

## Integracja LabVIEW z rozwiązaniami IoT

**Streszczenie.** W artykule omówiono wybrane narzędzia programistyczne, które mogą być przydatne przy integracji oprogramowania tworzonego w środowisku LabVIEW z IoT (Internet of Things). Omówiono tworzenie aplikacji webowych oraz konteneryzację z wykorzystaniem Dockera.

**Abstract.** The article discusses selected programming tools that can be useful in integrating software created in the LabVIEW environment with IoT (Internet of Things). The development of web applications and containerization using Docker were discussed. (*Integration of LabVIEW with IoT devices*)

**Słowa kluczowe:** LabVIEW, Web Services, REST API, Docker  
**Keywords:** LabVIEW, Web Services, REST API, Docker

### Wstęp

Środowisko programowania LabVIEW [1] jest powszechnie wykorzystywane do tworzenia aplikacji testowych – w oparciu o nie działa większość urządzeń testujących działających na liniach produkcyjnych w pełni zautomatyzowanych lub wymagających ingerencji operatora. LabVIEW jest też szeroko rozpowszechnione w środowisku naukowym, jako narzędzie wspomagające eksperyment badawczy [2, 3, 4]. W obu tych przypadkach dostarcza dużej ilości danych, które są przetwarzane częściowo on-line, a w wielu przypadkach off-line. Jednym z kluczowych problemów jest duża ilość gromadzonych danych, ich archiwizacja, udostępnianie, przetwarzanie i wizualizacja (ang. Big Data, Big Data Analytics). W eksperymentach naukowych-badawczych oprogramowanie tworzone w LabVIEW do obsługi eksperymentu i gromadzenia danych działa zazwyczaj lokalnie na komputerze PC, a dane zapisywane są do plików w formacie tekstowym, np. csv (ang. comma-separated values) czy xlsx (MS Excel), rzadziej binarnym, np. TDMS (ang. Technical Data Management Streaming). Przetwarzanie tych danych odbywa się również lokalnie, a dostęp do tych danych, jak i do oprogramowania, jest zdalnie realizowany na ogół z wykorzystaniem usługi VPN (ang. Virtual Private Network) i pulpitu zdalnego.

W artykule omówiono wybrane narzędzia programistyczne, które umożliwiają przede wszystkim zdalny dostęp do danych i ich wizualizację, a także zdalny nadzór nad oprogramowaniem. To drugie działanie realizowane jest zwykle w ograniczonym zakresie, tj. nie wszystkie funkcje dostępne lokalnie są udostępniane w aplikacji zdalnej.

### Nakreślenie problemu

LabVIEW jest środowiskiem programistycznym, które umożliwia integrację tworzonego w nim oprogramowania z programami do zarządzania testami. Do tego celu wykorzystywane są na ogół komercyjne rozwiązania tego samego producenta co LabVIEW, co zapewnia bezproblemową integrację. Komercyjne oprogramowanie zapewnia całościową obsługę testów i eksperymentów badawczych umożliwiając uruchamianie poszczególnych – zdefiniowanych w oprogramowaniu – procedur testowych, rejestrację i archiwizację danych pomiarowych, ich przetwarzanie, generowanie raportów, zdalne udostępnianie oraz przetwarzanie danych, w tym w chmurze. Dla celów eksperymentów badawczych w wielu przypadkach tworzone jest oprogramowanie specyficzne dla określonego eksperymentu. Dane z eksperymentu zapisywane są w wybranym formacie do pliku i następnie

przetwarzane off-line z wykorzystaniem dostępnego oprogramowania (Excel, Matlab, itd.). W samym oprogramowaniu do rejestracji danych, implementowana jest często wizualizacja on-line celem uzyskania podglądu stanu działania aplikacji i weryfikacji poprawności rejestrowanych danych. Do tworzenia oprogramowania dla eksperymentu badawczego wykorzystuje się różnego rodzaju narzędzia komercyjne, najczęściej jest to środowisko LabVIEW, często Matlab, a rzadziej TestStand. Wymagają one uzyskania licencji badawczej lub badawczo-dydaktycznej. Powszechnie jest także wykorzystanie darmowego środowiska programowania Python. Właściwości scharakteryzowanego oprogramowania zestawiono w zwartej formie na rysunku 1.



Rys.1. Oprogramowanie systemu do eksperymentu badawczego

W celu zdalnego udostępnienia danych lub umożliwienia korzystania z interfejsu użytkownika (UI) można wykorzystać inne komercyjne oprogramowania, jak np. G Web Development Software (podobnie jak LabVIEW, firmy NI, udostępniane uczelniom w drodze licencji) lub w pełni komercyjny SystemLink (firmy NI). Alternatywą jest wykorzystanie darmowych narzędzi programistycznych dedykowanych do tworzenia serwisów WWW, które mogą komunikować się z aplikacją stworzoną w LabVIEW, wykorzystując usługi sieciowe - Web Services lub udostępniając dane zapisane w bazach danych przez tzw. kokpity menadżerskie lub pulpity nawigacyjne, w języku angielskim zwane *dashboards*.

W dalszej części artykułu omówiono pokrótce wybrane narzędzie programistyczne dedykowane do integracji LabVIEW z rozwiązaniami IoT.

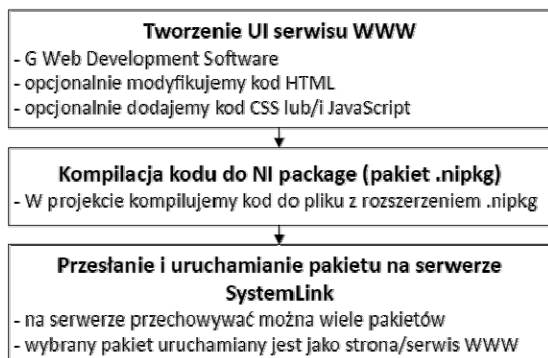
### Web Publishing Tool

Jednym z pierwszych narzędzi dostępnych w LabVIEW, które umożliwia publikowanie UI przez stronę WWW było Web Publishing Tool. Jest to obecnie narzędzie archaiczne, choć ciągle dostępne. Problemem jest brak kompatybilności z aktualnymi wersjami popularnych przeglądarek internetowych wynikający z braku obsługi wtyczki przeglądarki, wymaganej do poprawnego wyświetlania UI. Ponadto od wersji LabVIEW 2022 Q2 narzędzie Web Publishing Tool nie udostępnia już opcji zdalnej kontroli aplikacji przez stronę WWW – wyświetlany jest jedynie statyczny, z możliwością okresowego odświeżania, obraz UI.

### G Web Development Software i SystemLink

Obecnie, do tworzenia pulpitów nawigacyjnych (dashboards), uruchamianych jako serwis WWW, preferowane jest narzędzie G Web Development Software, w którym rdzeń serwisu tworzony jest identycznie jak to ma miejsce w LabVIEW, czyli graficznie. Jest to szeroko wykorzystywane narzędzie, praktycznie zawsze, jeśli tworzymy pulpity nawigacyjne uruchamiane z poziomu oprogramowania SystemLink. Równoległe do kodu graficznego, tworzony jest kod HTML, w którym osadzone są elementy interfejsu użytkownika. Kod HTML uzupełnić można o dodatkowe elementy, natywnie niedostępne w G Web Development Software, np. skrypty CSS czy JavaScript.

SystemLink to komercyjne rozwiązanie firmy NI, wykorzystywane do zarządzania zasobami (zarówno danymi jak i sprzętem), orkiestracji laboratoriów oraz narzędzi do analizy danych i zarządzania danymi. Pojęcie orkiestracji, zaczerpnięte jest z branży IT i rozumiane jest tutaj jako proces polegający na automatyzacji i zarządzaniu różnymi elementami infrastruktury informatycznej.



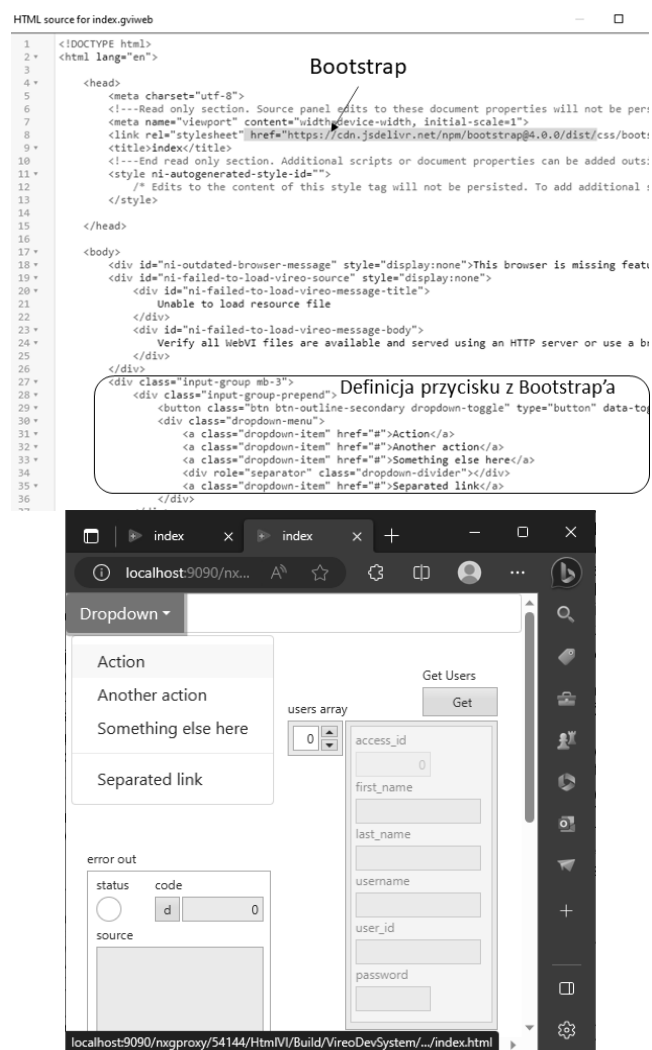
Rys.2. Proces tworzenia pulpitu nawigacyjnego

Przygotowane w G Web pulpity nawigacyjne kompiluje się do pakietów instalacyjnych NI Package Manager – pliki instalacyjne z rozszerzeniem .nipkg – i wysyła na serwer SystemLink. Tam są uruchamiane w chmurze, jako serwisy WWW. Na rysunku 2 przedstawiono krok po kroku schemat tworzenia i uruchamiania serwisu WWW. Na rysunku 3 pokazano kod HTML aplikacji oraz uruchomioną jako serwis WWW (pulpit nawigacyjny). Panel czołowy oraz diagram blokowy tworzy się jak w LabVIEW.

### Web Services

Usługi sieciowe – Web Services – pozwalają na komunikowanie się pomiędzy aplikacją kliencką i serwerową. Dla przykładu, aplikacja kliencka to serwis WWW uruchamiany lokalnie w przeglądarce internetowej a

aplikacja serwerowa działa w chmurze. Najczęściej komunikacja realizowana jest z wykorzystaniem protokołu HTTP, przy czym istnieje kilka mechanizmów komunikacji [5], wśród których najczęściej wykorzystywany jest REST (ang. Representative State Transfer). Formalnie REST rozumiany jest jako styl architektury oprogramowania ze zdefiniowanym zbiorem reguł, określających sposób zdefiniowania zasobów. Dzięki przyjętemu zbiorowi reguł, implementacja REST jest możliwa w wielu językach programowania niezależnie od platformy. Należy zapewnić jedynie odpowiednie API (ang. Application Programming Interface), czyli zestaw reguł definiujących komunikację pomiędzy systemami – np. klientem a serwerem. Wprowadzimy tu jeszcze dwa pojęcia. Pierwszym jest RESTful, które oznacza, że API (dla REST) spełnia przyjęte reguły. Nie będziemy wchodzić tu w szczegóły a odsyłamy do literatury [6]. Drugim pojęciem jest Endpoint, czyli adres zasobów, który jednoznacznie wskazuje do jakiego zasobu się odwołujemy. Pojęcie to pojawi się jeszcze w dalszej części artykułu.

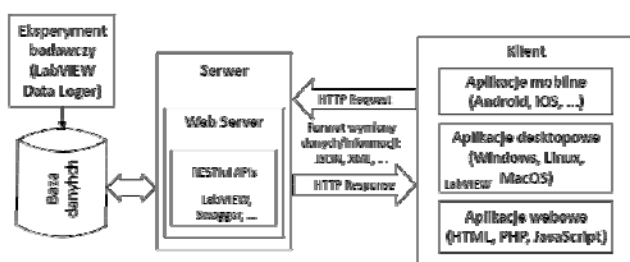


Rys.3. Przykład serwisu WWW stworzonego w G Web Development Software. U góry: fragment kodu HTML, u dołu: serwis uruchomiony w przeglądarce

W standardzie HTTP dla REST zdefiniowano metody, spośród których najczęściej wykorzystuje się cztery: GET, POST, PUT oraz DELETE. W uproszczeniu można je opisać następująco: GET – służy do pobrania określonego zasobu, POST – tworzy nowy zasób, PUT – aktualizuje określony zasób, DELETE – usuwa określony zasób.

Na rysunku 4 zobrazowano mechanizm wymiany danych pomiędzy aplikacjami klienckimi a serwerem, z wykorzystaniem RESTful API. Przed tworzeniem serwisu, należy skonfigurować NI Web Server oraz Application Web Server, korzystając z kategorii Web Server w ustawieniach środowiska LabVIEW (menu Tools >> Options...). Domyślny port HTTP dla NI Web Server to 9090, zaś dla Application Web Server – 8001. Jest to istotna informacja z uwagi na sposób uruchamiania serwisu bezpośrednio z okna projektu jak i wersji instalacyjnej (wspomnianego wcześniej pakietu .nlpkg). Zostanie to jeszcze poruszone w dalszej części artykułu.

W przypadku Application Web Server, konfiguracja odbywa się przez stronę WWW wyświetloną po wybraniu opcji Configure Application Web Server. Należy zwrócić uwagę na to, aby wybrać odpowiednią wersję bitową serwera, identyczną jak wersja LabVIEW (32 lub 64 bitowa).



Rys.4. Mechanizm wymiany danych z wykorzystaniem Web Services

W LabVIEW, Web Services [7] dodajemy w oknie Project Explorera, jak to pokazano na rysunku 5. Odpowiedni folder wirtualny pojawi się w oknie projektu. Jeśli chcemy udostępnić zasoby (np. pliki), można do serwisu dodać tzw. Public Content i połączyć go z folderem na dysku komputera, gdzie umieszczone są udostępniane zasoby.

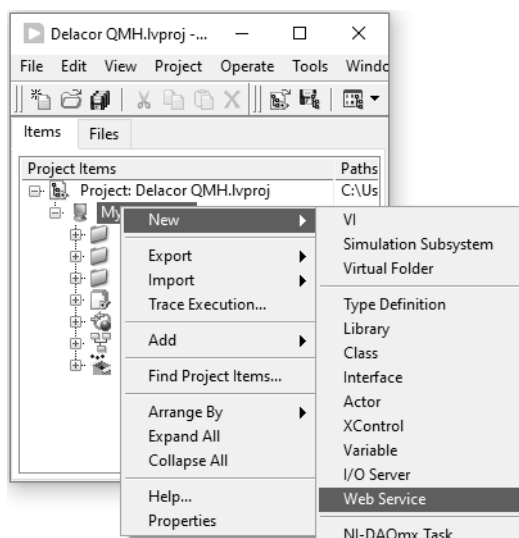
Uruchamianie serwisu, może odbywać się na dwa sposoby (patrz rysunek 6), w trybie normalnym, poprzez wybór polecenia Start a następnie Application Web Server>>Publish, lub w trybie Debug, wówczas bezpośrednio wybieramy Application Web Server >>Start(Debug Server). Diagram blokowy przykładowej metody zamieszczono na rysunku 7. Metody wywoływane są jako adresy WWW, czy to w przeglądarce czy w innej aplikacji. Adres danej metody sprawdzić można wywołując z menu kontekstowego polecenie Show Method URL.

Należy również mieć na uwadze również odpowiedni format danych przesyłanych pomiędzy klientem a serwerem. Najczęściej korzysta się z formatu JSON (ang. JavaScript Object Notation) lub XML (ang. eXtensible Markup Language). W LabVIEW dostępne są funkcje wykorzystywane do konwersji danych do oraz z JSON oraz XML.

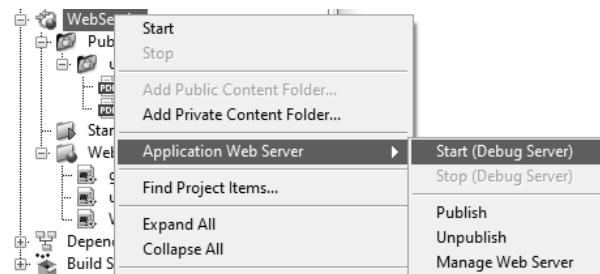
### Konteneryzacja

LabVIEW jako środowisko programowania stworzone przede wszystkim do tworzenia aplikacji automatyzujących pomiary czy testy, ma swoje ograniczenia – głównie jeśli chodzi o narzędzia do tworzenia nowoczesnych pulpitów nawigacyjnych. Wspomniane wcześniej G Web Development Software jest narzędziem dosyć topornym, z uwagi na bardzo ograniczony zasób kontrolki i wskaźników, z których buduje się interfejs pulpitu nawigacyjnego. Umożliwia tworzenie jedynie prostych wizualizacji. Jednak dodanie Web Services do projektu LabVIEW umożliwia wykorzystanie innych, dedykowanych

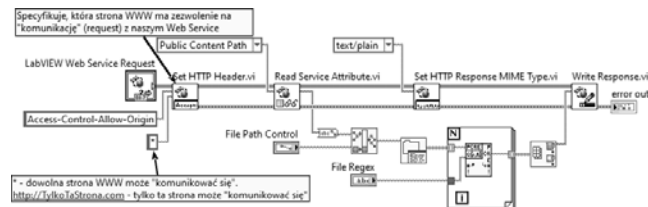
tworzeniu serwisów WWW narzędzi, jak choćby Vue.js. Innym podejściem jest wykorzystanie narzędzia Grafana [8], służącego do tworzenia pulpitów nawigacyjnych. Grafana posiada zestaw wtyczek (ang. plugins), umożliwiających pobieranie danych z różnych źródeł, np. baz danych SQL Server, PostgreSQL czy SQLite, do których LabVIEW może zapisywać. W rozbudowanych aplikacjach często korzysta się z różnorodnych narzędzi do generowania i przetwarzania danych, np.: Kafka, Flink, Elasticsearch, Prometheus, FastAPI. Z uwagi na brak miejsca na szerszy ich opis, ograniczono się jedynie do podania nazw. Ponieważ są to rozwiązania typu open-source, dostępna jest szczegółowa dokumentacja w Internecie.



Rys.5. Dodawanie Web Service do projektu



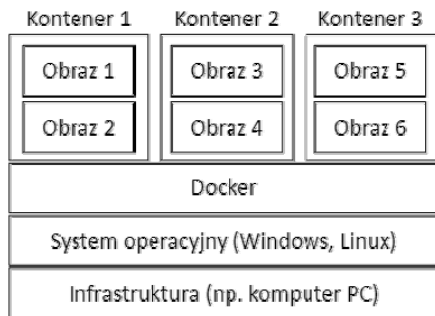
Rys.6. Uruchamianie Web Service



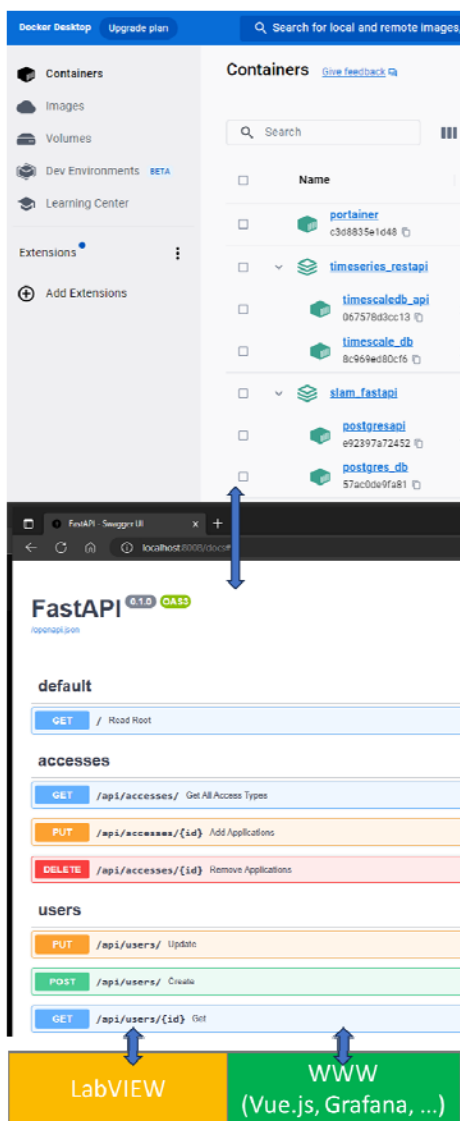
Rys.7. Przykładowy kod (zapis) graficzny metody GET

Narzędzi tych na ogół nie instaluje się bezpośrednio w systemie operacyjnym komputera czy serwera a korzysta się z maszyn wirtualnych oraz działających w nich kontenerów. Kontenery działają też bezpośrednio w systemie operacyjnym komputera i takie rozwiązanie stosowane jest zazwyczaj na etapie rozwoju oprogramowania.

Jednym z najpopularniejszych narzędzi do konteneryzacji jest Docker (np. Docker Desktop for Windows) [9]. W odróżnieniu od typowej maszyny wirtualnej, realizuje on wirtualizację na poziomie systemu operacyjnego. Docker to oprogramowanie służące do tworzenia, uruchamiania i kontroli kontenerów.



Rys.8. Konteneryzacja aplikacji (Obraz 1, 2,...) z wykorzystaniem Dockera



Rys.8. Konteneryzacja aplikacji (Obraz 1, 2,...) z wykorzystaniem Dockera

Kontenery, w uproszczeniu, zawierają obrazy np. aplikacji. Obrazy można tworzyć samemu (np. aplikację napisaną w języku Python), wywołując odpowiednie

polecenie Dockera, ale najczęściej korzysta się z zasobów obrazów umieszczonych na stronie <https://hub.docker.com>. Kontener może składać się z jednego obrazu (np. bazy danych do obsługi użytkowników) lub wielu (np. baza SQLite, do której zapisujemy dane generowane w LabVIEW oraz Grafana, która z danych zapisanych w bazie korzysta). Obrazowo przedstawiono to na rysunku 8.

Na rysunku 9 pokazano przykładową aplikację, na którą składa się wiele komponentów: oprogramowanie generujące dane, zrealizowane w LabVIEW, aplikacje webowe – generująca i odbierająca dane, napisana w Vue.js oraz Grafana do wizualizacji danych. Dalej, FastAPI (uruchomione jako kontener), w którym stworzono API (z wykorzystaniem REST) zapewniające komunikację wspomnianych wcześniej komponentów z kontenerami (poprzez wspomniane wcześniej endpointy), w których z kolei uruchomione są obrazy aplikacji bazodanowych.

### Podsumowanie

Dostępne technologie internetowe i narzędzia programistyczne umożliwiają integrację aplikacji tworzonych w LabVIEW z IoT. LabVIEW udostępnia rozbudowane struktury i zestawy bibliotek (ang. frameworks), umożliwiające tworzenie skalowalnych aplikacji. Implementacja Web Services w projekcie LabVIEW, pozwala z kolei na wymianę danych z aplikacjami sieciowymi z wykorzystaniem RESTful API. Daje to możliwości szerokiego wykorzystania narzędzi do archiwizacji i przetwarzania danych, generowania statystyk i pulpitów nawigacyjnych. Dodatkowo, większość tych narzędzi można konteneryzować, co ułatwia zarządzanie rozbudowanymi systemami informatycznymi, wspomagającymi pomiary i testy.

**Autorzy:** dr inż. Roman Wyżgolik, Politechnika Śląska, Katedra Pomiarów i Systemów Sterowania, ul. Akademicka 16, 44-100 Gliwice, E-mail: [roman.wyzgolik@polsl.pl](mailto:roman.wyzgolik@polsl.pl); dr hab. inż. Sebastian Budzani, prof. Politechniki Śląskiej, Politechnika Śląska, Katedra Pomiarów i Systemów Sterowania, ul. Akademicka 16, 44-100 Gliwice, E-mail: [sebastian.budzani@polsl.pl](mailto:sebastian.budzani@polsl.pl).

### LITERATURA

- [1] Jennings R., De la Cueva F., LabVIEW Graphical Programming, Fifth Edition, McGraw-Hill Education Ltd, (2019)
- [2] Nizioł M., Opracowanie oprogramowania do obsługi systemu pomiarowego zgodnego ze standardem IEEE 1451 w środowisku LabVIEW, *Przegląd Elektrotechniczny*, 98 (2022), nr 11, 181-184
- [3] Kapri R.K., Dubey P.K., Sharma P., Development of photon counting technique to operate from discrete photons to continuous regime using DAQ card in LabVIEW platform, *Optics & Laser Technology, Part A*, vol. 158 (2023)
- [4] Javed M.A., Naveed M.A., Hussain A., Hssain S., Integrated data acquisition, storage and retrieval for glass spherical tokamak (GLAST), *Fusion Engineering and Design*, vol. 152 (2020)
- [5] Kumar P. K., Agarwal R., Shivaprasad R., Sitaram D., Kalambur S., Performance Characterization of Communication Protocols in Microservice Applications, *International Conference on Smart Applications, Communications and Networking (SmartNets)*, Glasgow, United Kingdom, (2021), pp. 1-5
- [6] Richardson L. Amundsen, Ruby S., RESTful Web APIs. Services for a changing world, O'Reilly Media, (2013)
- [7] Senthilnathan K., Annapoorani I., Modified dual output single phase current source back end converter with resilient cyber infrastructure, *Electrical Power and Energy Systems*, 124 (2021)
- [8] Bers T., Biermann D., Erkorkmaz K., M'Saoubi R., Digital twins for cutting process, *CIRP Annals - Manufacturing Technology*, 00 (2023), pp 1-21
- [9] Gonzalez I., Calderon A.J., Fogado F.J., IoT real time system for monitoring lithium-ion battery long-term operation in microgrids, *Journal of Energy Storage*, 51 (2022)