SPIE Energotest

Fundusze Europejskie Program Regionalny
Rzeczpospolita Polska
Śląskie.
Unia Europejska Europejski Fundusz Rozwoju Regionalnego

Michał Hanusek
SPIE Energotest

Konwerter protokołów EKM Smart

EKM Smart protocol converter

W dzisiejszych czasach, wraz z dynamicznym rozwojem technologii, przemysł również zmienia się i ewoluuje. Coraz większą wagę przykłada się do automatyzacji procesów przemysłowych. Jednym z kluczowych aspektów w tym zakresie jest stosowanie różnych protokołów komunikacyjnych, które umożliwiają wymianę danych między urządzeniami i systemami akwizycji danych. Ze względu na zróżnicowanie często konieczne jest stosowanie konwerterów protokołów, które umożliwiają przesyłanie informacji między urządzeniami pracującymi w różnych systemach.

Konwerter protokołów EKM Smart to nowoczesne rozwiązanie, które oferuje nie tylko konwersję między różnymi protokołami, ale także funkcje związane z IIoT oraz Industry 4.0, a także protokołem IEC-61850. Jedną z kluczowych cech tego urządzenia jest posiadanie REST API, co pozwala na integrację z systemami monitorującymi pracę urządzeń oraz proces przemysłowy.

Konwersja protokołów w sieciach przemysłowych

Konwersja protokołów umożliwia komunikację między urządzeniami, które wykorzystują różne protokoły poprzez zmianę formatu danych, tak aby przesyłane informacje były zgodne z wymaganiami protokołu docelowego. Często konwersja nie dotyczy tylko samego protokołu, ale również warstw fizycznych, na których operują urządzenia np. RS-232, RS-485, Ethernet.

Protokół komunikacyjny zgodny z normą IEC-61850

Celem pracy sieci przemysłowych jest dostarczanie danych procesowych oraz sterowanie procesem. Główne wymagania, którym mają sprostać to niezawodność oraz determinizm czasowy. Oczekuje się, aby system był bezpieczny, a dane pozyskiwane przez niego były użyteczne. Z powodu transformacji z energetyki konwencjonalnej do energetyki rozproszonej można zauważyć umacnianie się pozycji protokołu IEC-61850. Protokół ten został specjalnie zaprojektowany do wymiany danych w energetyce.

Główne założenia zawarte w normie IEC-61850 to m.in.:

- interoperacyjność – zapewnienie bezproblemowej komunikacji między urządzeniami różnych producentów,
- abstrakcyjny model danych – norma skupia się na funkcjach, a nie na fizycznych urządzeniach,
- długi okres funkcjonowania opracowanych standardów,
- ograniczenie wykorzystania osprzętu – wymiany danych przez sieć, redukcja tradycyjnych połączeń dwustanowych.

Norma wymaga utworzenia modelu danych (*data model*), który jest zbiorem wszystkich parametrów danego urządzenia oraz zestawu danych (*data set*), które mają być przesyłane do innych urządzeń. Informacje między urządzeniami mogą być wymieniane na kilka sposób:

- raporty,
- wiadomości GOOSE (*Generic Object Oriented Substation Event*),
- wiadomości Sampled Values.

Norma wprowadza język SCL (ang. Substation Configuration Description Language), oparty na standardzie XML, formalizujący proces konfiguracji urządzeń w systemie. Konfiguracja w zależności od zastosowania może być reprezentowana przez różne, poniżej wymienione pliki [3].

- **SSD** (*System Specification Description*) – jest to schemat stacji energetycznej oraz wstępna definicja funkcji. Plik SSD jest używany jako punkt wyjścia do projektowania systemu zgodnego z normą IEC-61850.
- **ICD** (*IED Capability Description*) – dostarczany przez producenta urządzenia; zawiera opis urządzenia. Opisuje, jakie funkcje i usługi są dostępne w danym IED.
- **CID** (*Configured IED Description*) – zawiera konfigurację konkretnego urządzenia w stacji, włączając w to informacje, takie jak punkty pomiarowe, punkty sterujące, parametry konfiguracyjne, połączenia między urządzeniami itp.,
- **SCD** (*Substation Configuration Description*) – zawiera konfigurację stacji energetycznej.

Zastosowanie konwerterów protokołów

Obecnie w użyciu jest kilkadziesiąt przemysłowych protokołów komunikacyjnych. W przypadku łączenia systemów lub urządzeń, używających różnych protokołów komunikacyjnych lub mediów transmisyjnych, konieczne może być użycie urządzenia dopasowującego – konwertera protokołów. W starszych instalacjach OT (Operational Technology) często stosowano magistralę RS-485, w nowszych powszechnie stosuje się sieć Ethernet. Konwerter EKM3 pozwala łączyć te dwa media i wspiera wymianę danych pomiędzy nimi. Aby móc włączyć urządzenie do sieci Ethernet i skomunikować się z nim z poziomu systemu akwizycji danych można zastosować konwerter protokołów. W przeciwieństwie do kosztownej modernizacji – wymiany urządzeń na dostosowane do pracy w sieci Ethernet, konwerter protokołów umożliwi uzyskanie tego celu niskim kosztem.

Konwerter protokołów EKM Smart

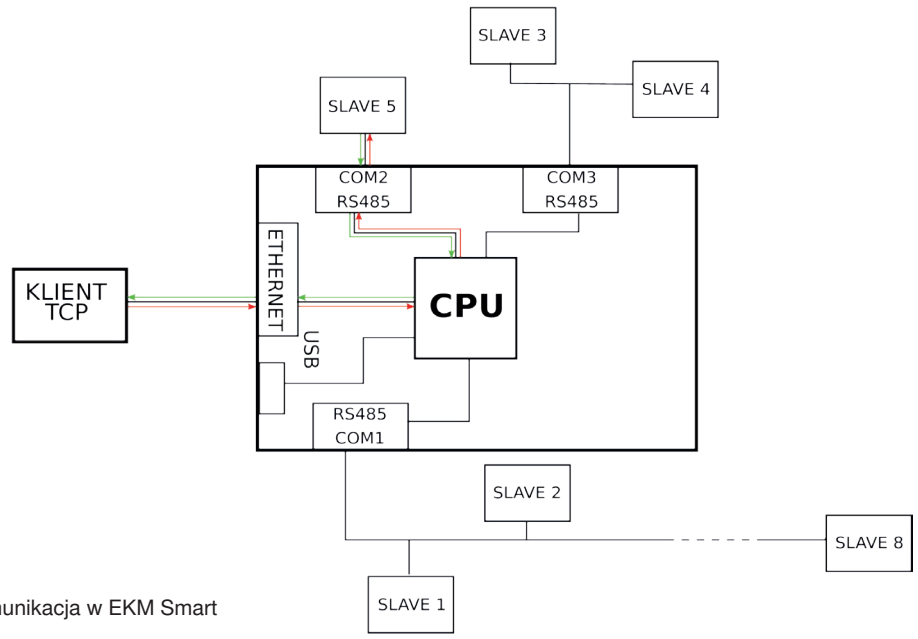
W urządzeniu zastosowano webowy interfejs graficzny konfiguratora, spełniający aktualne wymogi bezpieczeństwa informatycznego, których nie spełniał jego poprzednik. Konwerter EKM Smart jest następcą szeroko stosowanego na obiektach przemysłowych urządzenia EKM2. Dzięki niezmienionej obudowie i lokalizacji portów szeregowych, wymiana zainstalowanych u klientów EKM2 na EKM Smart jest szybka i łatwa. Podobnie jak poprzednik, posiada zaczep umożliwiający montaż na szynie DIN. Urządzenie ma czytelny interfejs użytkownika dostępny z przeglądarki internetowej, co umożliwia łatwe i intuicyjne konfigurowanie urządzenia. Użytkownik może również zaktualizować oprogramowanie urządzenia za pomocą tego samego kanału dostępu. Oprogramowanie wbudowane urządzenia zostało oparte na Embedded Linux, języku programowania Rust oraz frameworku Vue.js.

Język Rust jest nowym i innowacyjnym językiem programowania, ale już zdążył zyskać dużą popularność ze względu na swoje zalety. Jego najważniejszymi cechami są: bezpieczeństwo, wysoka wydajność, łatwość w utrzymaniu kodu i aktywna społeczność. Szczególnie istotną przewagą języka Rust nad powszechnie stosowanymi w systemach wbudowanych C i C++ jest bezpieczne zarządzanie pamięcią. Firmy Microsoft i Google przeanalizowały błędy występujące w ich oprogramowaniu i stwierdziły, że problemy z bezpieczeństwem pamięci oprogramowania leżą u podstaw około 70% luk w zabezpieczeniach [1, 2].

Rysunek 1 przedstawia komunikację w EKM Smart. Linia ***A*** reprezentuje zapytanie, linia ***B*** odpowiedź.

Urządzenie może działać w jednym z trybów pracy:

- tryb transparentny,
- tryb bramki Modbus RTU/Modbus TCP,
- tryb bramki Modbus RTU/IEC-61850.



Rys. 1. Komunikacja w EKM Smart

Tryb transparentny

W trybie transparentnym urządzenie pełni funkcję stacji nadrzędnej (*master*) podłączonej do portu szeregowego oraz serwera TCP w sieci TCP/IP. Cykl wymian wygląda następująco: klient TCP wysyła pakiet TCP, pakiet zostaje odebrany i przetworzony przez urządzenie, dane z pakietu zostają wyłuskane, a następnie wysłane na magistralę szeregową. Następnie na porcie szeregowym następuje odczyt, surowe dane zostają upakowane w pakiet TCP i odesłane do klienta TCP.

Tryb bramki Modbus

W tym trybie urządzenie dokonuje konwersji, a cykl wymian jest bardzo podobny do trybu transparentnego z tą różnicą, że ramka jest konwertowana z Modbusa TCP do Modbusa RTU i na odwrót. Konwerter wykonuje również walidację sumy kontrolnej CRC-16 (*Cyclic Redundancy Code*).

Dodatkowo w tym trybie można włączyć funkcję restrykcyjnej kontroli zapytań i odpowiedzi Modbus. Jeśli ta funkcja jest aktywna bramka sprawdza każde zapytanie, tzn. że każda część ramki musi być zgodna z protokołem Modbus. Standardowo funkcja ta jest nieaktywna, użytkownik może ją aktywować za pośrednictwem interfejsu webowego.

Mapowanie ID

Podczas integracji systemów działających na obiektach przemysłowych czasem istnieje konieczność niejako ukrycia wielu urządzeń za bramką Modbus. Wtedy można zastosować tzw. mapowanie identyfikatorów. Polega ono na podmianie przez bramkę adresu ID w ramce modbus na inny identyfikator podany w konfiguracji przez użytkownika. Urządzenie umożliwia dodanie wiązań ID przez interfejs webowy.

Tabela 1

Mapowanie identyfikatorów

RTU ID	TCP ID
1	2
5	99

Tabela 1 przedstawia przykłady mapowania identyfikatorów. Kolumna **RTU ID** przedstawia ID, jakie będzie występować w sieci RS-485. Natomiast kolumna **TCP ID** w sieci TCP/IP.

Bramka Modbus RTU/IEC-61850

Konwerter EKM Smart posiada zaimplementowaną usługę konwersji protokołu Modbus RTU na protokół zgodny z normą IEC-61850. Uściślając, urządzenie pełni funkcję serwera MMS działającego w sieci Ethernet oraz urządzenia nadrzędnego (*master*) w sieci RS-485. Obecna implementacja posiada jedynie funkcję Modbus RTU Master. Jednak system został zaprojektowany i zaimplementowany tak, aby istniała możliwość rozbudowy o kolejne protokoły działające w sieci RS-485.

Definicja zmiennej

Projektując system konwersji danych operujących na protokołach IEC-61850 oraz Modbus napotkano problem definicji zmiennej oraz tworzenia powiązań między nimi (mapowania). Protokół Modbus dostarcza abstrakcji, takiej jak rejestr. Natomiast w nomenklaturze normy IEC-61850 istnieją pojęcia atrybutu, obiektu danych i węzła logicznego. W wielu systemach SCADA (*Supervisory Control And Data Acquisition*) w konfiguracji sterowników komunikacyjnych przyjmuje się mapowanie rejestrów Modbus na zmienne. tj. uint8, uint16, czy typ zmiennoprzecinkowy zgodny z normą IEEE 754 [4]. Zwykle dokonuje się tego podając adres rejestru, offset oraz kolejność bajtów (*Endianness*). W przypadku IEC-61850 nazwa zmiennej składa się z nazwy urządzenia logicznego, węzła logicznego, obiektu danych i atrybutu, np. ProtectSi/GGIO7.Ind81.stVal, gdzie ProtectSi – urządzenie logiczne (*Logical Device*), GGIO7 – węzeł logiczny (*Logical Node*), Ind81 – obiekt danych (*Data Object*), stVal – atrybut (*Attribute*).

Mapowanie zmiennych

Mapowanie zmiennych odbywa się przez plik arkusza kalkulacyjnego w formacie xlsx. Edycji możemy dokonać za pomocą programu Microsoft Excel lub LibreOffice Calc, a następnie gotową konfigurację przesłać do urządzenia za pomocą interfejsu webowego. Plik zawiera tabelę w zakładce modbus-rtu, gdzie z lewej strony w komórki kolumny IEC_VAR_NAME wpisujemy nazwy zmiennych IEC-61850, np. ProtectSi/GGIO7.Ind81.stVal. Natomiast w komórkę DRV_VAR_NAME nazwę zmiennej modbus, np. mb_var_1. W przypadku sterownika modbus ta komórka nie ma specjalnego znaczenia. Wiersze stanowią konfigurację powiązania zmiennych. Następnie konfigurujemy dane dla protokołu Modbus:

- SLAVE_ID – identyfikator stacji podrzędnej (*slave*),
- REG_TYPE – typ rejestru np. HR (*Holding Registers*),
- REG_OFFSET – przesunięcie (*offset*),
- REG_COUNT – liczba rejestrów,
- REG_ENDIAN – kolejność bajtów (LittleEndian lub *BigEndian*),
- REG_FORMAT – formatowanie (dec/hex/bin),
- REG_BIT – numer bitu (dotyczy typu BOOL).

Tabela 2

Fragment konfiguracji zmiennych

SLAVE_ID	REG_TYPE	REG_OFFSET	REG_COUNT	REG_ENDIAN	REG_FORMAT	REG_BIT
1	HR	0	1	LittleEndian	Dec	1
5	HR	1	1	BigEndian	hex	

Tabela 2 zawiera fragment konfiguracji dwóch zmiennych. Pierwsza zmienna ma wpisaną w kolumnie REG_BIT wartość 1. Oznacza to, że oczekujemy bitu na pozycji nr 1. Zmienna wewnętrzna będzie typu BOOL, a następnie zostanie przypisana do zmiennej IEC-61850.

W przypadku konfiguracji zmiennych IEC-61850 tabela zawiera kolumnę QUALITY_ACK, służy ona do włączania funkcji – kontroli poprawności komunikacji. Algorytm działania tej funkcji jest następujący: w przypadku odczytu, gdy sterownik działający na porcie szeregowym prawidłowo odczytał wartość, zostanie ustawiana wartość atrybutu q na GOOD, w przypadku błędu na INVALID. Również w przypadku niepowodzenia sterowania zostanie ustawiona wartość INVALID.

Interfejs REST API

W 2000 roku Roy Thomas Fielding zdefiniował w swojej rozprawie doktorskiej styl architektury REST (*Representational State Transfer*) [6]. Obecnie jest to powszechnie stosowana architektura dla usług i aplikacji webowych w Internecie jak i przemysłowym Internecie rzeczy (*Industrial Internet of Things*). Otwarty interfejs REST API umożliwia integrację systemu z zewnętrznymi systemami i aplikacjami w łatwy sposób.

REST API stanowi interfejs bazujący na protokole HTTP (*Hypertext Transfer Protocol*) lub wersji szyfrowanej HTTPS między klientem a serwerem. Klient stosując standardowe metody protokołu HTTP (GET, POST, PUT etc.) ma dostęp do obiektów (danych) udostępnianych przez serwer. Komunikacja między klientem a serwerem jest bezstanowa.

Bezstanowość oznacza, że klient wysyłając żądanie musi zawrzeć w nim wszelkie możliwe informacje niezbędne do przetworzenia żądania przez serwer [5]. Podstawowym elementem dokumentacji interfejsu REST API jest lista i opis punktów końcowych (*endpoint*). Punkt końcowy to adres URI (*Uniform Resource Identifier*), do którego klient wysyła żądanie, następnie serwer przetwarza żądanie i odsyła odpowiedź.

Stosując odpowiednie metody HTTP można przeprowadzać operacje na zasobach udostępnianych przez serwer:

- metoda GET – służy do pobierania danych,
- metoda POST – służy do wysyłania danych do serwera, np. tworzenia jakiegoś zasobu na serwerze,
- metoda PUT – służy głównie do nadpisywania istniejących danych,
- metoda DELETE – służy do usuwania danych.

Dane odsyłane przez serwer najczęściej są w formacie JSON (*JavaScript Object Notation*).

REST API w bramce Modbus RTU/IEC-61850

Konwerter EKM Smart posiada serwer, w którym został zaimplementowany interfejs REST API. W obecnej implementacji interfejs działa w trybie tylko do odczytu. Oznacza to, że za pomocą zapytań HTTP GET istnieje możliwość odczytu informacji o konfiguracji usługi oraz zmiennych, na jakich operuje usługa.

API zostało podzielone na dwie grupy:

- informacje o konfiguracji – punkty końcowe ***/api/v1/config/mappings***, oraz ***/api/v1/config/iec-variables***; zawiera konfigurację stworzoną na podstawie modelu SCL i pliku konfiguracyjnego w formacie xls;
- informacje procesowe o zmiennych – punkty końcowe ***/api/v1/variables/iec***, oraz ***/api/v1/variables/iec***; zawiera informacje aktualizowane podczas działania usługi.

W API zostały zdefiniowane i zaimplementowane następujące punkty końcowe:

- /api/v1/config/mappings,
- /api/v1/config/iec-variables,
- /api/v1/variables/iec,
- /api/v1/variables/driver.

Informacje o mapowaniach zamiennych

Zmienne IEC-61850 są powiązane ze zmiennymi z sterownika Modbus RTU. Wysyłając zapytanie HTTP GET do serwera REST API, serwer w odpowiedzi zwróci dokument JSON. Poniżej zamieszczono fragment dokumentu JSON.

```
[
{
"iec_var_id":
"23D03382310BC22477663869AEA2579527ECF33F-
C005421446667038759A3EBA",
"other_var_id":
"6FBA12BC3FEF99FE73ED3A130D6CE681C-
309C407434692E1F5F9263BD4922DC5"
},
```

Rys. 2. Komunikacja za pomocą interfejsu REST API

```
{
"iec_var_id":
"F22A328367129542DFC19C45596E3D38233417F-
315F53305219173B51CDC5226",
"other_var_id":
"F51018C1CB0DAC705E3FAF887491E0B034F5AF-
FDCF73AECE374C2C9537D67F0C"
}
]
```


Każdy element listy dokumentu JSON zawiera:
- iec_var_name – identyfikator zmiennej IEC-61850,
- other_var_id – identyfikator zmiennej sterownika działającego na porcie szeregowym, w tym przypadku Modbus RTU.

Informacje o konfiguracji IEC-61850

W celu pobrania informacji o zmiennych, jakie zostały dodane do konfiguracji, należy wysłać zapytanie HTTP GET do punktu końcowego ***/api/v1/config/iec-variables***. Poniżej fragment dokumentu JSON.


```
[
  {
    "id":"23D03382310BC22477663869A-
    EA2579527ECF33FC005421446667038759A-
    3EBA",
    "name":"ProtectSi/GGIO7.Ind81.
    stVal",
    "type":"BOOLEAN",
    "fc":"ST"
  },
  {
    "id":"F22A328367129542DFC19C45596E3D
    38233417F315F53305219173B51CDC5226",
    "name":"ProtectSi/GGIO7.Ind82.
    stVal",
    "type":"BOOLEAN",
    "fc":"ST"
  }
]
```


Elementy listy w dokumencie JSON posiadają takie pola, jak:
- id – identyfikator zmiennej,
- name – nazwa zmiennej,
- type – typ zmiennej,
- fc – typ IEC-68150 Functional Constraints.

Informacje o zmiennych Modbus RTU

Aby otrzymać informacje o zmiennych przypisanych do sterownika działającego na porcie szeregowym należy wysłać zapytanie HTTP GET do punktu końcowego ***/api/v1/variables/driver***. W tym konkretnie przypadku są to zmienne drivera Modbus RTU.


```
{
"1":{
"BDC56CAA37A9407A864FC57A67DE3610FA487E-
BE2777143B763F93AC1D68FDBD":
{
    "value":{
      "U16":0
    },
    "quality":"Invalid",
    "last_update_ts":"2023-02-08T03:38:00.
    561230604Z",
    "last_interval_ms":998,
    "last_control_ts":"None"
},
"C75AC8FFC0C775C5095083DD8E34FB45AF34C8368C0CD2E-
303F84C522003600F":
{
    "value":{
      "BOOL":false
    },
    "quality":"Invalid",
    "last_update_ts":"2023-02-08T03:38:00.
    561216437Z",
    "last_interval_ms":998,
    "last_control_ts":"None"
}
}
```


Na wyżej zamieszczonym listingu został widnieje fragment wiadomości odesłanej przez serwer. Jest to dokument w formacie JSON składający się z mapy klucz wartość, gdzie klucz to numer portu szeregowego, w tym przypadku to 1 (COM1). W wartości mapy zawarta jest kolejna mapa klucz – wartość, gdzie kluczem jest identyfikator zmiennej.

Pojedynczy element mapy zawiera:
- value – struktura zawierająca informacje o typie i wartość zmiennej,
- quality – informacja stanie (jakości) zmiennej, zgodna z normą IEC-61850. Może przyjmować wartości: Good, Invalid, Reserved, Questionable,
- last_update_ts – stempel czasu, kiedy była ostatnia aktualizacja zmiennej,
- last_interval_ms – interwał aktualizacji zmiennej. Wartość wyrażona w milisekundach,
- last_control_ts – stempel czasu, kiedy było ostatnie sterowanie zmienną.

Istnieje również możliwość zapytania o konkretną zmienną przypisaną do sterownika. Aby otrzymać informacje na temat konkretnej zmiennej należy wysłać zapytanie HTTP GET z numerem portu szeregowego oraz ID zmiennej np. ***api/v1/variables/driver/:1/:77777cf650c3a20cd83ffe4bc1ed4dd***.

Informacje o zmiennych IEC-61850

Serwer REST API posiada zdefiniowany punkt końcowy ***/api/v1/variables/iec***, który służy to pozyskania informacji o zmiennych działających w domenie serwera MMS.

Poniższy fragment prezentuje odpowiedzi odesłanej przez serwer.


```
[
{
    "data_type":"BOOL",
    "id":"2B43D7772AE1F237584BDA-
    5893C86AA2F379B0616CAE04D6F-
    C7517E022C69D90",
    "name":"ProtectSi/GGIO7.Ind83.
    stVal",
    "value":"false"
},
{
    "data_type":"BOOL",
    "id":"F594B2D0E83DE1B667480A-
    9FAE21EE12BFF36520F66D11DCF0FE-
    0172BE82B4ED",
    "name":"ProtectSi/GGIO7.Ind89.
    stVal",
    "value":"false"
}
]
```


Dokument JSON zawiera listę zmiennych skonfigurowanych przez użytkownika i działających w usłudze bramki Modbus RTU/IEC-61850. Pojedynczy element zawiera:

- data_type – typ danych,
- id – unikalny identyfikator zmiennej w systemie,
- name – nazwa zmiennej w postaci nazwy IEC-61850,
- value – wartość zmiennej.

Podsumowanie

Konwerter protokołów EKM Smart jest zaawansowanym urządzeniem, które pozwala na integrację różnych protokołów używanych w przemyśle oraz zapewnia elastyczność, skalowalność i interoperacyjność w sieciach przemysłowych. Funkcje, takie jak REST API, konwersja protokołów Modbus RTU na IEC-61850 czynią z EKM Smart doskonałe narzędzie do zdalnego zarządzania i monitorowania urządzeń przemysłowych.

PIŚMIENNICTWO

[1] Blandy J., Orendorff J., *Programowanie w języku Rust. Wydajność i bezpieczeństwo,* Helion, 2020.

[2] Jackson J., *Microsoft: Rust Is the Industry's 'Best Chance' at Safe Systems Programming,* Artykuł dostępny pod adresem: https://thenewstack.io/microsoft-rust-is-the-industrys-best-chance-at-safe-systems-programming/ [dostęp: 03.04.2023].

[3] Lizer M., Szweicer W., *Norma IEC 61850 – nowy standard komunikacyjny systemu sterowania i nadzoru stacji elektroenergetycznych,* Artykuł dostępny pod adresem: https://www.academia.edu/63633136/Norma_IEC_61850_nowy_standard_komunikacyjny_systemu_sterowania_i_nadzoru_stacji_elektroenergetycznych [dostęp: 03.04.2023]

[4] Wikipedia, IEEE Standard for Floating-Point Arithmetic (IEEE 754), Wpis dostępny pod adresem: https://en.wikipedia.org/wiki/IEEE_754 [dostęp: 03.04.2023].

[5] Wikipedia Co to jest API REST?, Wpis dostępny pod adresem: https://www.ovhcloud.com/pl/learn/what-is-rest-api/ [dostęp: 3.04.2023]

[6] Lokesh Gupta What is REST, Wpis dostępny pod adresem: https://restfulapi.net/ [dostęp: 03.04.2023].

[7] Strona www: https://www.spie-energotest.pl/index/projekty-unijne/konwerter-ekm-smart.html

Projekt został dofinansowany z Funduszy Europejskich

Realizacja nowego urządzenia Konwerter protokołów EKM Smart została objęta umową o dofinansowanie projektu pt. „Budowa konwertera EKM_Smart dla magistral komunikacyjnych typu Ethernet z wykorzystaniem wielu protokołów" w ramach Regionalnego Programu Operacyjnego Województwa Śląskiego na lata 2014-2020 współfinansowanego ze środków Europejskiego Funduszu Rozwoju Regionalnego.

Wartość całkowita projektu: 571 147,36 PLN

Dofinansowanie z Funduszy Europejskich: 323 204,14 PLN