

# Platforma mobilna do badania interakcji człowiek-robot

Michał Maciejewski, Marek Niemiec, Łukasz Ryszka

Studenckie Koło Naukowe Robotyki SKaNeR, Politechnika Łódzka

**Streszczenie:** W artykule przedstawiono opis autonomicznego robota mobilnego, stworzonego z wielu różnych modułów sprzętowych i programowych. Platforma umożliwia realizację podstawowych zadań z zakresu interakcji człowieka z maszyną, takich jak synteza mowy oraz rozpoznawanie ruchów, jak również omijanie przeszkód oraz komunikacja bezprzewodowa. Część sprzętowo-programistyczna podzielona jest na trzy główne moduły: mikrokontroler z rodziny STM32 sterujący ruchem platformy, komputer wykonany w standardzie PC-104 umieszczony na platformie mobilnej odpowiedzialny za generowanie mowy oraz komputer PC, na którym uruchomiono aplikację stworzoną w LabVIEW, zarządzającą zdalnie zachowaniem robota. Takie podejście daje możliwość dalszej rozbudowy projektu dzięki jego skalowalności oraz modułowości. Konstrukcja została wyróżniona podczas zawodów robotów o randze ogólnopolskiej oraz regionalnej.

**Słowa kluczowe:** robot społeczny, interakcja człowiek-robot, Omnidirectional, STM32, LabVIEW, PC-104

## Wstęp

W ramach studenckiego projektu „Budowa robota społecznego” realizowanego przez członków Koła Naukowego Robotyki SKaNeR, powstała wielomodułowa platforma mobilna OmnIVOice (rys. 1), w której zaimplementowano następujące funkcje: generowanie mowy, zdalne sterowanie z wykorzystaniem joysticka lub ruchów ciała ludzkiego oraz unikanie kolizji z otoczeniem. To podstawowe umie-



Rys. 1. Robot mobilny OmnIVOice

Fig. 1. Mobile robot OmnIVOice

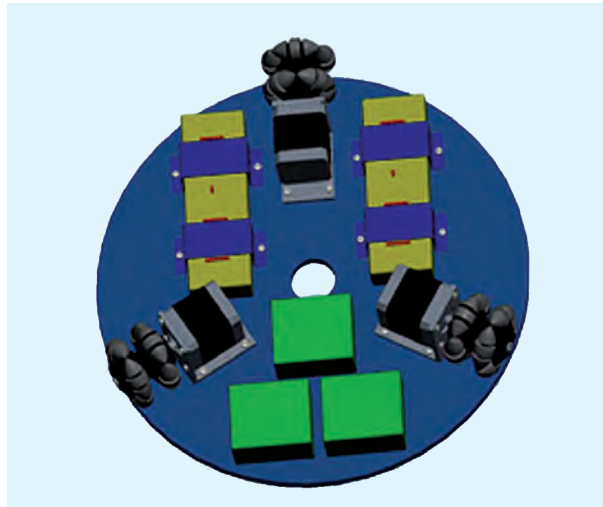
jetności umożliwiające badanie wzajemnej, możliwie naturalnej komunikacji między operatorem i robotem. Rozwój elektroniki, w tym różnego rodzaju czujników, jak również rosnąca moc obliczeniowa mikroprocesorów pozwoliły wykorzystać dotychczasowy dorobek naukowy z zakresu interakcji człowiek-maszyna do rozpoczęcia badań. Oprogramowanie do analizy i syntezy mowy, czy też systemy wizyjne pozwalające rozpoznawać gesty człowieka oraz położenie obiektów w przestrzeni stanowią duże wsparcie podczas realizacji tego typu zadań.

Roboty społeczne, w odróżnieniu od szeroko znanych i stosowanych robotów przemysłowych, swoim działaniem wspomagają ludzi w realizacji podstawowych, codziennych czynności życiowych. Znane są projekty robotów pracujących jako pomocnik osób starszych, osobisty asystent, czy też przewodnik po wystawie lub siedzibie firmy [1, 4]. Z reguły są to konstrukcje kosztowne, o dużych wymiarach, które trudno jest zrealizować w ramach „studenckiego” budżetu, i które nie mogą pracować w małych, ograniczonych przestrzeniach – na przykład na biurku podczas zajęć laboratoryjnych. W takim środowisku szczególnie ważne jest zapewnienie platformie mobilnej dużej swobody ruchu, co można uzyskać przez odpowiedni dobór struktury kinematycznej [1].

## Opis konstrukcji

### Kinematyka robota

Prace nad konstrukcją mechaniczną rozpoczęto od stworzenia modelu 3D (rys. 2) w programie Pro/ENGINEER Wildfire 5.0. Pozwoliło to na weryfikację wielu koncepcji,

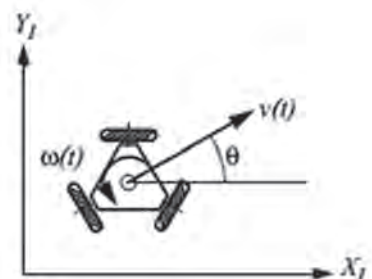


Rys. 2. Model 3D platformy

Fig. 2. 3D model of the platform

a następnie precyzyjne wykonanie elementów składowych w oparciu o pliki wynikowe.

Dokładność wykonania jest kluczowym wymaganiem dla zastosowanej struktury kinematycznej. Główne elementy konstrukcji wykonano z plexiglasu oraz aluminium, ze względu na dużą wytrzymałość przy jednocześnie niskiej wadze. Nie bez znaczenia były również walory estetyczne konstrukcji. Trzykołowy robot należy do klasy Omnidirectional, co zapewnia możliwość dowolnego wyboru wektorów  $v$  i  $\omega$ , w każdej chwili czasowej  $t$  (rys. 3). Ta właściwość robota jest osiągnięta dzięki specjalnemu rozmieszczeniu geometrycznemu kół (co  $120^\circ$  na obwodzie podstawy) oraz ich typowi – zastosowano tzw. koła szwedzkie z rolkami obróconymi o  $90^\circ$  w stosunku do osi głównej koła.



Rys. 3. Kinematyka robota typu Omnidirectional  
Fig. 3. Kinematics of the Omnidirectional robot

Takie podejście do projektu mechanicznego zapewniło dużą zwrotność robota, który z założenia przeznaczony jest do pracy w ograniczonych przestrzeniach (np. biurko, stół), przy zachowaniu prostoty wykonania i łatwości sterowania.

W celu opracowania właściwego sposobu poruszania się robota aplikacja nadrzędna rozwiązuje odwrotne zadanie kinematyki dla prędkości, opisane układem równań wektorowo-macierzowych [1]

$$\begin{bmatrix} \dot{\phi}_1 r \\ \dot{\phi}_2 r \\ \dot{\phi}_3 r \end{bmatrix} = \begin{bmatrix} \sin \alpha & -\cos \alpha & -d \\ 0 & 1 & -d \\ -\sin \alpha & \cos \alpha & -d \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (1)$$

gdzie:

- $\dot{\phi}_i$  oznacza prędkość obrotową  $i$ -tego koła, dla  $i=1, 2, 3$ ,
- $\dot{x}, \dot{y}, \dot{\theta}$  oznaczają odpowiednio prędkości liniowe i obrotową w nieruchomym układzie kartezjańskim  $X_I Y_I$ ,
- pozostałe parametry związane są z wymiarami konstrukcji:  $\alpha=60^\circ$ ,  $r=60$  mm,  $d=350$  mm.

Wyznaczone w ten sposób prędkości obrotowe kół są następnie przeskalowane i przekazywane do jednostki sterującej napędami. Prędkość kół jest ograniczona od górnio do 5 obr/s, ze względu na możliwość występowania poślizgów.

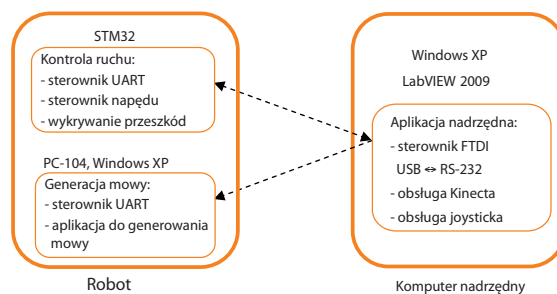
## Architektura sprzętowa

Architektura systemu bazuje na następujących elementach sprzętowych (rys. 4):

- system mikroprocesorowy przeznaczony do sterowania zespołem napędowym robota i detekcji przeszkód oraz komputer przemysłowy służący do generowania mowy, umieszczone na robocie,

- laptop stanowiący stację operatorską, na której zainstalowano aplikację nadzorującą pracę robota.

Ze względu na modułarną budowę robota konieczne było wybranie interfejsu komunikacyjnego, który umożliwi integrację wszystkich elementów (łatwą konfigurację i pracę na każdym z modułów sprzętowych) przy zapewnieniu odpowiedniej szybkości przesyłania krótkich komunikatów oraz bezprzewodowej pracy platformy. Wykorzystując dotychczasowe doświadczenie z zakresu interfejsów komunikacyjnych, zdecydowano się zastosować protokół bazujący na standardzie RS-232, zaś do realizacji warstwy fizycznej wybrano gotowe moduły radiowe MOBOT-RCR-USB-V2 pracujące w paśmie 868 MHz. Dzięki takiej konfiguracji oba moduły sprzętowe zainstalowane na robocie mogą odbierać jednocześnie informacje przesyłane z komputera nadrzędnego. W celu rozpoznania adresata informacji oraz zwiększenia odporności na błędy transmisji zastosowano ramki danych zawierające znacznik początku oraz końca.



Rys. 4. Architektura systemu

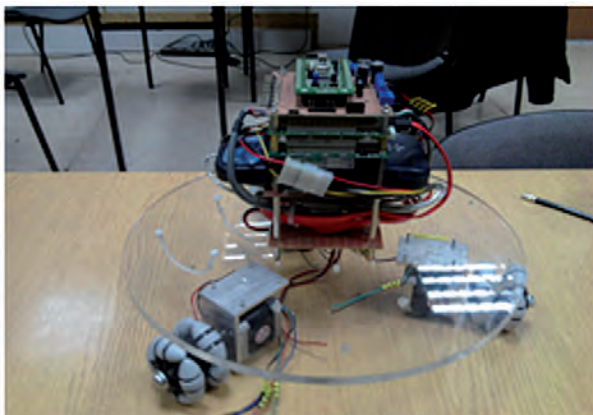
Fig. 4. Architecture of the system

Podczas pracy robota ważne jest zapewnienie bezpieczeństwa osobom i przedmiotom znajdującym się w jego przestrzeni roboczej. Dzięki zastosowaniu ośmiu czujników odległości SHARPGP2D120XJ00F rozmieszczonych na obwodzie platformy, możliwe jest wykrywanie i omijanie przeszkód, co przekłada się na prawidłowe rozpoznawanie przeszkód leżących wokół robota.

Każde z kół wyposażone jest w niezależny napęd zrealizowany na silniku krokowym. W ten sposób wyeliminowano konieczność stosowania sprzężenia zwrotnego do sterowania prędkością obrotową, co wymagałoby użycia stosunkowo drogich przetworników obrotowo-impulsowych. Przeprowadzone testy wykazały słuszność tego założenia – silniki krokowe wystarczająco dobrze odwzorowują zadany profil prędkości.

Do sterowania napędem służy dedykowany sterownik bipolarnych, dwufazowych silników krokowych. Na jednym obwodzie drukowanym zostały umieszczone trzy sterowniki pracujące synchronicznie, dzięki czemu pomiar prądu w obwodzie każdego z silników jest odporny na samozakłócenie się wynikające z komutacji tranzystorów. Sterownik został zaprojektowany w programie Altium Designer 2009.

Sterowaniem napędami, jak również detekcją przeszkód, zajmuje się mikroprocesor STM32F100RB wbudowany w zestaw uruchomieniowy STM32 DISCOVERY firmy STMicroelectronics [2]. Algorytm jego pracy został zapisany w obiektowym języku C++, co usprawniło tworzenie aplikacji. Aby zapewnić determinizm czasowy realizowanych zadań pętla programu głównego jest wykonywana

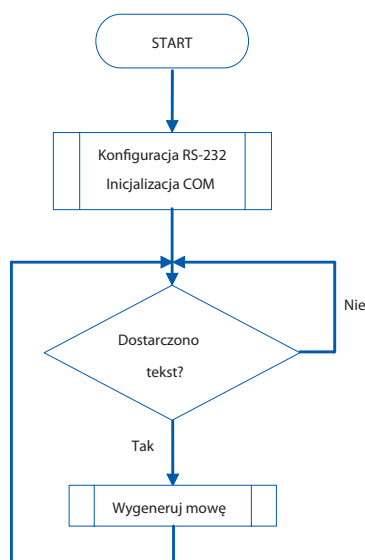


**Rys. 5.** Rozmieszczenie układów elektronicznych  
**Fig. 5.** Placement of the electronic components

w stałym, określonym czasie. Ponadto aplikacja wykorzystuje mechanizm przerwania pochodzących od urządzeń peryferyjnych. W programie zostały zaimplementowane takie moduły programowe, jak: generacja rampy, interfejs komunikacyjny RS-232, wykrywanie przeszkód, analiza stanu akumulatorów. Czas wykonywania pętli głównej jest krótszy od 1 ms, co zapewnia płynne odwzorowanie zadanego profilu prędkości oraz szybką reakcję na zmiany w otoczeniu robota.

## Generowanie mowy

Drugi moduł sterujący zamontowany na robocie stanowi komputer wykonany w standardzie przemysłowym PC-104, na którym zainstalowano system operacyjny Windows XP SP3. Wykorzystując wbudowane biblioteki API (ang. *Application Programming Interface*) oraz dodatkowe biblioteki SAPI (ang. *Speech Application Programming Interface*), na komputerze PC-104 stworzono aplikację do generowania mowy. Program interpretuje komendę (zawierającą tekst do wypowiedzenia) nadesłaną od aplikacji nadrzędnej, a następnie konfiguruje głos Mai dostarczony przez firmę IVONA Software.



**Rys. 6.** Schemat blokowy generatora mowy  
**Fig. 6.** Block diagram of the speech synthesizer

Generowanie mowy odbywa się w czterech głównych krokach:

- wstępna konfiguracja aplikacji,
- inicjalizacja Component Object Model (COM),
- ustawienia głosu,
- mowa.

W pierwszej części do programu dołączane są niezbędne biblioteki `altbase.h` oraz `altcom.h`, w drugiej natomiast inicjalizowany jest COM. Stworzona aplikacja nie wykorzystuje go bezpośrednio, lecz jak wiadomo SAPI jest aplikacją opartą na COM. W trzeciej części tworzony jest obiekt. Reprezentuje on głos, a konfigurowany jest przez podanie m.in. takich parametrów jak identyfikator głosu, który chcemy usłyszeć. Ostatnim krokiem dla tej części aplikacji jest wygenerowanie mowy. Wystarczy odwołać się do wcześniej stworzonego obiektu i podać odpowiednio sformatowany tekst do wygenerowania. Inną część aplikacji stanowi konfiguracja RS-232. Algorytm pracy tego modułu działa według schematu pokazanego na rys. 6.

W standardowej konfiguracji komputer przemysłowy nie ma karty dźwiękowej, w związku z czym zastosowano kartę zewnętrzną ESI MAYA 44 USB oraz głośniki zasilane z portu USB.

## Aplikacja nadrzędna

Sercem projektu jest aplikacja działająca na zewnętrznym laptopie i stworzona w środowisku graficznym LabVIEW [3]. Za wyborem tego środowiska przemawiała dostępność sterowników do wszystkich użytych układów peryferyjnych oraz łatwość tworzenia programu. Aplikacja nadrzędna realizuje główne zadanie, jakim jest pobieranie informacji na temat zachowania operatora, by następnie przesłać komendy sterowania ruchem oraz tekst wypowiedzi do platformy mobilnej. Poza możliwością ręcznego testowania robota z poziomu panelu czołowego LabVIEW, możliwe jest użycie joysticka 3DOF i czujnika Kinect. Ciekawym elementem składowym systemu jest, znany z gier wideo, czujnik Microsoft Kinect. Umożliwia on wykrycie człowieka oraz rozpoznawanie jego ruchów dzięki wbudowanej funkcji szkieletyzacji. Możliwa jest zatem dowolna interpretacja ruchów całego człowieka, jego kończyn lub dłoni i przełożenie ich na określone zachowanie robota mobilnego. Dodatkowo do sterowania użyto joystick 3DOF, który oprócz pracy w osiach X oraz Y umożliwia również zadawanie obrótu wokół osi Z, co pokrywa się w pełni z możliwościami kinematycznymi robota. Uzyskana w ten sposób informacja jest przeliczana, z wykorzystaniem równań wektorowo-macierzowych (1) na prędkości obrotowe poszczególnych kół. Wszystkie wymienione urządzenia mają gotowe sterowniki LabVIEW umożliwiające ich integrację z aplikacją. Do komunikacji bezprzewodowej został stworzony sterownik FTDI.

## Zasilanie

Ze względu na zróżnicowany pobór mocy przez opisane elementy składowe robota, dokonano dywersyfikacji źródeł zasilania. Takie podejście ogranicza również ewentualne zniszczenia wywołane zwarciami w jednym z modułów. W projekcie zastosowano akumulatory litowo-polimerowe, ze względu na ich stosunkowo niską masę. Silniki krokowe wykorzystują źródło o wysokiej wydajności prądowej i napięciu niestabil-

zowanym 11,1 V DC. Do zasilacza impulsowego dla modułu PC-104 zastosowano zasilanie 14 V DC. Najmniej wydajne prądowo jest zasilanie procesora STM32 o stabilizowanym napięciu 3,3 V DC. Dzięki temu cały układ może pracować do 5 godzin bez konieczności ponownego ładowania.

## Podsumowanie

Zastosowana architektura sprzętowa systemu zapewnia dużą skalowalność. Poszczególne elementy składowe mogą być dalej rozbudowane o kolejne moduły. Duże rezerwy mocy obliczeniowej oraz zasoby wejść/wyjść pozwalają na rozwój projektu w przyszłości, m.in. możliwe jest wzbogacenie robota o system wizyjny służący do lokalizowania i rozpoznawania twarzy, czy też aplikację do rozpoznawania i analizy mowy.

Szczegółowe informacje o projekcie można znaleźć na stronie internetowej poświęconej opisanej konstrukcji <http://omnivoice.prototypowanie.com.pl>.

## Bibliografia

1. Siegwart R., Nourbakhsh I. R., *Introduction to Autonomous Mobile Robots*, MIT 2004.
2. Paprocki K., *Mikrokontrolery STM32 w praktyce*, BTC, Legionowo 2009.
3. Chruściel M., *LabVIEW w praktyce*, BTC, Legionowo 2008.
4. Granosik G., Stanusch M., Wójtowicz K., *Robot społeczny TEPSON*, PAR 6/2010, 12–16.

### A mobile platform for testing human-robot interaction

**Abstract:** In this paper the autonomous mobile robot composed of various hardware and software modules is described. Project implements a few basic functions of human-machine interaction: voice generation, recognition of human body movement, obstacle avoidance, and wireless communication. Electronic components and associated software can be divided into three main modules: STM32 microcontroller controls movements of the robot, PC-104 computer located on the platform synthesizes voice, PC computer with LabVIEW application remotely controls behavior of the entire system. This approach, based on the scalability and modularity, opens an easy way for further development. The project was awarded in the regional and domestic robotics competitions.

**Keywords:** social robot, human-machine interaction, Omnidirectional kinematics, STM32, LabVIEW, PC-104

---

### inż. Michał Maciejewski

Student II stopnia kierunku Automatyka i Robotyka na Wydziale Elektrotechniki, Elektroniki, Informatyki i Automatyki Politechniki Łódzkiej. Członek Koła Naukowego Robotyki SKaNeR oraz prezes Studenckiego Koła Naukowego Inżynieria DSP, Koordynator projektu OmnIVOice. Zainteresowania: robotyka, programowanie mikrokontrolerów, LabVIEW, C/C++, systemy wbudowane, teoria sterowania.

e-mail: [maciejewskimichal@o2.pl](mailto:maciejewskimichal@o2.pl)



### inż. Marek Niemiec

Student II stopnia kierunku Automatyka i Robotyka na Wydziale Elektrotechniki, Elektroniki, Informatyki i Automatyki Politechniki Łódzkiej. Członek Studenckiego Koła Naukowego Robotyki SKaNeR oraz Studenckiego Koła Naukowego Inżynieria DSP. Zainteresowania: robotyka, automatyka przemysłowa, elektronika praktyczna, nowinki techniczne, muzyka, motoryzacja, sporty zimowe.

e-mail: [niemiec\\_marek@o2.pl](mailto:niemiec_marek@o2.pl)



### inż. Łukasz Ryszka

Student II stopnia kierunku Automatyka i Robotyka na Wydziale Elektrotechniki, Elektroniki, Informatyki i Automatyki Politechniki Łódzkiej. Członek Koła Naukowego Robotyki SKaNeR. Zainteresowania: robotyka, programowanie mikrokontrolerów, projektowanie elektroniki, napędy, systemy wbudowane, Linux, programowanie C#, C/C++, projektowanie sterowników do ekranów LCD/TFT z panelem dotykowym, teoria sterowania.

e-mail: [lukasz.ryszka@o2.pl](mailto:lukasz.ryszka@o2.pl)

