

# Mobilny robot manipulacyjny wykorzystujący technologie Internetu Rzeczy w systemie sterowania i monitorowania

Damian Dudek, Robert Kazała, Paweł Strączyński

Politechnika Świętokrzyska, Wydział Elektrotechniki Automatyki i Informatyki, al. Tysiąclecia Państwa Polskiego 7, 25-314 Kielce

**Streszczenie:** W pracy przedstawiono autorską konstrukcję mobilnego robota manipulacyjnego o sześciokołowym napędzie i sześciu stopniach swobody ramienia. Opisano modułowy i skalowalny system sterowania wykorzystujący technologie Internetu Rzeczy, takie jak sieć Wi-Fi, Ethernet oraz protokół MQTT. Przedstawiono strukturę oraz sposób przesyłania komunikatów związanych z odczytem danych z czujników oraz zapisem do układów wykonawczych. Zaprezentowano algorytmy wykorzystane do sterowania robotem oraz aplikację sterującą robotem dla komputera PC. Opisano zalety przedstawionego systemu sterowania.

**Słowa kluczowe:** robot mobilny, Internet rzeczy, protokoły komunikacyjne, sterowanie

## 1. Wprowadzenie

Internet Rzeczy IoT (ang. *Internet of Things*) jest stosunkowo nową koncepcją, w której przedmioty mogą wymieniać między sobą informacje z wykorzystaniem sieci Internet [1, 2]. Dostęp do danych jest możliwy z każdego miejsca, które jest objęte zasięgiem sieci. Taką ideę można wykorzystać na etapie projektowania systemów sterowania różnych urządzeń. Rozwiązanie takie stosowane jest często w instalacjach automatyki budynkowej, próbuje się je wprowadzać w urządzeniach gospodarstwa domowego. Idea ta jest wprowadzana do rozwiązań stosowanych w zakładach przemysłowych pod nazwą Przemysł 4.0 [3]. W przypadku rozwiązań zgodnych z koncepcją Internetu Rzeczy unika się hermetyzacji układu sterowania. Rozbudowa systemu jest zwykle bezproblemowa.

Rozwiązania opracowane dla Internetu Rzeczy można też wykorzystać w systemach sterowania robotów mobilnych. W istniejących konstrukcjach robotów najczęściej stosowane są zamknięte rozwiązania systemów sterowania, w których nie ma bezpośredniego dostępu z sieci Internet do sygnałów sterujących robotem [4]. Podejście takie powoduje, że utrudnione jest serwisowanie tych urządzeń oraz integracja z innymi systemami. Monitorowanie pracy tego typu konstrukcji wymaga zazwyczaj wykorzystania specjalizowanego oprogramowania. Wykorzystanie koncepcji Internetu Rzeczy pozwala na obniże-

nie kosztów i uproszczenie integracji systemu sterowania robota z jego otoczeniem.

Przykładem takiego rozwiązania jest system operacyjny ROS [5]. Wykorzystuje on w warstwie wymiany informacji mechanizm przekazywania komunikatów z wykorzystaniem sieci komputerowej. Nie jest to jednak rozwiązanie w pełni zgodne z koncepcją Internetu Rzeczy, gdyż w warstwie sterowania wykorzystuje własny system wymiany komunikatów, który wymaga dodatkowych systemów, umożliwiających publikowanie danych w sieci Internet. Innym przykładem są rozwiązania prezentowane w pracach [6, 7], gdzie wykorzystywany jest protokół MQTT, stosowany do wymiany danych między urządzeniami w Internecie Rzeczy [2]. Zaletą tych rozwiązań jest oparcie się na standardowym protokole, który pozwala na bezpośredni dostęp do systemu sterowania i danych udostępnianych przez robota z sieci Internet. W oparciu o tą ideę opracowany został w ramach przygotowań do zawodów European Rover Challenge oraz projektów [8, 9] system sterowania mobilnego robota manipulacyjnego, prezentowany w dalszej części pracy.

W ramach pracy przedstawiona została konstrukcja podwozia i ramienia mobilnego robota manipulacyjnego. Zaprezentowano architekturę sprzętową systemu sterowania. Przedstawiono wykorzystany protokół sieciowy oraz system komunikatów. Zaprezentowano także oprogramowanie sterujące podwoziem oraz ramieniem robota.

## 2. Budowa pojazdu

Projekt wielozadaniowego robota manipulacyjnego wymaga uwzględnienia warunków środowiskowych pracy tego urządzenia oraz obszaru jego zastosowań. Odpowiednio sprecyzowane założenia projektowe powinny dodatkowo uwzględniać zagadnienia obejmujące sposób obsługi urządzenia oraz sposób komunikacji między modułami układu. Dla konstrukcji

### Autor korespondujący:

Robert Kazała, rkazała@tu.kielce.pl

### Artykuł recenzowany

nadesłany 10.10.2016 r., przyjęty do druku 15.12.2016 r.



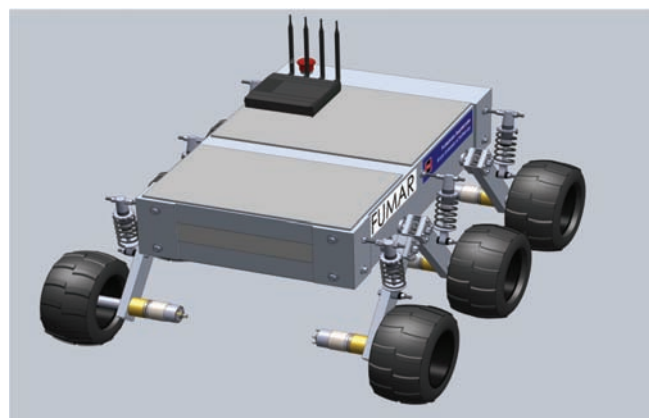
Zezwala się na korzystanie z artykułu na warunkach licencji Creative Commons Uznanie autorstwa 3.0

omawianego robota mobilnego sformułowano następujące założenia:

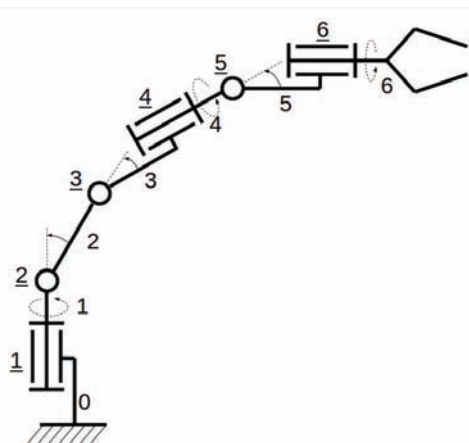
- gabaryty umożliwiające łatwy transport pojazdu,
- manipulator umożliwiający wykonywanie operacji na wysokości do 1,5 m od podłoża,
- układ jezdny pozwalający na poruszanie się w zróżnicowanym terenie,
- lekkość konstrukcji – masa poniżej 40 kg,
- układ zasilania pozwalający na 2 godziny ciągłej pracy,
- zmaksymalizowanie niezawodności urządzenia,
- modułowość układu sterowania – pełna niezależność podsystemów robota,
- łatwość rozbudowy urządzenia,
- możliwość zdalnego sterowania robotem przez sieć Internet,
- możliwość jednoczesnego sterowania robotem przez wielu operatorów,
- łatwość dostępu do danych rejestrowanych przez robota,
- niska cena.

## 2.1. Konstrukcja mechaniczna

Prezentowany robot mobilny składa się z sześciokołowej platformy jezdnej oraz osadzonego na niej ramienia manipulacyjnego. Konstrukcję podwozia pojazdu przedstawiono na rys. 1. Bazę stanowi aluminiowa rama zapewniająca wymaganą sztywność, wypełniona spienionym PVC. Zastosowanie w układzie jezdnych kół nieskrętnych ma na celu uproszczenie konstrukcji podwozia, co zapewnia poprawę niezawodności oraz umożliwia uzyskanie założonej masy urządzenia. W celu umożliwienia



Rys. 1. Projekt konstrukcji platformy jezdnej robota mobilnego  
Fig. 1. The structural design of the mobile robot chassis



Rys. 2. Schemat łańcucha kinematycznego zbudowanego manipulatora

Fig. 2. Diagram of manipulator kinematic chain

efektywnego poruszania w trudnym terenie, platforma wyposażona została w niezależny system amortyzacji dla każdego z kół. Do napędu układu jezdny robota wykorzystano silniki prądu stałego z przekładniami planetarnymi.

Realizację zadań manipulacyjnych umożliwiła zainstalowana na robocie manipulator. Jako materiał konstrukcyjny wykorzystano profile aluminiowe. Pozwoliło to uzyskać wymaganą sztywność oraz smukłość ramienia. Zaprojektowane ramię ma sześć stopni swobody. Dzięki temu kiść manipulatora może osiągnąć dowolną pozycję i orientację w przestrzeni roboczej ramienia. Do przemieszczania poszczególnych członów ramienia wykorzystano silniki szczotkowe prądu stałego oraz przekładnie zębate o odpowiednio dobranym przełożeniu. Sposób wykonywania ruchów względnych przez poszczególne pary kinematyczne uzyskanego łańcucha przedstawiono na rys. 2.

Na podstawie schematu kinematycznego manipulatora przygotowano projekt wykonawczy. W celu zapewnienia możliwości sterowania każdego ze stopni swobody w zamkniętej pętli, każdy przegub wyposażono w element służący do pomiaru jego położenia. Rozmieszczenie jednostek napędowych oraz elementów pomiarowych przedstawiono na rys. 3.

Na podstawie sformułowanych założeń projektowych oraz modeli wykonanych w środowisku CAD zbudowany został prototyp robota (rys. 4).

## 2.2. System sterowania

W celu zapewnienia możliwie największej niezawodności robota opracowano modułowy system sterowania, umożliwiający redundancję poszczególnych podsystemów. Utworzony wieloprocessorowy system rozproszony charakteryzuje się możliwością wykonywania poleceń przez nieuszkodzone moduły podczas awarii pozostałych. Na etapie projektu wyróżniono następujące moduły:

- moduł orientacji w przestrzeni i sterowania platformą jezdną,
- moduł sterowania pracą manipulatora,
- moduł zarządzania zasilaniem.

### 2.2.1 Wymiana danych

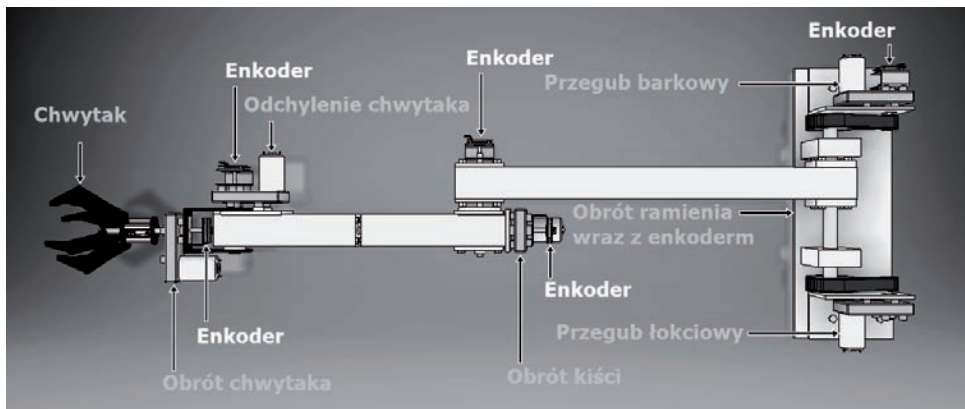
W prezentowanym systemie sterującym wyróżnić można komunikację na trzech poziomach:

- wymiana danych między robotem a stacją kontroli,
- wymiana informacji między podsystemami wewnątrz robota,
- akwizycja danych z sensorów.

Sposób wymiany informacji mikrokontrolera z czujnikami jest z góry określony przez producenta układów sensorycznych. Wykorzystywane są zwykle lokalne interfejsy komunikacyjne m.in. I<sup>2</sup>C, SPI, UART. W przypadku komunikacji na pozostałych poziomach możliwości są zwykle większe.

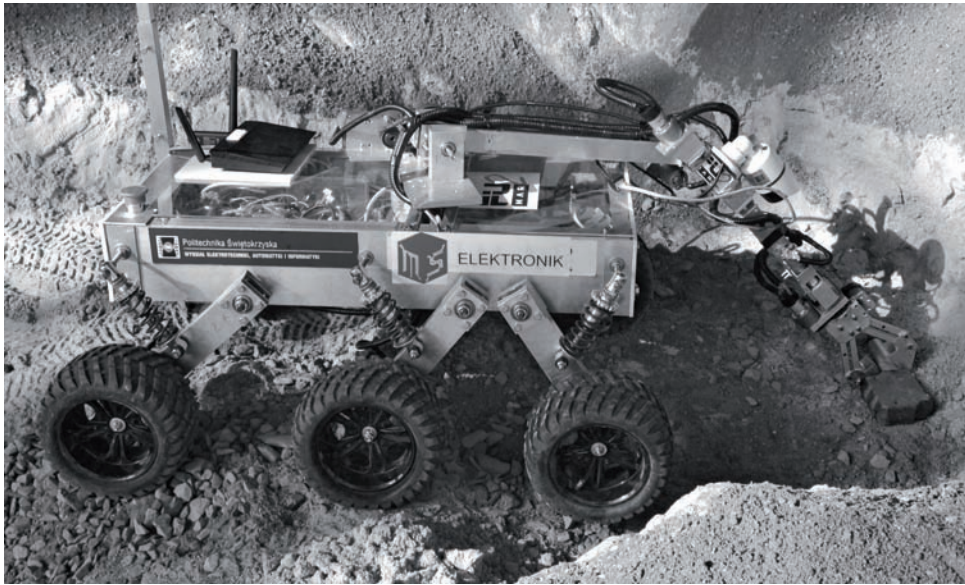
Mając na uwadze możliwość sterowania pojazdem nie tylko lokalnie, ale również na znaczne odległości, postanowiono system komunikacji oprzeć o ideę Internetu Rzeczy (IoT) [1, 2]. Identyfikowalność urządzenia pozwala na wykorzystywanie światowej sieci Internet do sterowania zdalnego. Przelamuje to bariery odległości występujące w przypadku innych rozwiązań. Idąc krok dalej można koncepcję tę zastosować także w komunikacji wewnątrzpokołowej. Przy takim rozwiązaniu nie tylko cały robot będzie dostępny w sieci jako urządzenie, ale każdy z jego podsystemów z osobna.

Kolejnym problemem podczas projektowania systemu sterowania jest określenie modelu wymiany informacji między urządzeniami. W przypadku rozległych systemów sterowania, zawierających wiele elementów składowych, stale wymieniających między sobą informacje, komunikacja bezpośrednia między urządzeniami P2P (ang. *point to point*) jest mało elastyczna i problematyczna w implementacji. Wymiana danych w architekturze klient-serwer może okazać się w tym przypadku również mało efektywna. Szczególnie widoczne jest to, gdy wiele



Rys. 3. Projekt budowy manipulatora

Fig. 3. Construction project of the manipulator



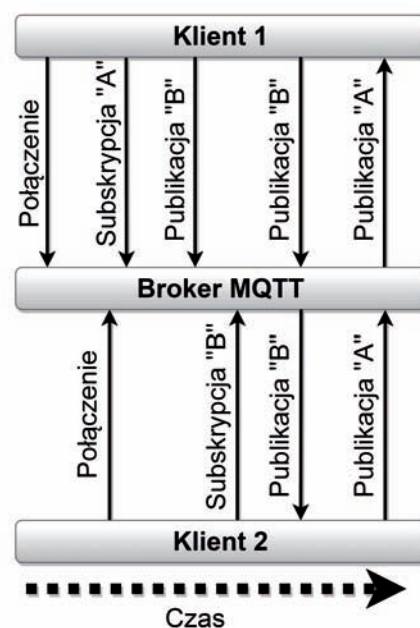
Rys. 4. Robot mobilny

Fig. 4. Mobile robot

urządzeń współdzieli te same dane. W sytuacji takiej lepszym wyborem jest oparcie systemu komunikacji o model publikuj-subskrybuj (ang. *publish/subscribe*).

Model ten przyjmuje formę gwiazdy w topologii logicznej sieci. Urządzenie (ang. *publisher*), będące elementem systemu przesyła informacje do struktury zwanej brokerem komunikatów, będącej punktem centralnym (koncentratorem) wymiany informacji między urządzeniami. Zadaniem brokera jest natychmiastowe przesłanie otrzymanej informacji do wszystkich urządzeń, które uprzednio subskrybowały jej otrzymywanie (ang. *subscribers*). Informacje w sieci są kategorizowane za pomocą tematów wiadomości (ang. *topics*). Zaletą tak zorganizowanej sieci jest zwiększona szybkość i prostota wymiany informacji, związana z niewielką nadmiarowością ramek wymienianych komunikatów.

Model *publikuj-subskrybuj* wykorzystywany jest między innymi przez protokoły XMPP, ZeroMQ czy MQTT [6]. To właśnie ostatni z wymienionych protokołów wykorzystano w prezentowanym systemie. Za jego wyborem przemawia dostępność implementacji klienta tego protokołu dla większości najpopularniejszych języków programowania m.in. C/C++, .Net, C#, Python, JavaScript czy Java. Opracowywaniem bibliotek klienckich protokołu MQTT na zasadzie wolnego oprogramowania zajmuje się fundacja Eclipse w ramach projektu PAHO. Ideę wymiany informacji w oparciu o model publikuj-subskrybuj i protokół MQTT przedstawiono na rys. 5.



Rys. 5. Wymiana danych na podstawie modelu publish-subscribe (MQTT)

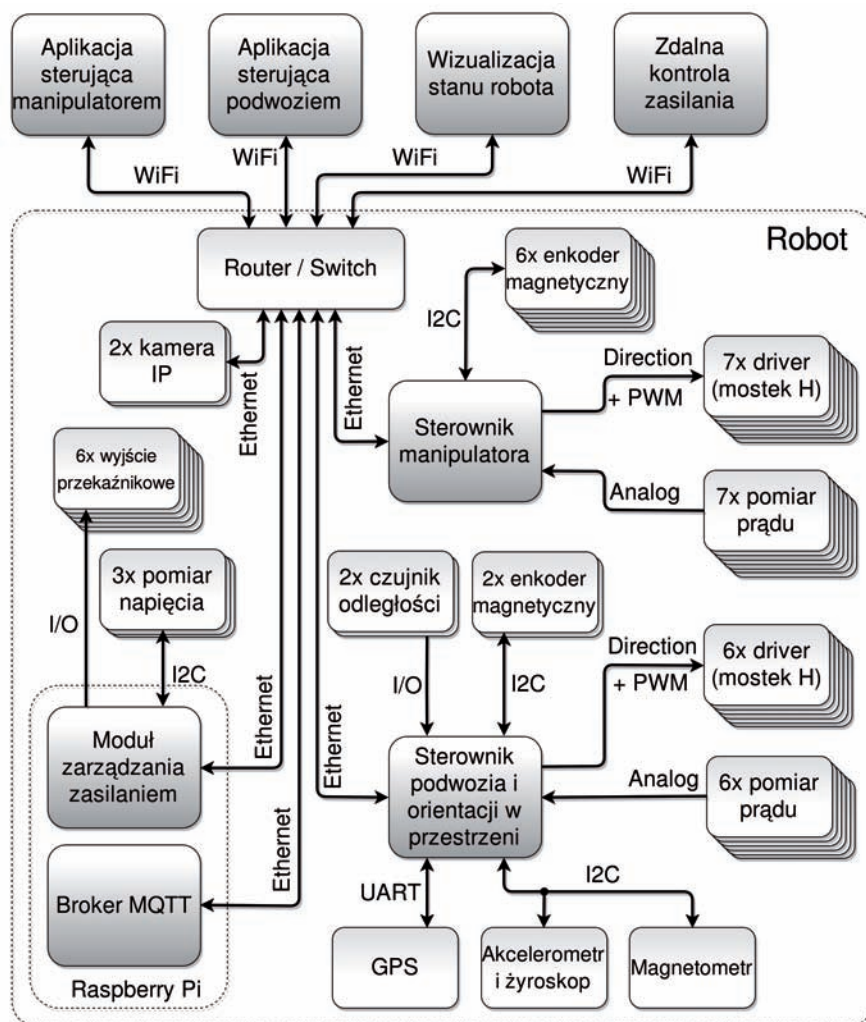
Fig. 5. Data exchange based on publish-subscribe model (MQTT)

### 2.2.2 Architektura sprzętowa systemu sterowania

Jako główny komputer pokładowy, obsługujący broker komunikatów protokołu MQTT, wykorzystano komputer jednopłytkowy Raspberry Pi z systemem operacyjnym Raspbian. Ponadto, minikomputer pełni rolę modułu nadzorczego umożliwiającego wyłączanie pozostałych podsystemów oraz monitorowanie stanu akumulatorów litowo-polimerowych. Pozostałe moduły są zarządzane przez mikrokontrolery TM4C1294 firmy Texas Instruments. Wybór układów sterujących został podyktowany przede wszystkim liczbą modułów peryferyjnych. Z punktu widzenia zastosowania mikrokontrolerów w układach sterująco-sensorycznych wymagane było wyposażenie ich w:

- ośmiokanałowy sprzętowy moduł PWM,
- interfejs komunikacyjny I<sup>2</sup>C,
- interfejs UART,
- sprzętowy kontroler Ethernet,
- dwa szybkie przetworniki A/C (1Mps) o rozdzielczości 12 bitów, obsługujące 8 kanałów przetwarzania.

Dodatkowym atutem wybranego mikrokontrolera jest nowoczesny rdzeń ARM Cortex-M4F. Architektura sprzętowa systemu sterowania robotem przedstawiona została na rys. 6.



Rys. 6. System sterowania robotem  
Fig. 6. Robot control system

## 3. Oprogramowanie sterujące

Opracowane oprogramowanie służące do sterowania robotem pracuje w trzech warstwach. Pierwszą stanowi oprogramowanie niskopoziomowe przeznaczone dla mikrokontrolerów. Warstwa druga obejmuje oprogramowanie wysokopoziomowe wbudowanego komputera sterującego oraz broker komunikatów. Trzecią warstwę stanowi oprogramowanie zewnętrznych systemów, pozwalających na zdalne sterowanie, zarządzanie i wizualizację pracy robota. Ze względu na obsługiwany moduł sterowania, oprogramowanie robota można podzielić na służące do: sterowania manipulatorem, sterowania podwoziem, zarządzania energią.

Przy tworzeniu oprogramowania przyjęto, że programy sterujące dla mikrokontrolerów napisane będą w języku C, z wykorzystaniem systemu operacyjnego czasu rzeczywistego FreeRTOS. Komunikacja między mikrokontrolerami odbywa się za pomocą sieci Ethernet, w oparciu o implementację stosu TCP/IP (biblioteka lwIP). Aplikacje sterujące dla komputerów PC napisano w języku Python z wykorzystaniem bibliotek interfejsu graficznego wxPython. Do obsługi protokołu MQTT wykorzystano bibliotekę PAHO MQTT Client.

### 3.1. Moduł do nawigacji i sterowania podwoziem

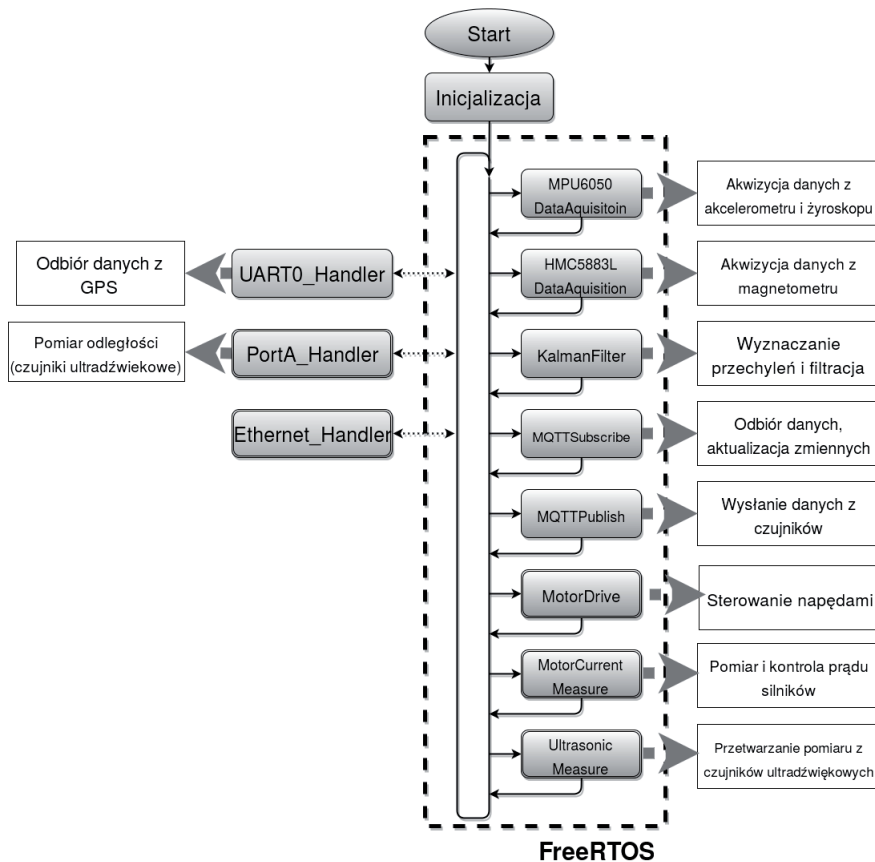
Stworzony podsystem nawigacji i sterowania podwoziem zapewnia obsługę sześciu niezależnych kół podwozia, poprzez sterowanie napędami oraz pomiar pobieranego przez nie prądu. Kolejnym zadaniem modułu jest pomiar zmiennych środowi-

skowych otoczenia robota, takich jak położenie względem przeszkód czy lokalizacja na mapie. W warstwie nadrzędnej wykorzystującej komputer PC, zapewnia wizualizację stanu modułu i zmiennych środowiskowych oraz dostarczanie informacji nawigacyjnych dla operatora urządzenia.

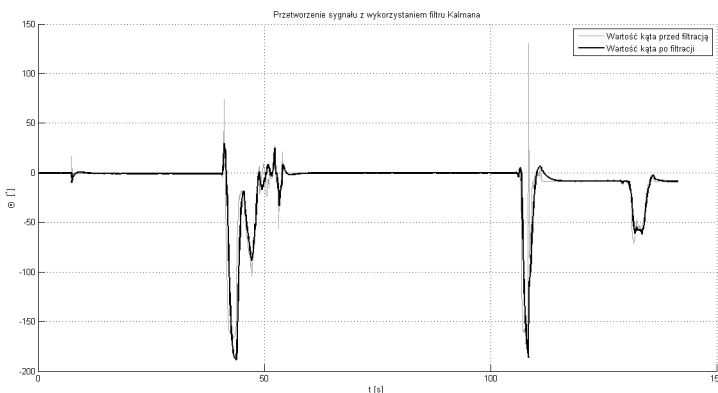
Wykorzystanie systemu operacyjnego w stworzonym oprogramowaniu dla mikrokontrolera, pozwoliło na zorganizowanie kluczowych zadań w postaci osobnych wątków. Strukturę programu przedstawiono na rys. 7.

Poza podejmowaniem cyklicznych zadań, zaplanowanych w poszczególnych wątkach systemu operacyjnego (np. akwizycja danych z czujnika, przesłanie wiadomości), oprogramowanie zapewnia obsługę określonych akcji (np. odbiór informacji z odbiornika GPS, odbiór danych z modułu Ethernet) wykorzystując system przerwań. Do zadań mikrokontrolera poza działaniami związanymi ze sterowaniem, akwizycją i przesyłaniem danych z czujników, należy również wstępne przetwarzanie zbieranych informacji. Przykładem może być wyznaczanie orientacji robota w przestrzeni (przechylenie wzdłużne i poprzeczne) na podstawie fuzji danych z czujników inercjalnych w oparciu o algorytm filtru Kalmana. Do opisu procesu wykorzystano model stanowy przedstawiony w publikacji [10]. Równanie stanu ma postać:

$$\begin{bmatrix} \Theta(k) \\ g(k) \end{bmatrix} = \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Theta(k-1) \\ g(k-1) \end{bmatrix} + \begin{bmatrix} dt \\ 0 \end{bmatrix} \cdot \omega$$



Rys. 7. Schemat funkcjonalny oprogramowania mikrokontrolera – sterownik podwozia  
 Fig. 7. Functional diagram of microcontroller software – chassis controller



Rys. 8. Przetwarzanie danych z wykorzystaniem filtru Kalmana  
 Fig. 8. Processing of data using the Kalman Filter



Rys. 9. Okno główne aplikacji do nawigacji i sterowania jazdą robota  
 Fig. 9. Main application window to navigate and robot steering

gdzie  $\omega$  – prędkość kątowna odczytana na podstawie pomiaru z żyroskopu,  $\Theta$  – odchylenie kątowne wyznaczone na podstawie danych z akcelerometru,  $g$  – dryft żyroskopu.

Na rysunku 8 przedstawiono efekt działania zaimplementowanego na mikrokontrolerze filtru.

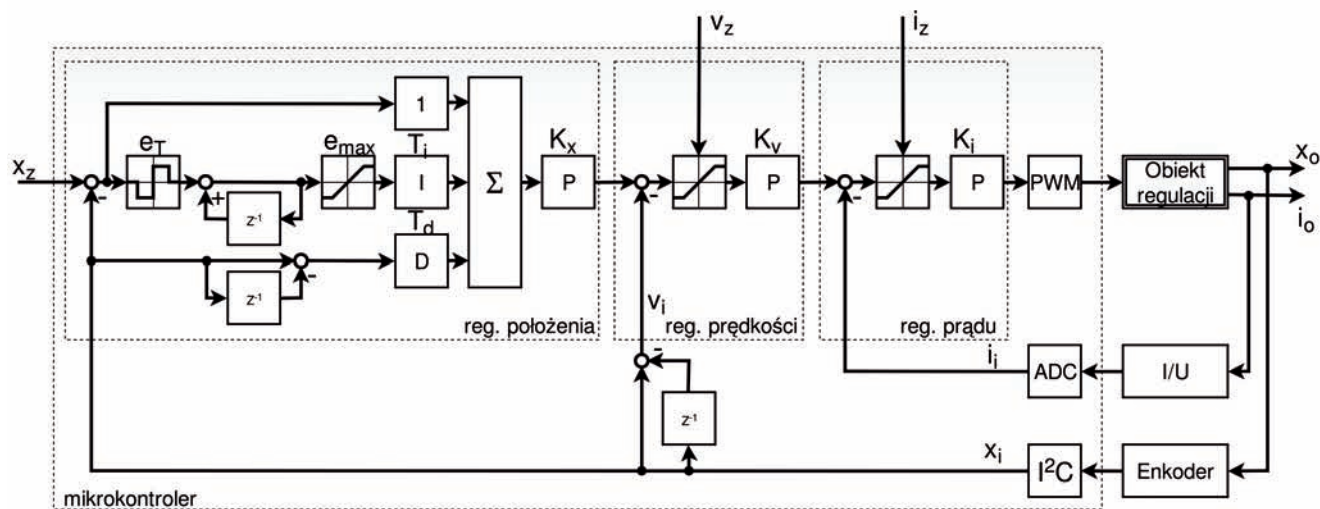
Do współpracy z układem mikroprocesorowym przygotowana została specjalnie dedykowana aplikacja do nawigacji i sterowania jazdą dla komputera PC. Główne okno aplikacji przedstawione zostało na rys. 9. Aplikacja umożliwia sterowanie pracą platformy jezdnej robota. Ponadto odbiera ona dane przesyłane przez mikrokontroler i dokonuje ich wizualizacji. Operatorowi dostarczane są m.in. informacje o:

- stanach wewnętrznych robota (prądy silników, napięcia akumulatorów),
  - orientacji robota w przestrzeni (współrzędne GPS, lokalizacja pojazdu na mapie, orientacja względem podłoża – sztuczny horyzont, kompas).
- Aplikacja została napisana w języku Python z wykorzystaniem środowiska graficznego wxPython. Do sterowania robotem wykorzystano standardowy kontroler komputerowy do gier firmy Microsoft.

### 3.2 Układ sterowania manipulatorem

Wydzielony moduł sterowania manipulatorem zapewnia pomiar położenia, prędkości i prądów każdego z napędów ramienia oraz chwytaka. Umożliwia kontrolę i sterowanie położeniem oraz prędkościami członów manipulatora. Podobnie jak w przypadku modułu nawigacji, podsystem sterowania manipulatorem został podzielony na warstwę niskopoziomą, którą stanowi mikrokontroler wraz z oprogramowaniem stworzonym w języku C oraz warstwę kontrolno-sterującą zrealizowaną na komputerze klasy PC – zapewniającą wizualizację stanów wewnętrznych modułu.

Dla zapewnienia pełnej kontroli nad manipulatorem konieczna jest znajomość zarówno wzajemnych pozycji członów, prędkości jak i wartości płynących prądów. Do pomiarów położenia i wyznaczania prędkości wykorzystano magnetyczne enkodery absolutne firmy AMS, każdy o 14-bitowej rozdzielczości. Komunikacja z układami odbywa się za pomocą interfejsu I<sup>2</sup>C. Informacja o natężeniach prą-



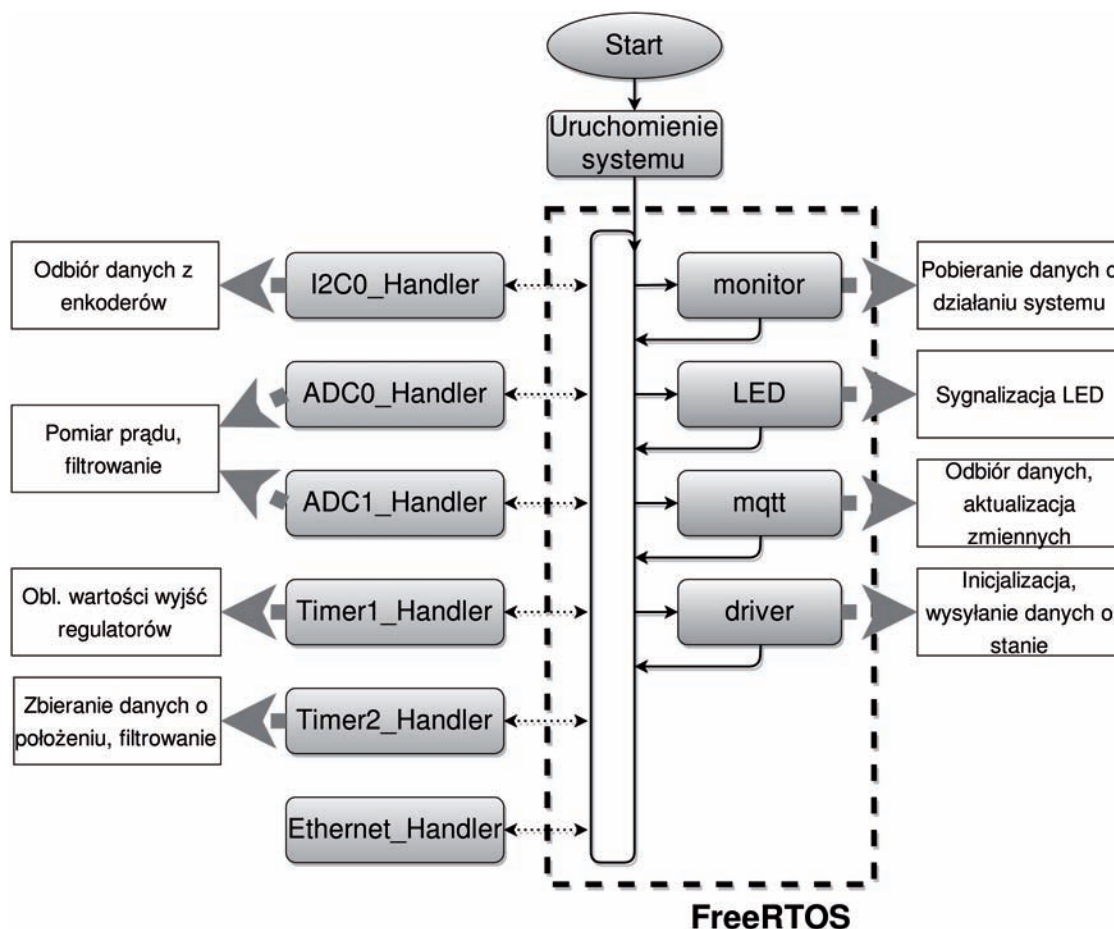
Rys. 10. Układ regulacji pojedynczego stopnia swobody manipulatora  
 Fig. 10. Control system of the manipulator single degree of freedom

dów dostarczanych do silników obsługujących manipulator, pochodzi z dwóch układów przetworników ADC wbudowanych w mikrokontroler, których wejścia podłączono do układów konwertujących.

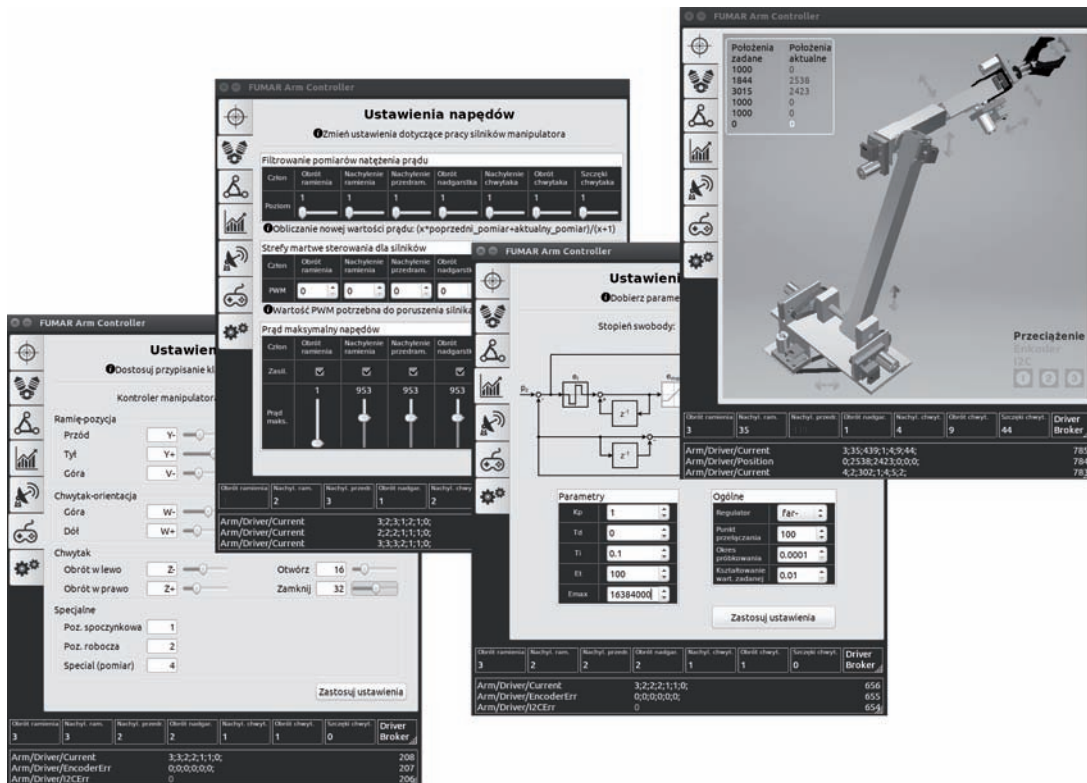
Zebrane dane pomiarowe są przetwarzane i wprowadzane do bloku programowego, realizującego regulator kaskadowy położenia, prędkości i prądu (rys. 10). Układ regulacji poło-

żenia stanowi regulator pętli zewnętrznej, natomiast w pętli wewnętrznej pracuje uproszczony regulator prędkości. W celu ograniczenia natężenia prądu napędów zastosowano regulator prądu sterujący wielkością wyjściową.

W celu zapewnienia poprawnej pracy regulatora, zdecydowano się wprowadzić następujące modyfikacje do struktury klasycznego regulatora PID [11, 12]:



Rys. 11. Graf przepływu sterowania dla oprogramowania sterownika manipulatora  
 Fig. 11. Control flow graph for manipulator driver software



Rys. 12. Okna aplikacji służącej do kontroli manipulatora  
Fig. 12. Windows application used to control the manipulator

- zabezpieczenie przeciw nadmiernej wartości błędu całkowania (ang. *windup*) poprzez wyłączenie działania całującego do czasu wkroczenia zmiennej w obszar sterowalny,
- dodatkowy, tzw. *anti-windup* przez określenie maksymalnego błędu całkowania,
- zabezpieczenie przed niekorzystnym wpływem pojawienia się skokowej zmiany wartości zadanej, poprzez obliczanie pochodnej jedynie z wartości wyjściowej.

Oprogramowanie sterujące pracą mikrokontrolera realizuje zadania związane z akwizycją danych, utrzymywaniem docelowej pozycji manipulatora, przemieszczaniem członów, kontrolą napędów oraz komunikacją za pomocą sieci Ethernet. W celu uproszczenia procesu szeregowania zadań wykorzystano system operacyjny FreeRTOS. Schemat działania oprogramowania przeznaczonego dla mikrokontrolera przedstawia rys. 11.

W warstwie sterowania nadrzędnego możliwa jest zmiana nastaw każdego z regulatorów, z wykorzystaniem aplikacji napisanej dla komputera PC. Na rysunku 12 zdemonstrowano możliwości konfiguracyjne opracowanego oprogramowania. Główna karta aplikacji przeznaczona jest do wizualizacji procesów zachodzących w manipulatorze, pozostałe karty pozwalają na zmianę poszczególnych parametrów regulatora.

Zbudowany układ sterowania umożliwia zdalny dostęp do parametrów przez sieć Ethernet i protokół MQTT. Możliwa jest więc współpraca z wieloma jednocześnie uruchomionymi programami, z których jeden może być odpowiedzialny za ruch ramienia, inny za jego kontrolę pod względem obciążeń lub niedozwolonych pozycji członów. Kolejna z aplikacji może pełnić rolę wizualizacji działania układu regulacji. W tabeli 1 przedstawiono przykładowe tematy subskrybowane przez sterownik manipulatora robota, natomiast w tabeli 2 wybrane informacje przekazywane przez sterownik do aplikacji współpracujących.

Tabela 1. Tematy komunikatów subskrybowanych przez sterownik manipulatora

Table 1. Topics messages subscribed by the manipulator controller

<i>Arm/Controller/Position</i>	Zadane pozycje członów
<i>Arm/Controller/MaxSpeed</i>	Prędkości członów
<i>Arm/Controller/PIDParameters</i>	Parametry regulatorów
<i>Arm/Controller/MaxCurrent</i>	Prądy maksymalne
<i>Arm/Controller/Position.Max.Min</i>	Pozycje maksymalne i minimalne członów

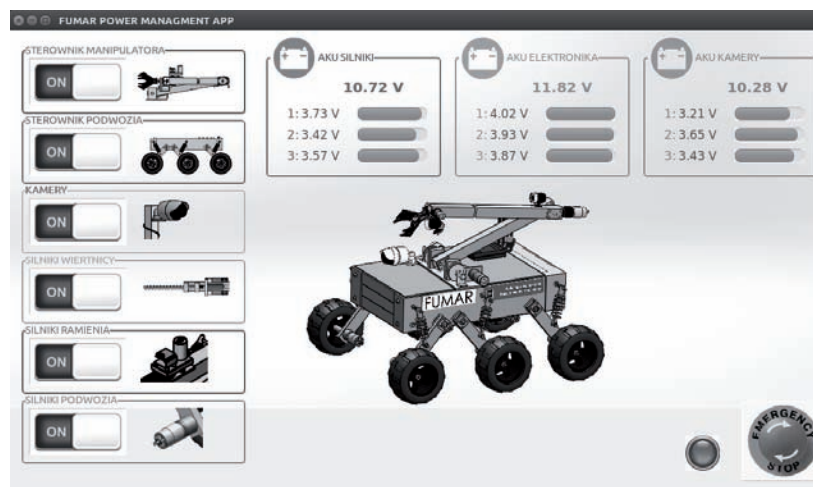
Tabela 2. Tematy komunikatów publikowanych przez sterownik manipulatora

Table 2. Topics messages published by the manipulator controller

<i>Arm/Driver/Position</i>	Pozycje aktualne członów
<i>Arm/Driver/Speed</i>	Prędkości aktualne członów
<i>Arm/Driver/Current</i>	Prądy aktualne napędów
<i>Arm/Driver/EncoderErr</i>	Komunikaty błędów enkoderów

### 3.3. System zarządzania zasilaniem

Część programowa systemu zarządzającego energią robota, podobnie jak w przypadku pozostałych modułów, składa się z oprogramowania pracującego na pokładzie robota (minikomputer Raspberry Pi) oraz aplikacji uruchamianej na komputerze PC operatora.



Rys. 13. Okno aplikacji służącej do zarządzania zasilaniem  
Fig. 13. The application window used for power management

Oprogramowanie pracujące na pokładzie robota uruchamiane jest na minikomputerze w postaci demona unixowego. Zarówno ten program jak i współpracująca z nim aplikacja nadrzędna napisane zostały w języku Python. System zarządzania zasilaniem monitoruje stan akumulatorów realizując pomiar napięć na każdej z cel (trzy 18-bitowe czterokanałowe przetworniki A/C) oraz zarządzania zasilaniem wewnątrz robota. Okno główne aplikacji nadzorczej przedstawiono na rys. 13.

## 4. Podsumowanie

Przedstawiony w pracy system sterowania mobilnego robota manipulacyjnego pozwala na łatwe zarządzanie robotem oraz zapewnia dużą skalowalność systemu. Efekt ten uzyskano dzięki modularnej budowie oraz wykorzystaniu sieci Ethernet i protokołu MQTT. Przedstawiona architektura pozwala na łatwe dołączanie do systemu nowych układów sensorycznych i wykonawczych, bez konieczności konfiguracji ze strony użytkownika. System komunikacji automatycznie rejestruje i rozsyła wymagane komunikaty. Tylko moduł nadający i odbierający muszą mieć skonfigurowane komunikaty niezbędne do ich pracy.

Wykorzystanie standardowych protokołów sieciowych pozwala na zdalne sterowanie robotem poprzez sieć Internet. Zapewnia to możliwość sterowania przez osoby znajdujące się w dowolnym miejscu na Ziemi, które mają dostęp do sieci.

Opracowana architektura pozwala w przyszłości na łatwe rozbudowanie robota o systemy zapewniające autonomię jego pracy, które mogą być zainstalowane lokalnie wewnątrz robota, lub na zewnętrznych komputerach. Zastosowanie zewnętrznych systemów przetwarzania danych i podejmowania decyzji, pracujących w sieci Internet, pozwoli na stosowanie złożonych algorytmów rozpoznawania otoczenia i wyznaczania trajektorii ruchu.

## Podziękowania

Praca finansowana z funduszy Wydziału Elektrotechniki, Automatyki i Informatyki Politechniki Świętokrzyskiej na działalność kół naukowych, w ramach przygotowań do udziału Studenckiego Koła Naukowego FUPLA w zawodach European Rover Challenge. Pragniemy podziękować członkom SKN FUPLA za pomoc przy budowie i uruchamianiu robota.

## Bibliografia

1. Czajkowski R., Nowakowski W., *IoT jako naturalna ewolucja Internetu*, „Elektronika: konstrukcje, technologie, zastosowania”, Vol. 57, Nr 4, 2016, 28–32, DOI: 10.15199/13.2016.4.6
2. Al-Fuqaha A., Guizani M., Mohammadi M., Aledhari M., Ayyash M., *Internet of Things: A Survey on Enabling Technologies, Protocols and Applications*, “IEEE Communications Surveys & Tutorials”, Vol. 17, No. 4, 2015, 2347–2376, DOI: 10.1109/COMST.2015.2444095.
3. Wan J., Tang S., Shu Z., Li D., Wang S., Imran M., Vasylakos A., *Software-Defined Industrial Internet of Things in the Context of Industry 4.0*, “IEEE Sensor Journal”, Vol. 16, No. 20, 2016, 7373–7380, DOI: 10.1109/JSEN.2016.2565621.
4. Kasprzyczak L., Trenczek S., *Górnictwo mobilny robot inspekcyjny do monitorowania stref zagrożonych wybuchem*, „Pomiary Automatyka Robotyka”, R. 15, Nr 2, 2011, 431–440.
5. Quigley M., Gerkey B.P., Conley K., Faust J., Foote T., Leibs J., Berger E., Wheeler R., Ng A.Y., *ROS: an open source Robot Operating System*, [in:] ICRA workshop on open source software, 2009.
6. Kazala R., *Wykorzystanie protokołu MQTT w systemie sterowania mobilnego robota transportowego*, „Logistyka”, 3/2015, 2118–2127.
7. Kazala R., Taneva A., Petrov M., Penkov S., *Wireless Network for Mobile Robot Applications*, IFAC-PapersOnLine, Vol. 48, No. 24, 2015, 231–236
8. Dudek D., *System sterowania ramieniem robota mobilnego wykorzystujący mikrokontroler Arm i protokół MQTT*, Praca dyplomowa, Politechnika Świętokrzyska 2016
9. Strączyński P., *System nawigacji i sterowania robotem mobilnym wykorzystujący mikrokontroler Arm*, Praca dyplomowa, Politechnika Świętokrzyska 2016
10. Kędziński J., *Filtr Kalmana – zastosowania w prostych układach sensorycznych*, Koło Naukowe Robotyków KoNaR, Politechnika Wroclawska, 2007.
11. Azar A.T., Serrano F.E., *Design and Modeling of Anti Wind Up PID Controllers*, 2015, Springer International Publishing, DOI: 10.1007/978-3-319-12883-2\_1.
12. Peng Y., Vrancic D., Hanus R., *Anti-windup, bumpless, and conditioned transfer techniques for PID controllers*, “IEEE Control Systems”, Vol 16, No. 4, 1996, 48–57, DOI: 10.1109/37.526915.



# Mobile Handling Robot That Uses The Internet of Things Technology in The Control and Monitoring System

**Abstract:** The paper presents an original design of the six wheeled mobile robot equipped with a handling arm with six degrees of freedom. Describes a modular and scalable control system using the Internet of Things technologies such as Wi-Fi, Ethernet network and MQTT protocol. The structure of messages associated with reading data from sensors and writing to actuators and how they are transmitted. It presents the algorithms used to control the robot and robot control application for PC computer. Describes the advantages of the proposed control system.

**Keywords:** mobile robot, Internet of Things, communication protocols, control

## mgr inż. Damian Dudek

damiand@tu.kielce.pl

Doktorant w Zakładzie Urządzeń i Systemów Automatyki na Wydziale Elektrotechniki Automatyki i Informatyki Politechniki Świętokrzyskiej w Kielcach. Zainteresowania badawcze obejmują zagadnienia robotyki mobilnej oraz układów pomiarowych.



## dr inż. Robert Kazała

rkazala@tu.kielce.pl

Adiunkt w Zakładzie Urządzeń i Systemów Automatyki na Wydziale Elektrotechniki Automatyki i Informatyki Politechniki Świętokrzyskiej. Zajmuje się zagadnieniami projektowania systemów sterowania z wykorzystaniem sterowników PLC oraz komputerów przemysłowych, przetwarzania sygnałów, przetwarzania obrazów, systemami sterowania robotów mobilnych i manipulacyjnych. Sprawuje funkcję opiekuna studenckiego koła naukowego FUPLA.



## mgr inż. Paweł Strączyński

pstraczynski@tu.kielce.pl

Asystent w Zakładzie Urządzeń i Systemów Automatyki na Wydziale Elektrotechniki Automatyki i Informatyki Politechniki Świętokrzyskiej. Zainteresowania obejmują robotykę oraz systemy wbudowane.



