycznej	89
Rys. 5.17 Schemat układu pomiarowego z zasobnikiem energii elektrycznej	90
Rys. 5.18 Badania termowizyjne urządzenia Smart EMS	91
Rys. 5.19 Instalacja prosumenta I	92
Rys. 5.20 Instalacja prosumenta II	92
Rys. 5.21 Moc czynna w badanym miesiącu - Instalacja PV I	93
Rys. 5.22 Dobowy profil generacji wartości maksymalne ze średnich agregowanych PV I..	93
Rys. 5.23 Energia skumulowana w badanym miesiącu, instalacja PV I.....	93
Rys. 5.24 Moc czynna w badanym miesiącu - Instalacja PV II.....	94
Rys. 5.25 Dobowy profil generacji, wartości maksymalne ze średnich agregowanych PV II	94
Rys. 5.26 Energia skumulowana w badanym miesiącu, instalacja PV II	95
Tab. 4.1 Wykaz mierzonych parametrów elektrycznych	46
Tab. 5.1 Produkcja miesięczna instalacja PV I	94
Tab. 5.2 Produkcja miesięczna instalacja PV II.....	95