

dr inż. Michał Gnatowski  
Instytut Podstawowych Problemów Techniki PAN  
Kacper Kulczycki  
Wydział Fizyki Uniwersytetu Warszawskiego

**AUTONOMICZNY ROBOT STUDENCKI  
– WYKORZYSTANIE SYSTEMU LEGO MINDSTORMS NXT  
W KONSTRUKCJACH ROBOTÓW MOBILNYCH**

*W tekście została omówiona konstrukcja gąsienicowego robota mobilnego, stworzonego przez studentów, należących do Naukowego Koła Sztucznej Inteligencji przy Wydziale Fizyki Uniwersytetu Warszawskiego. W opisie, szczególny nacisk położono na możliwości i ograniczenia, wynikające z zastosowania gotowego systemu konstrukcyjnego firmy LEGO, oraz dostępnego dla tej platformy oprogramowania open source.*

**AUTONOMOUS STUDENTS' ROBOT  
– APPLICATION OF THE LEGO MINDSTORMS NXT SYSTEM IN  
CONSTRUCTIONS OF MOBILE ROBOTS**

*In this article a construction of a caterpillar mobile robot is presented. The robot was created by students, belonging to Scientific Group of Artificial Intelligence, by the Faculty of Physics of the University of the Warsaw. In the description, particular emphasis stayed for possibilities and restrictions, resulting from applying of the ready construction system of the LEGO company, and available open source software for this platform.*

## **1. WSTĘP**

W ostatnich latach, widoczny jest znaczny rozwój robotyki mobilnej. Roboty przemysłowe są stosowane od wielu lat. Obecnie jednak, upowszechniają się badania, dotyczące robotów usługowych, czyli robotów mobilnych poruszających się w otoczeniu człowieka. Przykładami takich robotów, mogą być roboty policyjne i wojskowe, roboty wykonujące zadania transportowe w budynkach użyteczności publicznej lub realizujące proste prace w otoczeniu, w którym przebywanie człowieka może być dla niego szkodliwe. Istnieją znane polskie konstrukcje robotów interwencyjno-inspekcyjnych, które wyszły już poza fazę badań laboratoryjnych i są eksploatowane przez służby policyjno-wojskowe [1]. Oprócz robotów sterowanych zdalnie przez operatora, na polskich uczelniach tworzy się autonomiczne roboty własnej konstrukcji. Przykładem mogą być roboty „Elektron” i „Warrior I” wykonane na Politechnice Warszawskiej [2, 18], robot do inspekcji trudnodostępnych obiektów technicznych wykonany na Politechnice Śląskiej [19], czy robot „Pathfinder” stworzony w Polsko-Japońskiej Wyższej Szkole Technik Komputerowych [3].

Wraz z rosnącym postępem elektroniki, a przede wszystkim miniaturyzacją, na rynku hobby-styczno-modelarskim zaczęły pojawiać się zestawy umożliwiające tworzenie miniaturowych robotów. Przykładami mogą być produkty takie jak VEX [4] czy Mindstorms [5]. Pierwszy z nich, jest niejako skomputeryzowanym „młodym technikiem”, drugi bazuje na klockach LEGO Technics. Gotowe elementy i moduły takich systemów pozwalają na budowę robotów, o różnym stopniu skomplikowania, w dość krótkim czasie. Ma to znaczenie nie tylko w przypadku zastosowań hobbyistycznych czy akademickich [6], ale również w przypadku szybkie-

go prototypowania. W ostatnim z wymienionych zastosowań, daje możliwość nie tyle zweryfikowania parametrów bryły robota, lecz głównie jego dynamiki.

Ze względu na wymienione cechy systemu LEGO Mindstorms NXT, członkowie Naukowego Koła Sztucznej Inteligencji [7] podjęli decyzję o budowie robota, w oparciu o to rozwiązanie. Podstawowym elementem produktu firmy LEGO jest centralny mikrokomputer, umożliwiający obsługę do trzech serwowmotorów (również dostarczane przez producenta), oraz czterech portów przeznaczonych do podłączenia czujników. Dane z sensorów mogą być przesyłane w postaci analogowej lub cyfrowej. W drugim przypadku, zgodnie ze standardem magistrali I<sup>2</sup>C (a w przypadku jednego - opatrzonego numerem 4 - portu, również zgodnego ze standardem RS-485). Opublikowanie przez firmę specyfikacji softwaru i hardware [8] dało możliwość innym producentom, oraz społeczności użytkowników, tworzenia alternatywy dla czujników i oprogramowania dostarczonych przez LEGO.

W artykule pokazano, że z dostępnych obecnie na rynku, gotowych elementów, możliwe jest proste i stosunkowo tanie wykonanie zaawansowanego robota. Parametry techniczne umożliwiają zrealizowanie procesu lokalizacji robota z wystarczającą dokładnością, co umożliwia wykorzystanie go, do późniejszych praktycznych zastosowań, w szczególności do dalszych prac naukowych.

## 2. KONSTRUKCJA

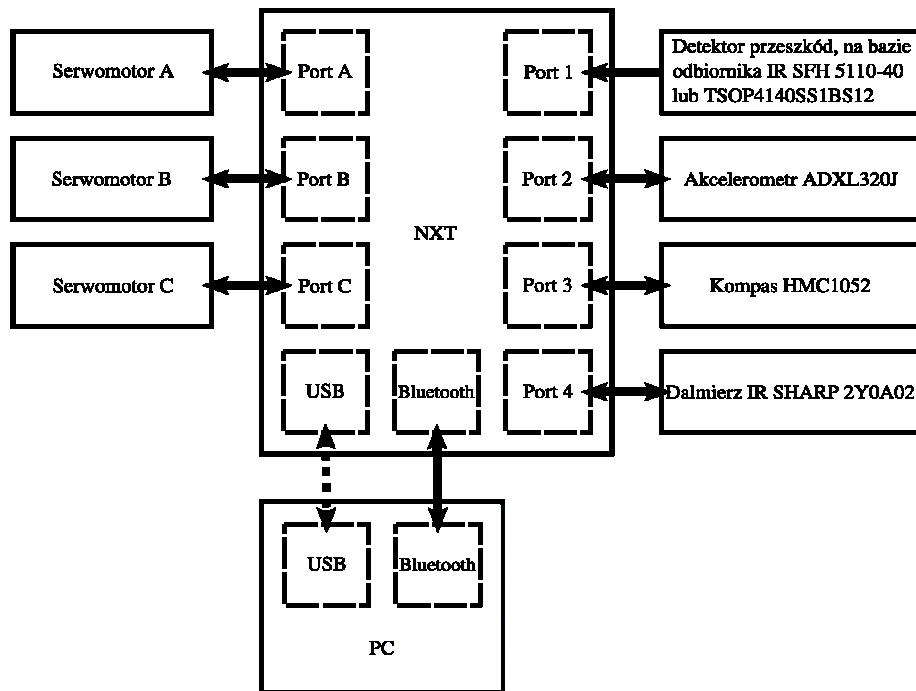
Sercem robota jest wspomniany we wstępie mikrokomputer, który oprócz portów do komunikacji z serwowmotorami i sensorami, posiada port USB oraz łącze bezprzewodowe Bluetooth klasy II (zasięg do 10m). Zapewnia to możliwość programowania i sterowania nim przez łącze kablowe i radiowe. Ze względu na dość prosty zestaw sensorów dostarczanych przez firmę LEGO (sonar, czujniki dotyku, mikrofon, czujnik światła), zapadła decyzja o zakupie dodatkowych czujników firmy mindsensors.com [9]. Sensory tej firmy są gotowymi modułami, w pełni kompatybilnymi z Mindstorms NXT. Schemat połączeń modułów robota, przedstawia rys. 1.

Mikrokomputer robota bazuje na procesorze ARM7 wspomaganym koprocesorem - mikrokontrolerem Atmega48. Dysponuje on również 256 kilobajtami pamięci (około połowa zajęta przez firmware). Zasilanie stanowi akumulator litowo-jonowy o pojemności 1400 mAh.

Bryła robota ma wymiary:

- długość 222 mm,
- szerokość 211 mm,
- wysokość 122 mm (do szczytu bryły), 155 mm (do szczytu dalmierza),
- szerokość gąsienic 20,5 mm,
- długość pola kontaktu gąsienicy z podłożem 112 mm,
- rozstaw gąsienic 190,5 mm,
- prześwit między dołem korpusu a podłożem 19 mm,
- promień obwiedni zewnętrznej powierzchni gąsienicy, zawiniętej na kole napędowym 22,5 mm.

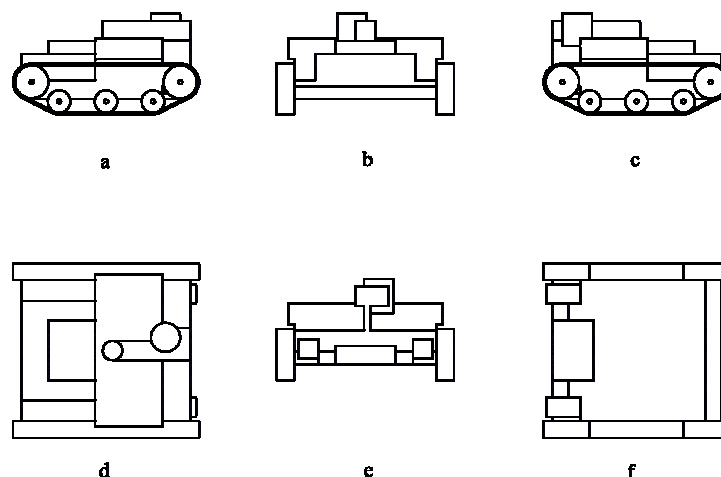
Masa robota 1057 g.



Rys.1. Schemat połączeń sensorów i silników z mikrokomputerem robota, oraz komunikacji ze stacją bazową – komputerem PC. (Złącze USB wykorzystywane tylko w celach diagnostycznych – przerywana linia)

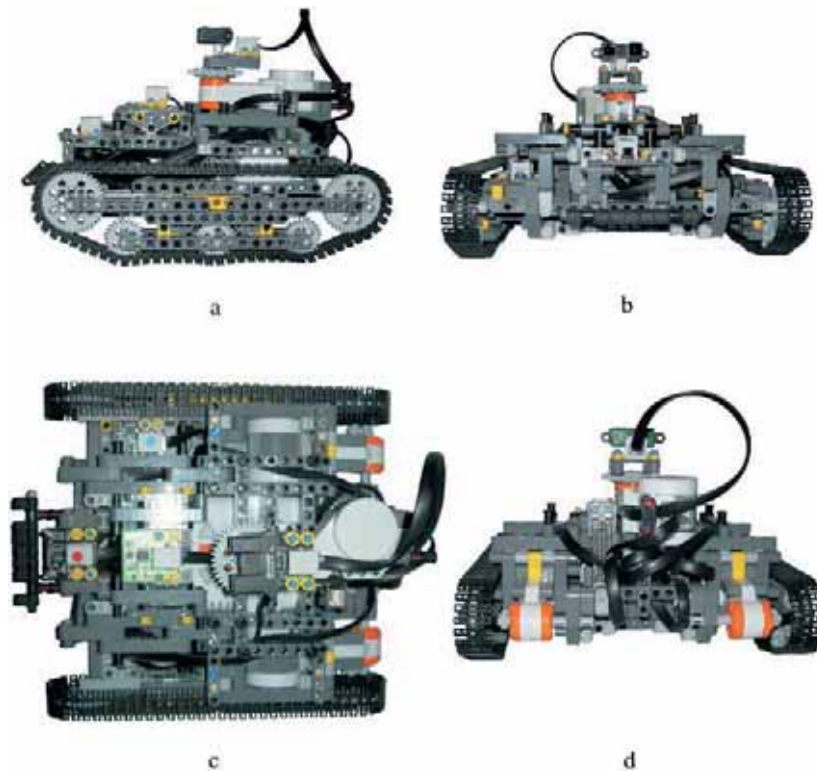
Sensory jak i serwomotory, są podłączone do mikrokomputera dedykowanymi kablami i gniazdami. Są one niejako lustrzanym odbiciem złączy RJ-12 systemu DEC. Kluczyk wtyczki znajduje się w tych systemach po przeciwnych stronach. Zapobiega to podłączeniu elementów systemu Mindstorms NXT do linii telefonicznej itp. przez młodszych użytkowników tego systemu. Jednak dla osób chcących stworzyć własny czujnik, to rozwiązanie jest nieco kłopotliwe. Na szczęście, do konstrukcji ARS, zastosowane zostały gotowe czujniki przeznaczone dla NXT, co pozwoliło ominąć taki problem.

Schematyczny wygląd bryły robota ilustruje rys. 2.



Rys. 2. Rzuty bryły robota (bez uwzględnienia montażu czujników): a – lewa strona, b – przód, c – prawa strona, d – widok z góry, e – tył, f – widok od spodu

Rzeczywisty wygląd robota przedstawia rys. 3. Rozmieszczenie poszczególnych sensorów w robocie, zostało natomiast pokazane na rys. 4.



Rys. 3. Zdjęcia robota (na przodzie robota dołożony zderzak, z tyłu widoczna przejściówka umożliwiająca dostęp do portu USB, który ze względu na zwartą budowę robota został „zabudowany”): a – lewa strona, b – przód, c – widok z góry, d – tył.



Rys. 4. Widok ogólny robota, z zaznaczeniem poszczególnych sensorów: a – kompas cyfrowy, b – akcelerometr (pomiar w dwóch wymiarach), c – detektor przeszkód, d – dalmierz, umieszczony na osi silnika

W pierwotnym założeniu, robot miał być dwukołowym urządzeniem o napędzie różnicowym, z rolką toczną, co zmniejszało mechaniczną komplikację konstrukcji [10]. Robot miał być wzorowany, na takich konstrukcjach jak „Pioneer DX”. Ostateczne, gąsienicowe rozwiązanie zostało zastosowane, ze względu na duży problem z wykonaniem rolki tocznej. Dostępne w ramach systemu Mindstorms komponenty nie pozwoliły na wykonanie takiego elementu, o satysfakcjonujących własnościach. Rolka wykonana z plastikowej kuli, miała tendencje do blokowania się. Dodatkowo, pojawił się problem z uzyskaniem odpowiedniej sztywności ramy nośnej korpusu. Zastosowanie napędu gąsienicowego, co prawda spowodowało większą komplikację w przypadku sterowania (a przede wszystkim lokalizacji), jednak pozwoliło na stworzenie bardziej zwartej, a przez to sztywniejszej konstrukcji, oraz wyeliminowało kluczowy problem – rolkę toczną. Problem opisu ruchu pojazdów gąsienicowych, jest na szczęście dobrze opisany w literaturze [11], a w przypadku zastosowanej szerokości gąsienic, długości pola ich kontaktu z podłożem, i rozstawu, można w pierwszym przybliżeniu stosować model ruchu pojazdu kołowego sterowanego różnicowo (mającego poślizg podczas wykonywania skrętu).

Taka konstrukcja umożliwia również, efektywne wykorzystanie wszystkich czujników. W szczególności, usytuowanie dalmierza na osi silnika pozwala na skanowanie otoczenia wokół robota. Obrót tego czujnika jest możliwy, o kąt  $\pm 135$  stopni (zero na osi przód-tył). W jednym skanie jest 270 punktów – tachometry serwomotorów mają rozdzielczość jednego stopnia kąтового. A pomiar odległości następuje w zakresie 200-1500mm z dokładnością jednego milimetra. Odchylenie robota od kierunku północnego (liczone zgodnie z ruchem wskazówek zegara) podawane jest przez kompas, z dokładnością do ok. 0,1 stopnia. Akcelerometr jest w stanie rejestrować zakres przyspieszeń  $\pm 5G$ , z dokładnością do ok.  $0,01 \text{ m/s}^2$  (wynik podawany w całkowitych wielokrotnościach  $0,00918 \text{ m/s}^2$ ). Detektor przeszkód jest w stanie zarejestrować obiekt w odległości około 200 mm, zakres kątowy jego działania to 120 stopni, a selektywność kierunków ma rozpiętość 45 stopni dla sektora lewego i prawego, i 30 stopni dla środkowego. Dodatkowo dzięki tachometrom serwomotorów, możliwe jest rejestrowanie odometrycznych przesunięć z dokładnością do około 5mm (jest to prawdą tylko w przypadku jazdy „na wprost” - skręty są wysoce nieholonomiczne).

Porty czujników, jak było wspomniane, mogą obsługiwać zarówno czujniki analogowe jak i cyfrowe. Jest więc możliwość podłączania jednocześnie dwóch takich czujników do jednego portu, i ich współdziałanie (nie równoczesne, ale wymienne). Czujniki cyfrowe mogą być stosowane w większej ilości, szeregowo podłączone do jednego portu (dzięki kompatybilności z magistralą I<sup>2</sup>C). Oczywiście w takiej sytuacji może okazać się, że większa ilość czujników będzie wymagała większego zapotrzebowania prądowego, a co za tym idzie dodatkowego zasilania. Nieco gorsza sytuacja dotyczy możliwości podłączenia dodatkowych silników. Są one sterowane sygnałem PWM. W związku z tym, każdy port przeznaczony jest tylko dla jednego serwomotoru. Powstały jednak rozwiązania pozwalające na dodanie dodatkowych serw i kontrolowanie ich przez porty sensorów (rozwiązanie takie oferuje wspomniany już producent sensorów użytych w konstrukcji robota).

Minusem stosowania LEGO Mindstorms jest konieczność stosowania do budowy, elementów o skwantowanych wymiarach (7,5x7,5x7,9 mm, lub 7,9x7,9x9,5 mm). Jednak ten sam zarzut można postawić również systemom takim jak VEX – gdzie konstrukcja jest ograniczana przez rozstaw otworów montażowych w metalowych elementach.

W przypadku tworzenia własnego czujnika, kłopotliwe mogą być również, opisane wcześniej, zastosowanie przez producenta złącza portów. Na szczęście w firmie mindsensors.com można

nabyć potrzebne wtyki i gniazda, a także narzędzie do zaciskania tych wtyków na kablach. Firma ta oferuje również płytki prototypowe, przeznaczone do konstruowania własnych czujników, lub innych komponentów systemu.

Należy w tym miejscu jeszcze raz podkreślić, że dostarczane przez LEGO elementy, wykonane z tworzywa sztucznego, pozwoliły na stworzenie sztywnej ramy konstrukcyjnej robota. Jest to o tyle ciekawe, że ich łączenie realizowane jest, przez czopy zatrzaskiwane w otworach, a nie śruby czy nity.

### 3. OPROGRAMOWANIE

Podstawowym elementem oprogramowania tego systemu jest firmware dostarczony przez LEGO i zainstalowany na mikrokomputerze Mindstorms NXT. Zawiera on obsługę portów, zaimplementowane algorytmy takie jak PID, czy choćby funkcje informujące o poziomie naładowania baterii, czy ilości i kierunku wykonanych kroków przez serwomotor. Dużym jednak jego mankamentem jest brak implementacji zmiennych zmiennoprzecinkowych. Dla równowagi zawiera on jednak obsługę wielowątkowości. Oczywiście dzięki otwarciu kodu firmwaru powstało wiele alternatywnych rozwiązań [12]. W projekcie ARS, zostało jednak użyte oryginalne oprogramowanie dostarczone przez producenta.

Kolejny element oprogramowania to kompilator zdolny do tworzenia binarnych plików wykonywalnych, zgodnych z wewnętrznym oprogramowaniem Mindstorms NXT. Oczywiście producent dostarcza własne środowisko, jest jednak ono bardziej skierowane na łatwość obsługi przez podstawowego odbiorcę – a więc młodzież, niż wysoką użyteczność programistyczną. Po analizie dostępnego oprogramowania wybór padł na środowisko Next Byte Codes i Not eXactly C [13]. NBC jest rodzajem asemblera, a NXC językiem wysokiego poziomu, mającym składnię naśladowującą język C. Wygodna składnia NXC, wysoka kompatybilność z funkcjami dostarczonymi przez firmware, wsparcie dla czujników firmy mindsensors.com, zaważyły na tym wyborze.

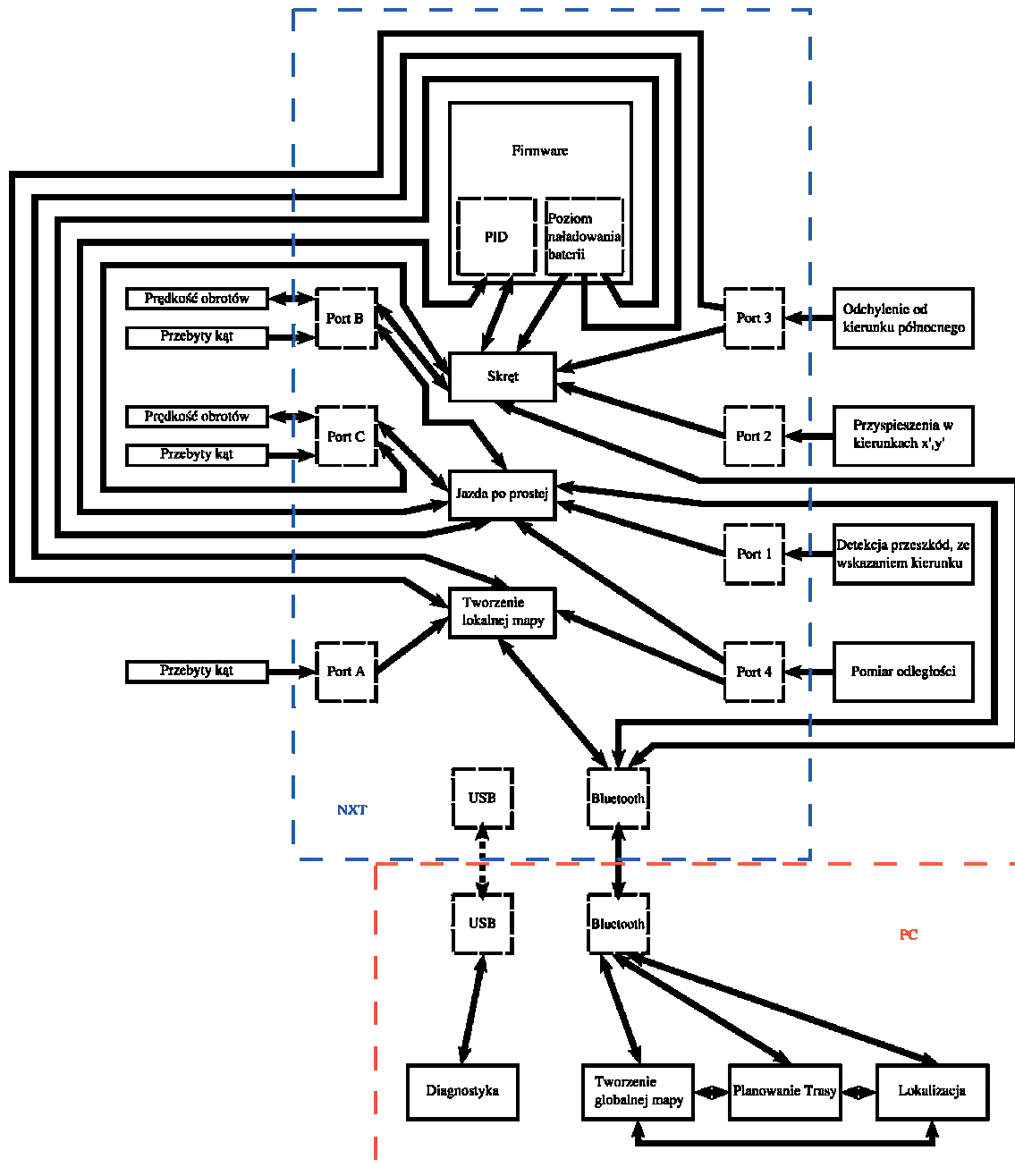
Pozostał jednak, problem braku obsługi zmiennych zmiennoprzecinkowych. Rozwiązaniem tego problemu, oraz metodą na poradzenie sobie z dość ograniczonym zasobem pamięci, dostępnej w mikrokomputerze NXT, było przeniesienie części obliczeń do stacji bazowej stanowiącej komputer PC pracujący pod kontrolą systemu Linux (dystrybucja Gentoo [14]). W celu umożliwienia komunikacji robota z bazą, wykorzystane zostało łącze Bluetooth. Do obsługi i zdalnego uruchamiania programów wewnątrz Mindstorms NXT, posłużyła biblioteka nxtlibc [15] wykorzystywana przez program sterujący, pracujący wewnątrz komputera-bazy.

Dodatkowo w celach diagnostycznych, a dokładnie w celu przesyłania skompilowanego oprogramowania napisanego w NXC, na mikrokomputer NXT, użyty został program linxt [16] (umożliwiający wygodną obsługę połączenia bazy z robotem, za pośrednictwem portu USB).

Ze względu na ograniczenia wymienione wcześniej, oprogramowanie które zostało napisane w celu uruchomienia i pracy w mikrokomputerze NXT, jest ograniczone do minimum. Składa się ono z trzech podstawowych modułów realizujących jazdę na wprost, skręt (obrót w miejscu, z przeciwbieżnie pracującymi silnikami), oraz skanowanie otoczenia w celu budowy lokalnej mapy (procedura odbywa się podczas postoju). Programy te mają jedynie za zadanie, zebranie odpowiednich danych i przesłanie ich do bazy, lub poinformowanie o błędzie podczas ich realizacji.

Podobny podział można było wprowadzić w oprogramowaniu pracującym w bazie – zasadniczo da się je podzielić również na trzy części – pierwszą, odpowiedzialną za budowanie mapy otoczenia, drugą za lokalizację i trzecią, za planowanie trasy.

Ostateczny schemat oprogramowania, wraz z danymi, które są zbierane podczas pracy robota przedstawia rys. 5.

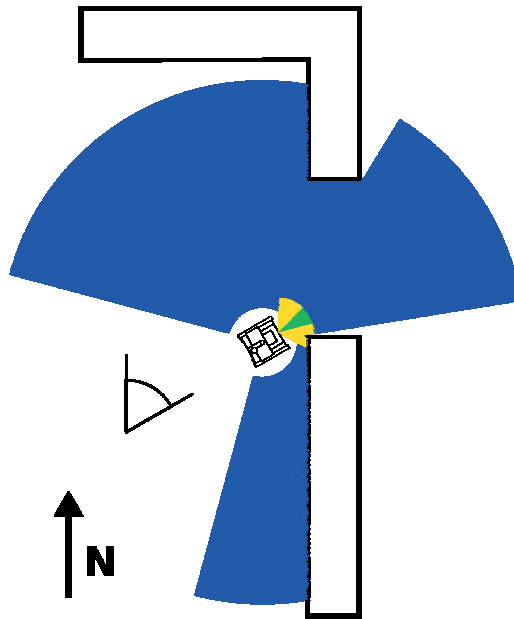


Rys. 5. Schemat oprogramowania, pracującego w bazie i robocie. (Zasadniczo wszystkie informacje wymieniane między modułami jazdy na wprost, skretu i budowania lokalnej mapy – skanowania otoczenia oraz portami, powinny przechodzić przez blok firmwaru, jednak zaciemniłoby to i tak dość skomplikowany graf)

#### 4. STEROWANIE

Skanowanie otoczenia daje dość duży zakres lokalnej mapy. Korelując ją ze wskazaniem kompasu, otrzymuje się kolejne elementy mapy globalnej (gdy robot znajduje się w położeniu dotychczas nie odwiedzanym), lub dodatkową informację wykorzystywaną przy lokalizacji. Jednocześnie widać, że dzięki temu, podczas jazdy „na wprost” możliwe jest znalezienie kompromisu między szybkością jazdy, a dokładnością skanowania otoczenia. Co więcej, możliwe jest zastosowanie podczas jazdy, poprawek w określaniu aktualnej pozycji, poprzez uwzględnianie odczytów z dalmierza (bez przechodzenia w fazę skanowania).

Przykładową sytuację podczas pracy robota przedstawia rys. 6.



Rys. 6. Zakres działania sensorów (dalmierza i detektora przeszkód) podczas typowej sytuacji dojazdu robota do otworu drzwiowego (na rysunku zachowano skalę wielkości korpusu robota i zasięgu sensorów).

Detektor przeszkód pełni rolę pomocniczą, ze względu na umieszczenie dalmierza na szczycie robota. Niestety, ze względu na rodzaj stosowanego napędu (gąsienicowy, sterowany różnicowo), skanowanie otoczenia konieczne jest przed wykonaniem i po wykonaniu skrętu. Jak było poprzednio wskazane, rozdzielczość informacji o orientacji robota względem kierunku północnego, jest o rząd wielkości lepsza niż kątowa rozdzielczość skanów otoczenia. Zapewnia to dokładne i łatwe wyznaczenie skierowania robota (oś  $y$ , globalnego układu współrzędnych pokrywa się z kierunkiem północnym, a punkt 0,0 z punktem początkowego położenia robota). Jednak dopiero skorelowanie dwóch skanów otoczenia, oraz zebranych podczas wykonywania skrętu zmian przyspieszeń, (a stąd znajomości przybliżonego toru ruchu robota), daje możliwość precyzyjnego wyznaczenia położenia końcowego.

#### 5. MOŻLIWOŚCI DALSZEGO ROZWÓJU PROJEKTU

Obecna konstrukcja ma być wyjściowym elementem w przygotowywanej pracy magisterskiej, polegającej na modelowaniu ruchu tego robota w środowisku Player/Stage [17]. Przy tej okazji mają zostać wykorzystane istniejące już sterowniki zaimplementowane w tym śro-



dowisku. Jednak w obecnej chwili widać już, że istnieje konieczność stworzenia sterownika opierającego się na bibliotece nxtlibc. W przeciwieństwie jednak do typowego zastosowania Player/Stage, jak np. w przypadku robota „Elektron” [2], środowisko to będzie uruchomione w stacji bazowej, a nie wewnątrz samego robota, zastępując obecny program sterujący.

Otwartość specyfikacji sprzętowej Mindstorms NXT, powoduje powstawanie nowych komponentów współpracujących z tym systemem, produkowanych przez firmy niezwiązane bezpośrednio z LEGO. Istnieje więc możliwość dodawania kolejnych, nowych czujników do tego robota. W szczególności zastosowania nawigacji opartej na systemie wizyjnym (mindsensors.com oferuje kamery współpracujące z mikrokomputerem NXT).

Jako że projekt powstał na Uniwersytecie Warszawskim, doświadczenia zebrane podczas jego realizacji, wykorzystane zostaną przy tworzeniu propozycji programu zajęć z podstaw robotyki, dla studentów UW.

Dodatkowo, ciekawą funkcją zaimplementowaną w przez LEGO, jest możliwość komunikacji między mikrokomputerami NXT za pomocą łącza Bluetooth. Daje to możliwość, zastosowania kilku takich konstrukcji, w problemie współpracy grupy robotów. Zgodnie ze specyfikacją opublikowaną przez firmę LEGO [8], jeden mikrokomputer NXT, może bezprzewodowo kontrolować trzy inne.

Najbardziej odległą propozycją rozwinięcia ARS, jest stworzenie dedykowanego firmwaru, tak aby określenie autonomiczny w nazwie projektu, było w pełni zgodne z zastosowanym rozwiązaniem.

## 6. WNIOSKI

Realizacja projektu wykazała, że mimo skierowania zestawów LEGO Mindstorms NXT głównie do młodzieży, oraz ich rzeczywistych ograniczeń, systemy takie mogą stać się podstawą do stworzenia całkiem poważnych konstrukcji. Opublikowanie specyfikacji sprzętu jak i oprogramowania, daje duże możliwości rozbudowy mechanicznej i elektornicznej systemów bazujących na Mindstorms NXT, jak i tworzenia oprogramowania w pełni wykorzystującego możliwości mikrokomputera NXT. Niewątpliwą wadą rozwiązań bazujących na gotowych modułach, jest konieczność pozostawiania w zgodzie z ograniczeniami wymiarów stosowanych elementów, standardami złącz, itp. Z drugiej zaś strony, modularność pozwala na znacznie szybsze stworzenie konstrukcji i zweryfikowania przy jej pomocy problemów, stawianych np. prototypowi. Z doświadczeń wyniesionych podczas realizacji projektu wynika, iż od strony sprzętowej jedyną znaczącą wadą Mindstorms NXT może być ograniczona ilość dostępnej pamięci mikrokomputera. Natomiast w oprogramowaniu dostarczonym przez LEGO, najbardziej problematyczny stał się brak zmiennych zmiennoprzecinkowych. Oprócz tych dwóch zastrzeżeń, które udało się pokonać wykorzystując transmisję Bluetooth, Mindstorms NXT wydaje się być niezwykle wygodnym systemem bazowym, do tworzenia niewielkich konstrukcji robotów.

## BIBLIOGRAFIA

- [1] M. Trojnacki, P. Szykarczyk, A. Andrzejuk. *Tendencje rozwoju mobilnych robotów lądowych (1)*, Pomiary Automatyka Robotyka 6/2008, 11-14.
- [2] R. Chojecki, M. Olszewski, W. Szykiewicz, P. Trojanek. *Laboratoryjny robot mobilny Elektron i jego aplikacje*, Automation 2007, 1-10.

- [3] Ł. Jakóbiec, M. Żelaźewski, *Konstrukcja robota holonomicznego Pathfinder*, Automation 2008.
- [4] Strona internetowa VEX Robotics Design System: <http://www.vexrobotics.com/>
- [5] Strona internetowa LEGO Mindstorms: <http://mindstorms.lego.com/>
- [6] Strona internetowa zajęć z robotyki, prowadzonych na Politechnice Warszawskiej: <http://home.elka.pw.edu.pl/~mbrudka/ERO/>
- [7] Strona internetowa Naukowego Koła Sztucznej Inteligencji przy Wydziale Fizyki Uniwersytetu Warszawskiego: <http://nksi.igf.fuw.edu.pl/>
- [8] Strona internetowa z dokumentacją sprzętu i oprogramowania użytego w systemie Mindstorms NXT: <http://mindstorms.lego.com/Overview/nxtreme.aspx>
- [9] Strona internetowa firmy mindsensors.com: <http://www.mindsensors.com/>
- [10] T. Bräunl. *Embedded Robotics*, Springer 2003.
- [11] A. W. Chodkowski. *Konstrukcja i obliczanie szybkojeżdżących pojazdów gąsienicowych*, WKiŁ 1990.
- [12] Porównanie części dostępnego oprogramowania, współpracującego z Mindstorms NXT: <http://www.teamhassenplug.org/NXT/NXTSoftware.html>
- [13] Strona internetowa projektu kompilatora i języków Next Byte Codes i Not eXactly C: <http://bricxcc.sourceforge.net/nbc/>
- [14] Strona internetowa projektu Gentoo Linux: <http://www.gentoo.org/>
- [15] Strona internetowa projektu nxlbc: <http://www.quietearth.us/nxlbc.htm>
- [16] Strona internetowa projektu linxt <http://sourceforge.net/projects/linxt/>
- [17] Strona internetowa projektu Player/Stage: <http://playerstage.sourceforge.net/>
- [18] R. Chojecki, M. Olszewski, T. Pietrzak, P. Fryc, M. Wałęcki. *Budowa inspekcyjnego robot mobilnego WARRIOR I*, Krajowa Konferencja Robotyki, KKR 2008, tom I, 153-160.
- [19] M. Januszka, M. Adamczyk, W. Moczulski. *Nieholonomiczny autonomiczny robot mobilny do inspekcji obiektów technicznych*, Krajowa Konferencja Robotyki, KKR 2008, tom I, 143-152.