

# Systemy SCADA w środowisku Android

Krzysztof Oprzędkiewicz, Dominik Barwacz

AGH Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki,  
Informatyki i Inżynierii Biomedycznej, Katedra Automatyki i Inżynierii Biomedycznej

**Streszczenie:** W pracy zaprezentowano implementację systemu SCADA na urządzeniu mobilnym, pracującym pod nadzorem systemu operacyjnego Android. Aplikacja została przygotowana w Javie, z wykorzystaniem SDK Androida dla Eclipse, zgodnie z regułami programowania obiektowego. Połączenie z nadzorowanym sterownikiem PLC odbywa się za pomocą protokołu Modbus TCP i bezprzewodowego Ethernetu (Wi-Fi lub komórkowe). Wykonane testy wykazały poprawność działania proponowanego systemu.

**Słowa kluczowe:** systemy SCADA, urządzenia mobilne, Android

DOI: 10.14313/PAR\_214/95

## 1. Wprowadzenie

Urządzenia mobilne, pracujące pod nadzorem różnych systemów operacyjnych (np. Androida), są obecnie szeroko rozpowszechnione ze względu na rozbudowaną funkcjonalność. Z tego względu pojawia się coraz więcej rozwiązań programistycznych przeznaczonych dla tego środowiska, przy czym jest to nie tylko oprogramowanie służące rozrywce, lecz także zawiera profesjonalne aplikacje biznesowe i użytkowe. Naturalną konsekwencją tego faktu powinna być możliwość realizacji na platformach mobilnych również aplikacji SCADA, które stanowiłyby uzupełnienie istniejących systemów i pozwalałyby na zdalny nadzór systemu sterowania również spoza terenu sterowni lub zakładu.

Odwzorowanie ekranów aplikacji SCADA na urządzenie mobilne może być zrealizowane za pomocą prostych narzędzi informatycznych [3]; możliwa jest także budowa aplikacji pomiarowo-sterujących na urządzenia mobilne [5]. Należy jednak pamiętać, że w wielu przypadkach czynnikiem krytycznym, ograniczającym praktyczne zastosowania rozważanego podejścia jest konieczność zapewnienia bezpieczeństwa pracy systemu sterowania, co nie jest zagadnieniem prostym podczas stosowania technologii bezprzewodowych. Zagadnienia z tego zakresu są omawiane w wielu pracach [4].

Aktualnie oferta rynkowa aplikacji SCADA dedykowanych urządzeniom mobilnym nie jest jeszcze zbyt szeroka. Jako przykłady takich systemów można przytoczyć: ProficySCADA w wersji dla urządzeń mobilnych [8], TeslaSCADA [7], S7 PLC HMI Lite [9] oraz AirSCADA Lite [10].

ProficySCADA to zdalny klient, łączący się z klasycznym serwerem wizualizacyjnym CIMPLICITY lub iFIX. Jest to zdalny klient właściwej aplikacji SCADA i z tego względu ma dość duże możliwości, co jednak podnosi koszt całego systemu i sama aplikacja staje się raczej rozszerzeniem i dodatkiem do istniejących systemów, niż niezależną wizualizacją.

TeslaSCADA umożliwia tworzenie aplikacji wizualizacyjnych bezpośrednio na telefonie/tablecie. Do dyspozycji użytkownika są 42 obiekty graficzne. Każdy z nich jest konfigurowalny w zakresie położenia na ekranie, tekstu, kolorów itp. Interfejs użytkownika jest przyjazny, oferuje łatwe przenoszenie oraz zmienianie wielkości obiektów. Niektóre z nich pozwalają dodatkowo na definiowanie własnych skryptów, uruchamianych po naciśnięciu po wystąpieniu skonfigurowanego zdarzenia. Niestety jedynym dostępnym standardem komunikacyjnym jest OPC.

S7 PLC HMI Lite jest dedykowany wyłącznie do sterowników SIEMENS serii 1200 300 i 400, połączenie ze sterownikami przez WLAN oferuje dość duże możliwości, natomiast nie jest autoryzowany przez SIEMENSA i użytkownik może go stosować na własną odpowiedzialność.

AirSCADA Lite i Pro wspiera protokół Modbus TCP, natomiast jego funkcjonalność ogranicza się do prezentowania trendów i alarmów.

Jak widać, tylko jedna z przedstawionych aplikacji umożliwia wsparcie protokołu Modbus TCP. Stosując go można stworzyć aplikację będącą samowystarczalną aplikacją SCADA, a nie zdalnym klientem, jak ma to miejsce w powyższych przykładach.

Celem niniejszej pracy jest prezentacja możliwości realizacji aplikacji SCADA na platformie urządzenia mobilnego, pracującego pod nadzorem systemu Android (np. telefonu komórkowego) i komunikującej się ze sterownikiem za pomocą protokołu Modbus TCP. W pracy zostaną omówione następujące zagadnienia:

- podstawowe założenia do budowy systemu,
- zasada działania aplikacji,
- interfejs użytkownika,
- testy działania.

## 2. Podstawowe założenia do budowy systemu

Podczas opracowywania aplikacji przyjęto następujące fundamentalne założenia:

- aplikacja jest szkieletem otwartym na przyszły rozwój, w procesie projektowania największy nacisk kładziony jest na stworzenie jak najbardziej uniwersalnych rozwiązań,
- przeniesienie jak największego udziału tworzenia nowych ekranów z kodu na szablony *.xml*.

Komponenty sprzętowe aplikacji są następujące:

- telefon z systemem Android (w rozważanym przypadku wybrano model LG E960),
- sterownik PLC obsługujący protokół Modbus TCP w trybie serwer (w rozważanym wypadku testy wykonano na sterowniku PACSystems RXi).

Dodatkowo, w celu budowy i uruchomienia własnej aplikacji, niezbędne są następujące dodatkowe komponenty:

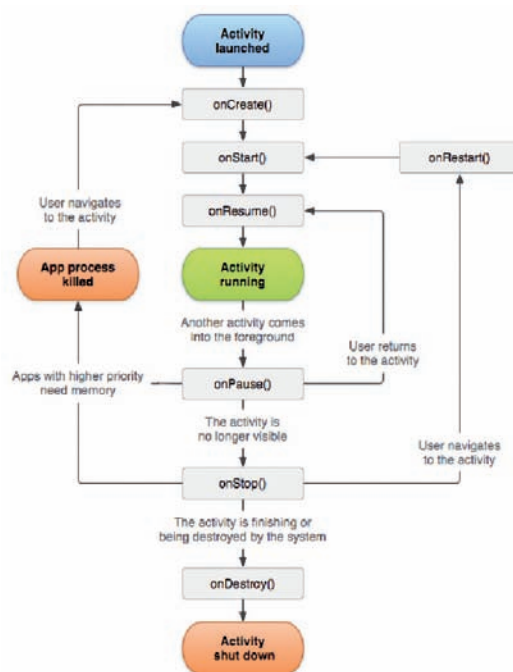
- komputer z zainstalowanym oprogramowaniem Eclipse JUNO działający pod kontrolą Linuksa, Windowsa lub MacOS,
- kabel komunikacyjny micro B USB,
- sterowniki deweloperskie do programowanego urządzenia.

Podczas realizacji części programowej prezentowanego systemu SCADA przyjęto następujące założenia:

- oprogramowanie ma być utworzone w języku Java, z wykorzystaniem SDK Androida dla Eclipse,
- schemat oraz logika programu powinna być zgodna z założeniami programowania obiektowego OOP (ang. *Object-Oriented Programming*),
- system powinien poprawnie działać na najnowszym dostępnym systemie Android (w chwili realizacji pracy był to KitKat 4.4.2, obecnie jest to KitKat 4.4.4),
- system powinien umożliwiać zdalną komunikację z dowolnym adresem IP z wykorzystaniem dowolnego portu,
- czytelna i estetyczna wizualizacja stanu serwera,
- organizacja i wygląd interfejsu aplikacji powinien być zgodny z zaleceniami producenta systemu operacyjnego,
- kod aplikacji powinien być „przyjazny” dla użytkowników niezapoznanych ze szczegółami działania programu.

## 3. Zasada działania aplikacji

Prezentowana aplikacja jest interfejsem łączącym użytkownika z biblioteką do komunikacji za pomocą proto-



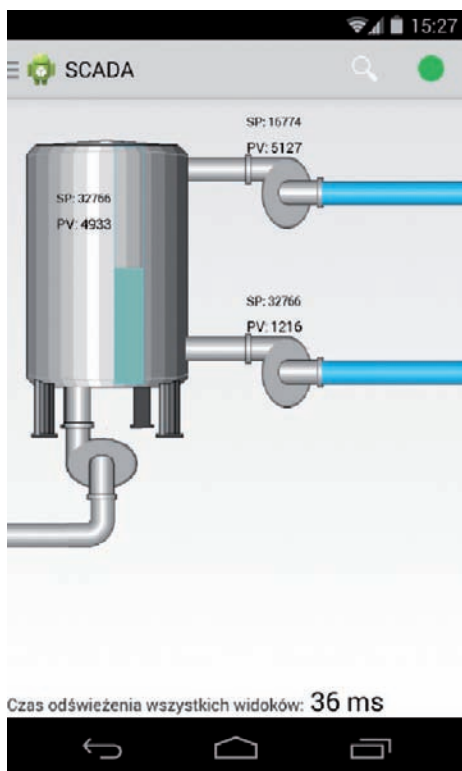
Rys. 1. Cykl życia aktywności związanej z działaniem aplikacji  
Fig. 1. Life cycle of activity associated to application

kołu Modbus TCP. Jej działanie jest realizowane przez zarządzanie ekranami użytkownika, zapamiętywanie parametrów połączenia i przekazywanie ich do biblioteki umożliwiającej komunikację po Modbus TCP w języku Java – *jamod*. W zależności od charakteru i funkcjonalności aktualnie otwartego ekranu oraz widocznym obiektom graficznym, aplikacja generuje w pętli zapytania za pomocą dostępnego łącza internetowego – Wi-Fi lub połączenia komórkowego. W przypadku łącza Wi-Fi należy pamiętać o zapewnieniu bezpieczeństwa wymiany danych (zastosowanie odpowiedniej metody szyfrowania).

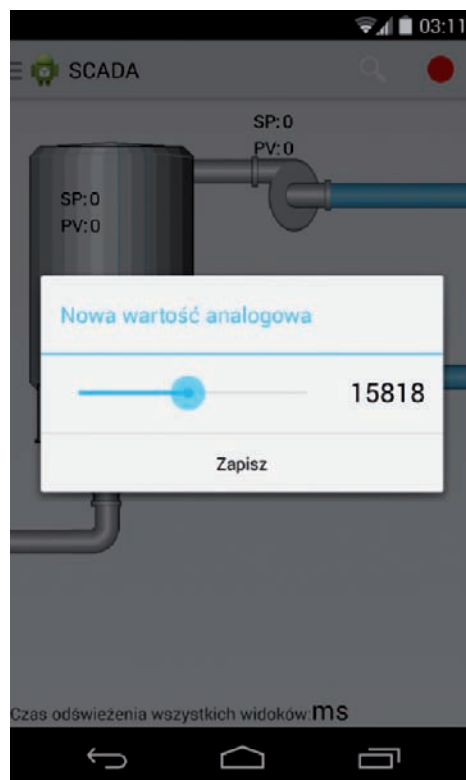
W skład kodu źródłowego aplikacji wchodzi następujące elementy:

- manifest definiujący globalne właściwości aplikacji,
- główna aktywność *MainActivity.java*, aktywna zawsze wtedy, gdy działa aplikacja,
- fragmenty będące „podaktywnościami” – dzieci głównej aktywności, które mogą wyświetlać własne ekrany równoległe z głównym tokiem działania aplikacji,
- zbiór dostępnych obiektów graficznych wraz z implementacją ich zachowania,
- zbiór instrukcji do wykonywania w równoległych wątkach, gwarantujących ciągłą komunikację z serwerem,
- plik *.xml* definiujący wygląd okna aplikacji.

System Android reaguje na polecenie użytkownika uruchamiające aplikację tworząc nowy wątek i w jego zakresie powołując do życia aktywność *MainActivity*. Do jej głównych zadań należy:



**Rys. 2.** Okno główne przykładowej aplikacji  
**Fig. 2.** The main window of exemplary application



**Rys. 3.** Pole do wprowadzania wartości analogowej  
**Fig. 3.** Input field for analog value

- generowanie interfejsu,
- wyświetlanie bocznych menu,
- wywołanie fragmentów kodu odpowiadających na kliknięcia użytkownika,
- odświeżanie obiektów graficznych,
- zawieszanie komunikacji po przejściu aplikacji w stan uśpienia.

Jak każda aktywność, *MainActivity* podlega pod system zarządzania zasobami i zgodnie z aktualnymi potrzebami lub w odpowiedzi na działanie użytkownika, może znaleźć się w jednym ze stanów przewidzianych na diagramie (rys. 1).

W przypadku przejścia aktywności z trybu działania do trybu uśpienia, przeładowana metoda *onPause()* wyłącza uruchomione przez aplikację wątki. Zmniejsza to zużycia energii oraz ogranicza nagrzewanie się urządzenia. Z kolei metoda *onResume()* otwiera ponownie wątek odpytywania serwera, chyba że połączenie nie zostało skonfigurowane. W takiej sytuacji wyświetlany jest monit przypominający użytkownikowi o konieczności ustawienia parametrów komunikacji.

Aktywność *MainActivity* musi również definiować sposób łapania wyjątków nieobsłużonych przez równoległe wątki. Jeśli wystąpi taka krytyczna sytuacja, aplikacja spróbuje zamknąć połączenie i otworzyć je ponownie.

Jeśli z jakiegoś powodu takie działanie również wygeneruje wyjątek, aplikacja zostanie zamknięta w trybie awaryjnym.

W trakcie tworzenia projektu zostały wyodrębnione najbardziej potrzebne typy obiektów. Są to:

- pasek postępu wskaźnika analogowego,
- graficzny wskaźnik dyskretny,
- agregat wartości analogowej z jej opisem.

Dla każdego z powyższych typów została utworzona klasa, której zadaniem jest przechowywanie informacji powiązanych z tym obiektem oraz wyświetlanie ich reprezentacji na ekranie.

Sposób wyświetlania każdego z obiektów podstawie zależy od parametrów zawartych w pliku *main\_scada.xml*. Dzięki temu użytkownik nie musi edytować samego kodu aby utworzyć lub edytować zawartość ekranów. Podstawowy zestaw parametrów, konieczny w przypadku każdego z obiektów to:

- *layout\_id* – unikalny identyfikator słowny,
- *layout\_width* – szerokość obiektu,
- *layout\_height* – wysokość obiektu,
- *layout\_alignTop/Bottom/Left/Right* – id obiektu, do którego rozpatrywany obiekt ma być przyciągnięty,
- *layout\_marginTop/Bottom/Left/Right* – przesunięcie względem obiektu, do którego ten jest przyciągany,
- *address* – adres do odpytywania,

- `editable` – definiuje, czy obiekt powinien mieć możliwość edycji wartości zmiennej powiązanej z adresem.

Położenie obiektów na ekranie jest określane względnie wobec siebie. Wybór takiego układu może wydać się niewygodny, należy jednak pamiętać, że jest to jedyny sposób, aby stworzyć aplikację działającą poprawnie na różnych urządzeniach. Takie rozwiązanie może być wygodniejsze, jeśli zdefiniowane zostaną odpowiednie zależności. Przy tworzeniu instalacji ciągłych, np. rurociągów, wystarczy zmienić położenie bezwzględne pierwszego obiektu, aby każdy kolejny (powiązany z poprzednim) automatycznie przesunął się w odpowiednie miejsce.

Wszystkie obiekty mają również swoje własne unikalne cechy, zależne od pełnionej w aplikacji funkcji (kolor ramki i wypełnienia, zakresy wartości, widoczność w zależności od warunku logicznego itp.)

## 4. Interfejs użytkownika

Forma graficzna okna aplikacji jest stosunkowo prosta. Po uruchomieniu użytkownikowi pokazuje się główne okno aplikacji. Jeśli użytkownik jeszcze tego nie zrobił, po uruchomieniu aplikacji pojawia się monit przypominający o konieczności skonfigurowania i nawiązania połączenia. U samej góry widoczna jest nazwa aplikacji, kolejna ikona w prawo to szybkie uruchomienie wyszukiwania w internecie hasła „SCADA”, dalej czerwone kółko sygnalizujące status połączenia. Po nawiązaniu połączenia z serwerem zmienia ono swój kolor na zielony.

Zgodnie z zaleceniami producenta systemu Android, zamiast klasycznych, spotykanych w systemie Windows przycisków, menu otwiera się przez przesunięcie palcem od bocznej krawędzi ekranu ku środkowi. Dzięki takiemu podejściu interfejs jest wygodniejszy dla użytkownika i ułatwia obsługę programu na urządzeniach z mniejszym wyświetlaczem.

Adres oraz port serwera można skonfigurować w ekranie „Ustawienia połączenia”. Adres podawany jest jako IP lub nazwa hosta.

W przypadku systemów automatyki typowym sposobem zabezpieczenia przed niepożądanym dostępem jest odcięcie dostępu do urządzeń spoza wewnętrznej, zaufanej sieci. Z tego względu najczęściej stosowaną przez użytkownika opcją będzie wpisany jawnie adres z puli adresów wewnętrznej sieci i pod tym kątem optymalizowane jest działanie aplikacji. Wprowadzenie błędnego adresu jest wykrywane i sygnalizowane odpowiednim komunikatem.

Rozwinięcie prawego menu w dowolnym miejscu aplikacji spowoduje otwarcie menu akcji. Daje ono możliwość nawiązania lub przerwania komunikacji, a także uruchomienia wątku odpowiadającego za regularne odpytywanie serwera.

Jako przykład zostanie pokazane główne okno aplikacji, która ma za zadanie wizualizację działania zbiornika służącego do mieszania dwóch składników oraz systemu

współpracujących z nim pomp (rys. 2). Dwie z nich dostarczają składników do mieszania, trzecia przekazuje gotową mieszaninę dalej. Aplikacja daje możliwość:

- wizualizacji stanu pomp,
- ręcznej zmiany ich stanu,
- ustalania zadanego poziomu cieczy do przelania do zbiornika (SP),
- ustalenia maksymalnej objętości zbiornika,
- wizualizacji aktualnego napełnienia zbiornika (CV),
- podglądu aktualnego czasu uzyskania odpowiedzi na temat wszystkich zmiennych reprezentowanych przez widoczne obiekty graficzne.

Wartości zmiennych skojarzonych z obiektami mogą być również wprowadzane z poziomu aplikacji. Przykładowe pole do wprowadzania wartości zmiennej analogowej pokazane jest na rys. 3.

## 5. Testy

W celu sprawdzenia poprawności działania aplikacji, przeprowadzono testy mające udowodnić jej przydatność w rzeczywistych zastosowaniach. Do tego celu wykorzystano sterownik PACSystems RXi z zaimplementowanym prostym programem symulującym działanie zbiornika napełnianego dwiema cieczami.

Sterownik PAC RXi programuje się korzystając z zintegrowanego środowiska dla produktów GE Intelligent Platforms – Proficy Machine Edition. Program zapisywany do sterownika dzielił się na część związaną z konfiguracją sprzętu oraz program sterujący. Pierwsza ze składowych definiuje elementy systemu, druga logikę działania. Program został utworzony w języku LD z wykorzystaniem dodatkowych funkcjonalności oferowanych przez kontroler RXi.

Testy polegały na sprawdzeniu działania aplikacji w czasie normalnego użytkownika. Składały się one m. in. na wykonywaniu żądań zapisu do różnych typów zmiennych, przeprowadzanych równocześnie z komunikacją, mającą na celu zapewnienie aktualnych informacji o stanie sterownika.

Oprócz kontroli sterowania zaworami (zmiennie dyskretne), użytkownik ma również możliwość podglądu wartości zmiennych analogowych. Dodatkowo, możliwe jest ręczne ustawienie pożądanego poziomu napełnienia zbiorników. Wszystkie te możliwości zostały sprawdzone i działały poprawnie.

Komunikacja ze sterownikiem była testowana zarówno z użyciem Wi-Fi, jak i sieci komórkowej – działanie aplikacji było w każdym wypadku poprawne.

## 6. Uwagi końcowe

W pracy zaproponowano przykład realizacji systemu SCADA dedykowanego dla systemu Android, który może



być dobrym narzędziem pomocniczym do monitorowania i nadzoru pracy systemu sterowania z dowolnego miejsca, gdzie tylko jest zasięg sieci komórkowej. Testy wykazały poprawność działania zbudowanego przykładowego systemu.

Omówiony system ma możliwości dalszej rozbudowy w kierunku zwiększenia jego funkcjonalności przykładowo w kierunku możliwości stosowania funkcji skryptowych lub też prostej archiwizacji zebranych danych, które później mogłyby być prezentowane w postaci trendów historycznych.

## Bibliografia

1. Bailey D., Wright E., *Practical SCADA for industry*, Elsevier 2003.
2. Barwacz D., *Bezprzewodowy system SCADA dla środowiska Android*, praca dyplomowa inżynierska zrealizowana w Katedrze Automatyki i Inżynierii Biomedycznej na Wydziale Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej Akademii Górniczo-Hutniczej (promotor Krzysztof Oprzędkiewicz), Kraków 2013 r.
3. Chrzanowski D., Kołek K., Sotwin J., *Transfer ekranów synoptycznych pakietu SCADA na urządzenia przenośne*, „Pomiary Automatyka Robotyka”, nr 3/2013, 116–119.
4. Geer D., *Security of Critical Control Systems Spark Concern*, “Computer”, Vol. 39, No. 1, 2006, 20–23. DOI: 10.1109/MC.2006.32
5. Kołek K., *Application of Android OS as real-time control platform*, „Automatics” (Automatyka) półrocznik Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie, rocznik 17, nr 2, 2013, 197–206.
6. ASTOR Integracja systemów IT w przedsiębiorstwach produkcyjnych. ASTOR – Automatyka przemysłowa | Efektywność produkcji | Technologie IT | Szkolenia | Consulting, [astor.com.pl].
7. Wikipedia. USB. *Wikipedia*. [http://en.wikipedia.org/wiki/File:MicroB\_USB\_Plug.jpg].
8. [http://developer.android.com/reference/android/app/Activity.html] – Google. Activity | Android Developers. *Android Developers*.
9. [http://jamod.sourceforge.net/kb/tcp\_master\_howto.html] – Jamod. Jamod – TCP Master How To. *Jamod*.
10. [http://teslascada.com] – Android OPC UA, Modbus and Siemens SCADA
11. [https://itunes.apple.com/us/app/proficyscada/id525792142?mt=8] – ProficySCADA on the App Store on iTunes.
12. [https://play.google.com/store/apps/details?id=com.prunsys.s7plchmi&hl=pl] – S7 PLC HMI Lite.
13. [https://play.google.com/store/apps/details?id=com.ssd.droidmod&hl=pl] – AirSCADA Lite. ■

## SCADA for Android

**Abstract:** In the paper an implementation of SCADA at mobile device, working under Android is presented. The application was prepared with use of Java and SDK Android for Eclipse with respect to Object Oriented Programming rules. The connection to supervised PLC is realized with the use of Modbus TCP and wireless Ethernet. Results of tests show, that the proposed system runs properly.

**Keywords:** SCADA, mobile devices, Android

Artykuł recenzowany, nadesłany 15.09.2014 r., przyjęty do druku 02.11.2014 r.

---

### inż. Dominik Barwacz

Absolwent Wydziału Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej Akademii Górniczo-Hutniczej w Krakowie, kierunek Automatyka i Robotyka z 2014 r. Obecnie Android Developer w krakowskiej firmie Softnauts. Interesuje się technologiami mobilnymi, ich praktycznym zastosowaniem w rozwiązaniach przemysłowych i szeroko rozumianymi zagadnieniami User Experience.

e-mail: dominik.barwacz@gmail.com



### dr hab. inż. Krzysztof Oprzędkiewicz

Urodzony w 1964 r. w Krakowie. Absolwent Akademii Górniczo-Hutniczej w Krakowie z 1988 r. (Wydział Elektrotechniki, Automatyki i Elektroniki, kierunek studiów: Elektronika, specjalność Automatyka). Doktorat na Wydziale Elektrotechniki, Automatyki i Elektroniki AGH w 1995 r., dyscyplina naukowa: Automatyka i Robotyka. Habilitacja na Wydziale Elektrotechniki, Automatyki, Informatyki i Elektroniki AGH w 2009 r., dyscyplina naukowa: Automatyka i Robotyka. Aktualnie zatrudniony na stanowisku profesora nadzwyczajnego w Katedrze Automatyki i Inżynierii Biomedycznej na Wydziale Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej AGH oraz w Państwowej Wyższej Szkole Zawodowej w Tarnowie. Zainteresowania naukowe: problemy sterowania systemami dynamicznymi o niepewnych parametrach, sterowanie cyfrowe, automatyka przemysłowa, systemy dynamiczne rzędu ułamkowego.

e-mail: kop@agh.edu.pl

