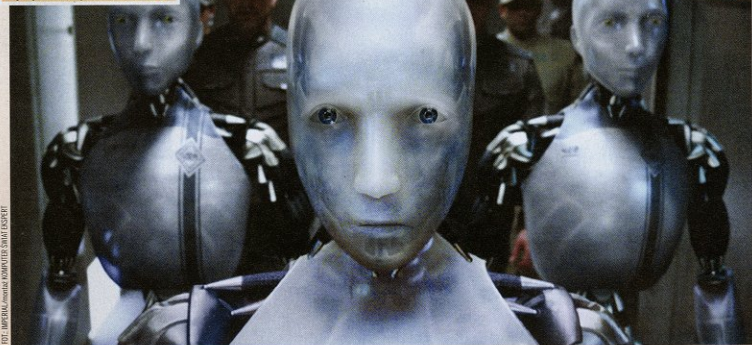


Indeks

Budowa robota	13
Przepisy minisumo	13
Moduły robota	14
Co w kołach piszczy	14
Moc z nami	14
Grunt to czujniki	14
Nazywam się kontroler. Mikrokontroler	15
Pomyśli, zanim pojedziesz	15
Co jest potrzebne, aby zbudować robota	15



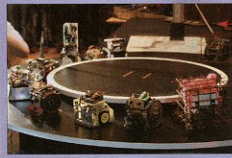
FOT. IMPIERIAL, JASNA, KOMPUTER, SMIK, EXPERT

Ja mam Robot!

Marzenie o posiadaniu myślącej maszyny towarzyszy nam od dzieciństwa. Czas zrealizować tę wizję i skonstruować własnego robota. Z Ekspertem teraz to możliwe

Zawody robotów minisumo

Od pewnego czasu na świecie organizowane są turnieje robotów sumo. Na podobieństwo walk sumo roboty podejmują w nich próby wypchnięcia przeciwnika z okrągłej maty, na której prowadzone są pojedynki. W Polsce takie zawody odbywają się w dwóch klasach: standardowej (organizowane na Politechnice Poznańskiej) i minisumo (na Politechnice Wrocławskiej i ostatnio także Gdańskiej). Zadanie robota biorącego udział w zawodach sumo polega na zlokalizowaniu przeciwnika, zaatakowaniu go i w końcu wypchnięciu poza matę, przy równoczesnym zadbaniu o to, aby samemu nie znaleźć się poza nią.



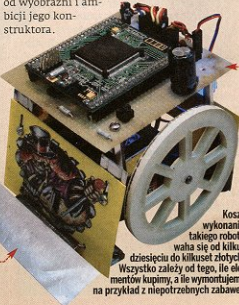
Dzisiaj prawie każdy z nas ma komputer osobisty. Zapewne też każdy słyszał o robotach, nie tylko o tych z filmów fantastycznonaukowych, lecz i o takich, które na liniach produkcyjnych montują samochody, wspomagają działania policji, czy też zostały skonstruowane w celu zapewnienia nam rozrywki. Czy znajdzie się ktoś, kto nie marzył o posiadaniu własnego robota?

Ekspert rozpoczyna cykl artykułów, z którego poznamy budowę robotów i dowiemy się, w jaki sposób wykonać własnego robota mobilnego gotowego do wzięcia udziału w zawodach robotów minisumo (patrz ramka Zawody robotów minisumo). W artykule przeczytamy, czym jest robot mobilny i z jakich elementów się składa. Ekspert przedstawi również podstawowe zasady obowiązujące w lidze robotów minisumo i pokaże różne rozwiązania konstrukcyjne stosowane w tej klasie robotów.

W kolejnych artykułach poznamy konstrukcję różnych modułów przykładowego robota.

Oczywiście konstruowany razem z Ekspertem robot nie musi być przeznaczony wyłącz-

nie do walk minisumo. Można mu wymyślić wiele innych zadań – wszystko zależy tylko od wyobraźni i ambicji jego konstruktora.

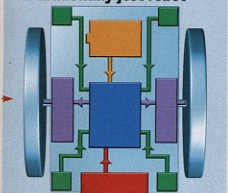


Koszt wykonania takiego robota waha się od kilkudziesięciu do kilkuset złotych. Wszystko zależy od tego, ile elementów kupimy, a ile wmontujemy na przykład z niepotrzebnych zabawek

Budowa robota

AbymoziliwszystkimCzytelnikomeksperymentowaniedurazbudowywłasnego robota minisumo, cykl artykułów o tworzeniu własnego robota Ekspert rozpoczyna od przedstawienia ogólnej charakterystyki modułów typowego robota mobilnego. Z tego artykułu dowiemy się również, jakie własności modułów robota minisumo są najbardziej pożądane i jakie rozwiązania techniczne są najczęściej stosuje.

Jak zbudowany jest robot



- czujniki białej linii
- czujnik wykrywający przeciwnika, razem z czujnikami białej linii tworzą system sensoryczny robota
- system zasilania (bateria)
- system napędowy
- sterownik

Czym jest robot

Pojęcie robot określa się automatycznie urządzenie wykonujące powierzone mu zadania, zazwyczaj z wykorzystaniem informacji odbieranych z otoczenia. Roboty można podzielić na stacjonarne, czyli nieprzemieszczające się, i mobilne. Typowy robot mobilny składa się z układu jezdnego, układu zasilającego, zestawu czujników i właściwie dobrego sterownika.

Roboty mogą być zdalnie sterowane lub działać autonomicznie, to znaczy samodzielnie wykonywać powierzone im zadania bez potrzeby ingerencji z zewnątrz. Robot autonomiczny zwykle postępuje zgodnie z zasadą „patrz, myśl, działaj”.

Na kołach, na nogach, byle z głową

Praktycznie każdy robot mobilny składa się z części mechanicznej i części elektronicznej (czyli z ciała i mózgu). Ciało stanowi o fizycznych możliwościach robota, mózg zaś determinuje jego zachowanie. Zadaniem konstruktora jest umiejętne połączenie tych elementów w jedną, spójną całość. Przede wszystkim należy przemyśleć sposób rozmieszczenia poszczególnych komponentów robota, co pozwoli na określenie postaci jego konstrukcji nośnej. Zazwyczaj rolę elementu nośnego pełni specjalnie przygotowana do tego celu rama z odpowiednio zaprojektowanymi interfejsami mechanicznymi. W wypadku prostych robotów jako elementu nośnego można użyć płytki drukowanej, na której za-

montowana będzie również elektronika robota.

W celu zapewnienia robotowi możliwości przemieszczania się, należy wyposażyć go w układ lokomocyjny. Konstrukcje tej części robota można rozwiązać na wiele sposobów. Zazwyczaj stosuje się kołowe lub gąsienicowe układy jezdne (rzadziej układy kroczące), w których źródłem siły napędowej są najczęściej silniki elektryczne. Do przeniesienia napędu na koła prawie zawsze wymagane jest zastosowanie właściwie dobranych przekładni mechanicznych.

Budując robota trzeba także pomyśleć o źródle energii. W chwili obecnej praktycznie jedynym rozwiązaniem są różnego rodzaju akumulatory.

Abymógł efektywnie działać w otoczeniu, potrzebuje informacji – z tego powodu jego konstruktor powinien pomyśleć o zamontowaniu na pokładzie różnego rodzaju czujników. Czujniki te mogą nie tylko informować robota o tym, co się dzieje wokół niego, ale także dostarczać mu danych o jego własnym stanie, na przykład o prędkości ruchu czy stopniu naładowania baterii.

Mózg jest najważniejszy

Wiemy już, z jakich elementów powinno składać się ciało robota. Najwyższy czas zastanowić się nad jego mózgiem, czyli układem sterowania. Musi to być prawdziwie centrum dowodzenia robota, w którym podejmowane będą wszystkie decyzje.

Budowę układu sterowania konstruktor powinien rozpocząć od zaprojektowania obwodów elektronicznych. To one pozwolą na implementację odpowiednich algorytmów, stanowiących o zachowaniu robota. Od przyjętego rozwiązania będzie zależał sposób realizacji zadania sterowania. Gdy w układzie sterowania zostanie wykorzystany mikroprocesor (na przykład Atmega48 firmy Atmel), algorytm przyjmie postać zwykłego programu komputerowego. Jeśli zaś zostanie użyte programowalne układy logiczne FPGA lub CPLD, to algorytm będzie zasztyt bezpośrednio w strukturze hardware'u.



Jeśli nie chcemy budować robota od podstaw, możemy kupić gotową, ale dość kosztowną, platformę robota mobilnego

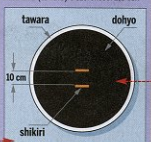
Wielki mały wojownik

Skoronaszym zadaniem jest skonstruowanie robota minisumo, trzeba się zastanowić, czym szczególnie powinien charakteryzować się robot mobilny, abymógł pretendować do miana mechanicznego wojownika klasy minisumo. Nasz robot musi być waleczny, groźny i co najważniejsze skuteczny w działaniu. Musi także spełniać wszystkie przepisy ligi robotów minisumo.

Spełnienie wymagań stawianych małym wojownikom nie jest łatwe. Klówna trudność polega na upakowaniu wszystkich elementów robota w niewielkiej przestrzeni i zachowaniu przy tym odpowiedniej wagi. To stanowi dopiero podstawę do dopuszczenia zawodnika do walki, ale pewnie nie wystarczy, aby wygrać zawody. W tym celu należy zadbać, aby robot był odpowiednio silny, szybki, zwrotny, a także czujny, sprężawczy i dobrze poinformowany. Oczywiście robot, aby zwyciężyć, powinien postępować odważnie, ale jednocześnie rozsądnie i efektywnie.

Przepisy minisumo

- Robot minisumo musi spełniać następujące kryteria:
 - robot musi być autonomiczny,
 - robot powinien mieścić się w pudełku o wymiarach 10 na 10 centymetrów, a jego waga nie może przekraczać 500 gramów,
 - po sygnale do walki, robot musi odczekać co najmniej 5 sekund w bezruchu,
 - robot nie może zawierać urządzeń, które mogą zakłócać działanie układu sterowania jego przeciwnika (na przykład urządzeń blokujących ruch przeciwnika czy błyskających świateł),
 - robot nie może zawierać części, które mogą uszkodzić powierzchnię, po której się porusza, ani urządzeń emitujących gazy, cieple, materiały sypkie,
 - niedozwolone jest emitowanie przez robota znacznych ilości ciepła,
 - robot nie może być przylutowany do podłoża.
- W poprzednim biurze udział dwa roboty, które walczy wewnątrz okrągłego ringu zwanego dohyo (czytaj dohyo), o średnicy 77 centymetrów. Górna powierzchnia dohyo jest płaska i gładka, a ma czarny kolor. Na brzegu dohyo namalowana jest biała linia (tawara) o szerokości 2,5 centymetra.



Na początku rozgrywki zawodnicy ustawiają się naprzeciwko siebie (na polach shikiri) i walczą do chwili, gdy jeden z nich zostanie wypchnięty poza matę. Pełny regulamin zawodów robotów minisumo znajdziemy na stronie 1.

Nano i femtosumo

Konstruuje się także roboty mniejsze od minisumo – na przykład nanosumo (muszą się zmieścić w sześcieniu o boku 2,5 centymetra) czy femtosumo (sześcienu o boku 1 cm). Autorytety w dziedzinie Polscy (jego konstruktorem jest Karol Sydor, współautor tego artykułu). Właściwie wciąż oczekuje na godnego swojego robota rywala.



Moduły robota

Znany już ogólną budowę robota. Czas poznać zasady konstrukcji jego podstawowych komponentów

CO W KOŁACH PISZCZY

Przyjmujemy, że nasz robot będzie poruszał się na kołach. Ich dobór wcale jednak nie jest taki prosty. Koła powinny być wystarczająco sztywne i odpowiednio przyczepne. Ważne jest również, aby zbudowany z ich wykorzystaniem układ jezdny pozwolił robotowi na uzyskanie dużej siły i zwrotności.

Każdy miłośnik Formuły 1 wie, że duża moc silnika to nie wszystko, aby zwyciężyć – nie mniej ważna jest odpowiednia przyczepność, o czym decyduje ogumienie. Powinno ono charakteryzować się jak największym współczynnikiem tarcia i odpowiednią elastycznością. W amatorskich konstrukcjach robotów minisumo w roli ogumienia sprawdzają się uszczelki do okien czy dętki rowerowe (bardziej zaawansowane rozwiązania wykorzystują neopren, ciętką gumę czy silikon).

Kolejne zagadnienie konstrukcyjne to silniki z przekładniami. Optymalny ich dobór



to sprawa skomplikowana, znacznie wykraczająca poza ramy tego artykułu. Warto jedynie wiedzieć, że muszą one generować dużą siłę napędową (aby robot sprostał przeciwnikowi) przy właściwie dobranej maksymalnej prędkości obrotowej (aby robot był odpowiednio zwinny). I co najważniejsze, silniki nie mogą być za ciężkie – zazwyczaj waga silników jest swoją główną składową ciężaru całego robota. W grę wchodzi więc w zasadzie wyłącznie silniki elektryczne prądu stałego lub zmiennego, serwomechanizmy czy silniki krokowe. W amatorskich rozwiązaniach najczęściej stosuje się szrotkowe silniki prądu stałego, na przykład modelarskie lub wymontowane z zabawek mechanicznych czy też wkrętarek. Profesjonaliści korzystają zwykle z silników bezszczotkowych, które niestety wymagają skomplikowanych układów sterowania.

Dobrze jest, gdy silnik jest zaopatrzony w przekładnię. Dobranie przekładni pasującej do silnika jest trudne i zazwyczaj kosztowne, a samodzielne wykonanie przeważnie niemożliwe. Ważne jest, aby moment obrotowy silnika i przełożenie przekładni były dobrze dobrane do przyczepności układu jezdnego – więcej na ten temat w następnym numerze Eksperta.

MOC Z NAMI

Nawet najlepszy silnik nie wykona choćby jednego obrotu bez zasilania, a bez baterii zamontowana w robocie elektronika będzie bezużyteczna. Z Ekspertem dowiemy się, na jakie parametry akumulatorów musimy zwrócić szczególną uwagę.

Po pierwsze, należy dobrać odpowiednią wartość napięcia zasilania, dostosowując ją do silników i elektroniki robota. Równie ważna jest pojemność baterii – musi ona przecież wystarczyć do tego, aby nasz wojownik wytrwał na ringu przynajmniej kilka minut. I w końcu nie można zapomnieć o właściwej wydajności prądowej zastosowanych ogniw – to ona decyduje o tym, ile energii jesteśmy w stanie „przepompać” z baterii do silników w danej chwili. Najczęściej do zasilania robotów klasy minisumo stosuje się różnego typu akumulatory, od popularnych ogniw nikielowo-kadmowych czy nikielowo-wodorkowych w rozmiarze AA, poprzez stosowane w telefonach komórkowych baterie litowo-jonowe po najnowsze (i niestety najdroższe) baterie litowo-polimerowe. Trzeba także pamiętać, że poza silnikami akumulatory są jednym z najcięższych elementów robota.

ESP w robocie

Aby ograniczyć niepożądane zjawisko buksowania kół, w profesjonalnych rozwiązaniach robota wyposaża się w zaawansowane układy kontroli trakcji, podobne do tych stosowanych w samochodach.

GRUNT TO CZUJNIKI

Dobre dobrany zestaw czujników to podstawa skutecznego działania. To, w jaki sposób zostanąysterowane układy jezdne, będzie niewątpliwie zależało od tego, co na podstawie odczytów z czujników zdecyduje układ sterujący. Ze względu na wymagania, jakie stawia się mechanicznemu wojownikowi sumo, konieczne jest użycie kilku typów sensorów służących do wypełnienia trzech głównych zadań: wykrywania brzegu ringu, namierzenia przeciwnika i samolokalizacji.

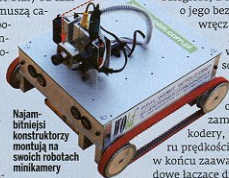


Aby zawodnik mógł jak najdłużej utrzymać się w polu walki, konieczne jest użycie czujników pozwalających na detekcję krawędzi maty. Na brzegu ringu znajduje się biała linia, z wykryciem której bez problemu powinny poradzić sobie na przykład czujniki koloru. W najprostszym wydaniu robota wystarczy, aby czujniki te potrafiły odróżnić kolor biały od czarnego. W tym celu muszą cały czas obserwować powierzchnię dochoy i informować układ sterowania o najechnaniu na białą linię. W profesjonalnych rozwiązaniach można w tym celu wykorzystywać bardziej rozbudowane systemy optyczne – nawet układy wykorzystujące kamery.

Czujników nigdy za dużo

Należy pamiętać, że duża liczba czujników będzie wymagała zastosowania odpowiednio potężnego układu sterowania i zaawansowanych algorytmów przetwarzania sygnałów. Pozwoli to jednak na znaczny wzrost uniwersalności rozwiązania, a także ułatwi eksperymentowanie i wprowadzanie nowatorskich pomysłów – być może prowadzących do stworzenia robota, którego zachowanie zaszkodzi nawet samemu konstruktorowi. Ograniczenie liczby czujników jedynie do podstawowych sensorów spowoduje zmniejszenie wymagań stawianych sterownikowi robota, ale praktycznie do zera zredukując możliwość łatwego dokonywania modyfikacji w kodzie sterującym.

Najambitniejsi konstruktorzy montują na swoich robotach minikamery



ki podczterwieni, czujniki indukcyjne, pojemnościowe, sensory dotykowe czy nawet minikamery. Niestety, pomiary pochodzące z takich czujników nie zawsze są wiarygodne. Dlatego też dobrym pomysłem jest równoczesne użycie czujników dokonujących pomiaru różnymi metodami i mających różny zasięg. Dzięki temu robot będzie w stanie wykryć przeciwnika na odległość, a także będzie informowany o jego bezpośredniej bliskości czy wręcz fizycznym kontakcie.

Najwięcej trudności sprawia ostatnie z zadań sensorów – samolokalizacja robota. Tutaj z pomocą przychodzi nam systemy odometrii wykorzystujące zamontowane na robocie enkodery, układy służące do pomiaru prędkości i przyspieszeń, czy też w końcu zaawansowane systemy hybrydowe łączące działanie kilku działających w różny sposób sensorów. Ponieważ prosty robot jest w stanie poradzić sobie bez systemu lokalizacji, Ekspert pozostawi ten temat do samodzielnych rozważań bardziej zaawansowanym Czytelnikom.

NAZYWAM SIĘ KONTROLER. MIKROKONTROLER

Nadeszła pora na przedstawienie mózgu robota, czyli jego układu sterowania. Centralną częścią tego układu jest zazwyczaj mikrokontroler, który zawiąduje pracą całego robota. Mikrokontroler to układ scalony, zawierający w sobie

mikroprocesor, pamięć programu, pamięć operacyjną oraz inne układy: czasowe, przerwań, komunikacji szeregowej. W zadaniu mikrokontroler to kompletny system komputerowy upakowany w jednym układzie scalonym. W pamięci programu umieszczony powinien być kod z zaszytym algorytmem postępowania robota, pamięć operacyjna zaś służy do przechowywania informacji o stanie robota i jego otoczenia. Układy peryferyjne służą do komunikacji mikrokontrolera ze światem zewnętrznym. Poprzez peryferia nazw mikrokontroler będzie pobierał



informacje z układów sensorycznych, a także, za pośrednictwem wykonawców układu sterującego zaprojektowanego stosownie do rodzaju użytych silników) będzie nadzorował pracę napędów robota. Aby zapewnić podstawową funkcjonalność, wystarczy naszego małego wojownika wyposażać w prosty 8-bitowy mikrokontroler na przykład z rodziny AVR. Robot, który ma się zachowywać w sposób bardziej zaawansowany, wymagać będzie zastosowania układów 32-bitowych, na przykład ARM.

Wymagania procesora

Pracując na wymagania współczesnych gier i programów komputerowych, parametry mikrokontrolerów dla robotów minimum nie rzucają na kolano: w większości wypadków wystarczy nam 8-bitowa jednostka centralna, pamięć programu rzędu kilku czy kilkunastu kilobajtów, pamięć operacyjną liczoną w setkach bajtów, a prędkość takowania wyrażoną w pojedynczych megahercach. Mimo niewielkich wymagań taki mikrokontroler pozwoli na stosowanie nawet bardzo rozbudowanych algorytmów czy obsługi zdarzeń w czasie rzeczywistym.

POMYŚL, ZANIM POJEDZIESZ

Wiele już wiemy o poszczególnych modułach robota. Skąd jednak robot ma wiedzieć, kiedy zaatakować, w którym momencie bronić się, a w którym uciekać? Przecież układ napędowy sam nie zdecyduje, w którą stronę pojechać, a same czujniki nie powieźdzą, jak namierzyć przeciwnika. Wszystko pozostaje w rękach programisty, który powinien napisać odpowiedni

algorytm sterowania naszego robota. To od niego zależy, jak wojownik będzie się zachowywał. Algorytm to przepis na osiągnięcie zwycięstwa w zawodach. W ostatniej części cyklu Ekspert podpowie, jak napisać algorytm postępowania ro-

Inteligentne maszyny

Źródłem algorytmów sterowania robotów są także dziedziny wiedzy, jak teoria sterowania i sztuczna inteligencja. Nowoczesne algorytmy sterowania są hierarchicznymi algorytmami wykorzystującymi sprzężenie zwrotne. Sterowanie w nich odbywa się najczęściej na kilku poziomach: pod-rzędym, zapewniającym regulację prędkości czy momentów napędowych, a także wstępną obróbkę sygnałów sensorycznych, oraz na poziomie nad-rzędym, odpowiedzialnym za odrucho robota czy też jego „przemysłowe” działania (dotkliwość może także pojawić się poziom planowania strategii).

botą oraz opracować na jego podstawie program komputerowy. Dzięki temu, gdy czujniki robota wykryją krawędź ringu, robot natychmiast przejdzie do fazy wycofywania się ku środkowi docho, a gdy sensory przekazują informację o położeniu przeciwnika, robot uda się w jego kierunku. O ostatecznym zwycięstwie zwykle nie decyduje więc wysoki poziom komplikacji robota i wyposażenie go w drogie, zaawansowane rozwiązania. Kluczem do sukcesu jest umiejętność i dobrze przemyślane połączenie poszczególnych, porządnie zaprojektowanych i niezawodnych komponentów, pozwalające na wykorzystanie cechy układów złożonych, zwanej emergencją. O tym, jak osiągnąć tę cechę, przeczytamy w kolejnych odcinkach cyklu.



Przy opracowywaniu oprogramowania robota należy korzystać z dedykowanych narzędzi

Trudne terminy

- » **algorytm** – formuła lub procedura pozwalająca rozwiązać pewien problem.
- » **emergencja** – cecha systemu, polegająca na tym, że własności systemu jako całości nie można wywnioskować z własności poszczególnych części składowych, rozpatrywanych z osobna.
- » **enkoder** – urządzenie do pomiaru położenia, które zlicza impulsy odpowiadające ruchowi obrotowemu, charakteryzujące się stałą liczbą impulsów na obrót.
- » **odometria** – sposób lokalizacji polegający na określeniu przemieszczenia pojazdu na podstawie pomiaru kąta obrotu jego kół.
- » **samolokalizacja** – określenie położenia robota w zadanym układzie odniesienia.
- » **sensor** – urządzenie informujące o wartości mierzonej wielkości fizycznej.
- » **serwozaczepienie** – automatacyjne urządzenie sterujące. Jego elementem wykonawczym jest silnik sterowany sygnałem będącym różnicą prędkości aktualnej i zadanej.
- » **układ programowalny** – cyfrowy układ scalony, którego wewnętrzna struktura fizyczna może być zmieniana pod wpływem różnych czynników.

Warto zajrzeć...

Adresy WWW:

- 1 www.konar.pwr.wroc.pl/regulamin_minisumo
- 2 www.konar.pwr.wroc.pl
- 3 www.skalp.pg.gda.pl
- 4 www.mobot.pl
- 5 www.robotika.cz
- 6 www.e-robotics.co.uk/minisumo
- 7 www.robots-dreams.com

Autorzy artykułu (Mariusz Janiak, Janek Kudzierski, Jakub Malewicz, Robert Muszyński oraz Karol Sydor) związani są z działającym przy Wydziale Elektroniki Politechniki Wrocławskiej Kołem Naukowym Robotyków „KoNaR”

Co jest potrzebne, żeby zbudować robota

Aby ułatwić Czytelnikom przygotowanie się do wykonania własnego robota minimum według opisu prezentowanego w kolejnych numerach, Ekspert przedstawia orientacyjny spis podstawowych materiałów i narzędzi.

Materiały:

- układ mechaniczny robota:
 - blacha 1 milimetrowa (aluminium) – 20 cm x 20 cm
 - laminat – 15 cm x 10 cm
 - pleksiak – 15 cm x 15 cm
 - serwo-mechanizmy Hitec HS-422 – 2 sztuki
 - uszczelka do okien typu E
 - słupki dystansowe, minizawiasy, śrubki M3

- układ zasilania, wykonawcy i czujniki:
 - akumulatory niklowo-wodorkowe w rozmiarze AA – 4 sztuki
 - dalmierz ultradźwiękowy M0BOF-LUS (www.mobot.pl) – 1 sztuka
 - dalmierz podczerwony SHARP GP2D02 (www.mobot.pl) – 1 sztuka
 - układ scalony L298 – 1 sztuka
 - drobne elementy dyskretnie (rezystory, kondensatory, diody, diody świecące, tranzystory, fototranzystory, mikrowyłączniki)
 - koszyki termokurczliwe
- układ sterownika:
 - procesor Atmega48 – 1 sztuka

- układ LM339 – 1 sztuka
- układ 74HC244 – 1 sztuka
- złącze db25F katłowe (męskie)
- taśma 10-żyłowa – 1 m
- zaciskanka złącza żeńskie na taśmę – 2 sztuki
- drobne elementy dyskretnie (rezystory, kondensatory, potencjometry montażowe, diody)
- laminat jednostkowy (ewentualnie płytka uniwersalna)
- goldpiny, podstawki

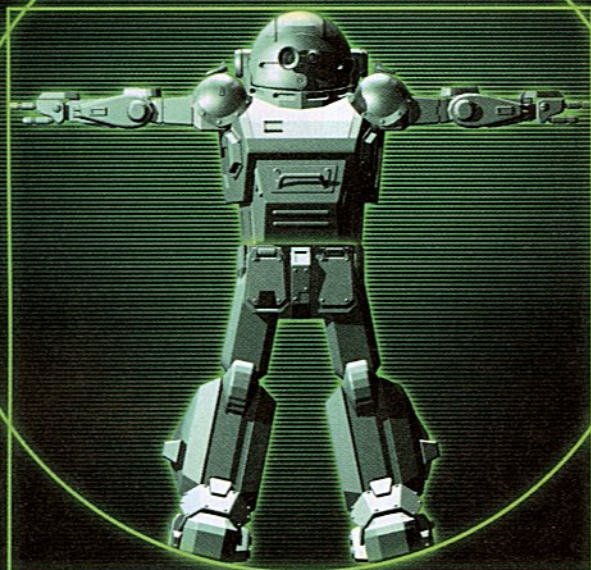
Narzędzia:

- układ mechaniczny robota:
 - imadło, młotek, pilniki

- nożyce do cięcia blachy, pilka do metalu
- wiertarka, wiertła (2 do 5 mm)
- gwintowniki i narzynka M3
- wrętki, szpycze
- papier ścierny
- układ zasilania, wykonawcy i czujniki, układ sterownika:
 - środki do wykonania obwodów drukowanych i lutownica
 - papier ścierny (1000)
 - cęgi, pilniki, miniwiertarka, wiertła (0,8 i 1 mm)
- oprogramowanie:
 - Bascom AVR DEMO Minikontroler (www.mcselab.com)

Na CD i DVD

PDF z rozszerzeniem artykułu
Projekty obudowy



Popularne roboty kroczące

Najbardziej znanymi robotami kroczącymi są humanoidy Honda P3 oraz Asimo. Są to bardzo złożone konstrukcyjnie i kosztowne maszyny, zbudowane przez światowe, bardzo bogate koncerny. Równie skomplikowany jest czteroosny robot BigDog, który skonstruowany został ze środków amerykańskiej agencji rządowej DARPA.



FOT.: HONDA

Mocny hardware

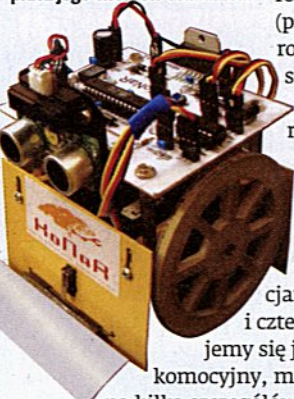
Robotowi minisumo należy zapewnić solidne ciało i sprawny napęd. Poznajmy więc różne rozwiązania i wybierzmy najlepsze

artykułu Ja mam Robot! zamieszczonego w Ekspercie 5/2007 dowiedzieliśmy się, czym jest i z jakich elementów zbudowany jest robot klasy minisumo. W tym artykule Ekspert opíše, jak skonstruować ciało przyszłego wojownika (na przykładzie robota ●). Dowiemy się także, w jaki sposób zapewnić mu odpowiedni napęd.

NA KOŁACH CZY NA NOGACH

Na początku zajmiemy się najbardziej przyziemną częścią robota, czyli jego systemem lokomocji. W bojowych platformach mobilnych, poruszających się po łądzie, zastosowanie znajdują głównie napęd kołowy oraz gąsienicowy. Zazwyczaj nie buduje się robotów kroczących (patrz ramka Popularne roboty kroczące) – są one skomplikowane, kosztowne i trudne w sterowaniu.

Mechaniczna część robota klasy minisumo, nazwanego przez jego twórców Grabarzem

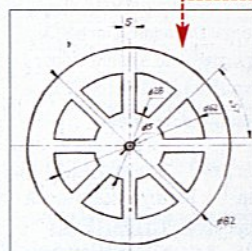


Napędów kołowych jest kilka typów (patrz ramka na następnej stronie). Najczęściej stosowanymi konstrukcjami są systemy dwu- i czterokołowe. Gdy zdecydujemy się już na jakiś system lokomocyjny, musimy zwrócić uwagę na kilka szczegółów przy jego wykonaniu.

- W przypadku kół najważniejsze parametry to:
- **średnica** – pośrednio przekłada się na prędkość i siłę robota,
 - **szerokość** – wpływa na przyczepność koła,
 - **materiał opony (ogumienie)** – decyduje o współczynniku tarcia koło-podłoże.

Konstruktorzy robotów najczęściej używają gotowych kół – kupionych w sklepie modelarskim lub wymontowanych z zabawek. Mając dostęp do obrabiarek, można koła wytworzyć samodzielnie, najlepiej z aluminium lub pleksiglasu. Można również zastosować bardziej „partyzancką” metodę, a mianowicie wykonać koła na przykład z zakrętek od słoików (wystarczy wywiercić w nich otwory i już można zamontować je do robota).

W robocie stworzonym przez autorów tego artykułu zastosowany został dwukołowy układ jezdny. Koła wykonane zostały z pleksiglasu o grubości 5 mm, według projektu ●. Ich wycięcie zrealizowała firma wykonująca tablice reklamowe.



Jeśli chcemy zlecić wykonanie kół (na przykład takich ●) odpowiedniej firmie, najpierw musimy narysować szczegółowy projekt



Kolejnym zadaniem jest zapewnienie kołom odpowiedniej przyczepności do podłoża.

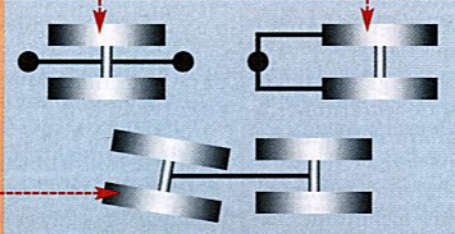
W amatorskich konstrukcjach w roli ogumienia używane są uszczelki do okien, najczęściej o profilu E oraz dętki rowerowe. Bardziej zaawansowanym rozwiązaniem jest zastosowanie neoprenu – miękkiego materiału używanego do produkcji skafandrów do nurkowania. Profesjonaliści wytwarzają opony, oblewając koła ciętką gumą lub masą silikonową. Te ostatnie rozwiązania dają największą przyczepność, jednak wiążą się ze znacznym wzrostem kosztów, gdyż wymagają zaangażowania firm zajmujących się wytwarzaniem i obróbką tworzyw sztucznych. Dobrym rozwiązaniem jest naklejenie na koła pasków z gumy ● (najlepiej klejem cyjanoakrylowym).

Oczywiście należy przewidzieć sposób połączenia kół z napędem – Ekspert poleca użycie wkrętów (jeśli korzystamy z serwo mechanizmów) lub mocowanie na wcisk (w wypadku silników).

i Rodzaje napędów

Napęd dwukołowy można zrealizować na dwa sposoby ● – os, na której zamontowano koła jezdne, może przechodzić przez środek ciężkości robota lub też znajdować się z dala od niego (w obu wypadkach należy umieścić dodatkowe punkty podparcia). Użycie dwóch kół zapewnia prostotę układu napędowego, jak również upraszcza sterowanie, przy dobrych własnościach lokomocyjnych.

Układ czterokołowy również występuje w dwóch odmianach: z napędem na jedną os i dwoma kołami skrętnymi ● lub z napędem na cztery nieskrętne, łączone parami koła (parę tworzą koła po jednej stronie platformy). W pierwszym wypadku sterowanie jest identyczne jak sterowanie samochodem. Stosując wersję z kołami łączonymi w pary tracimy nieco na zwrotności, lecz zawsze jakieś koło ma kontakt z podłożem, co pozwala na wydostanie się z wielu opresji na ringu.



Systemy wielokołowe w największym stopniu zapewniają stały kontakt grupy kół z podłożem. Występują one w takich samych odmianach, jak systemy czterokołowe. Niestety, w przypadku robotów minisumo (ograniczone wymiary) tego typu rozwiązania wymagałyby zastosowania kół o niewielkiej średnicy i małych silników, co pociąga za sobą zmniejszenie mocy całego robota.

System gąsienicowy, niemal identyczny jak ten stosowany w czołgach, gwarantuje bardzo dużą powierzchnię styku robota z podłożem. Niesie to jednak za sobą zmniejszenie nacisku jednostkowego na podłoże oraz znaczne tarcie boczne przy obracaniu się robota w miejscu, co skutkuje dużymi stratami energii. W napędzie gąsienicowym należy zapewnić docisk gąsienicy do podłoża na całej jej długości, w przeciwnym razie jej stosowanie będzie bezcelowe.

i Rodzaje serwomechanizmów

Wyróżniamy serwomechanizmy o stałej rotacji oraz zwykłe. Typowe serwo ma ograniczony kąt obrotu do około 180 stopni. Aby można je było zastosować w robocie, trzeba usunąć (na przykład pilnikiem) znajdującą się wewnątrz niego mechaniczną blokadę ograniczającą ruch, wymontować potencjometr oraz wyprowadzić na zewnątrz kable zasilające silnik. Tak zmodyfikowane serwo jest w zasadzie zwykłym silnikiem, jednak ma już fabrycznie wykonaną przekładnię i obudowę ułatwiającą montaż. Istnieją też serwa ze stałą rotacją, które mogą być zastosowane jako napęd robota bez zmian, gdyż ich wał obraca się o 360 stopni.



W robocie stworzonym przez autorów artykułu zastosowano serwomechanizmy Hitec 325HB, które do obudowy przymocowano za pomocą wkrętów ●. Bez problemu można jednak użyć również innych serw – na przykład TowerPro SG-5010 ●

PARA BUCH, KOŁA W RUCH

Wiemy już, w jaki sposób wykonać koła robota. Musimy się więc zastanowić nad tym, jak je wprawimy w ruch. W niewielkich robotach mobilnych stosowany jest niemal wyłącznie napęd elektryczny.

Można wyróżnić dwie grupy tego typu napędu: silniki oraz serwomechanizmy. Serwomechanizmy są stosowane w mniejszych robotach (mają zwykle małe momenty obrotowe i siłę). Zaletą serwomechanizmów o stałej rotacji jest prosty układ sterowania w stosunku do silników, wadą – ich cena. Te serwomechanizmy, które wymagają modyfikacji (patrz ramka Rodzaje serwomechanizmów) nie różnią się pod względem sterowania od silników, jednak ich obudowa pozwala na prosty montaż. Silniki natomiast można stosować we wszystkich rodzajach robotów. Do ich sterowania, podobnie jak do zmodyfikowanych serw, należy użyć specjalizowanych układów (przeczytamy o nich w kolejnej części cyklu o robotach).

W przypadku dwukołowych robotów minisumo stosuje się zwykle dwa identyczne małe elektryczne silniki modelarskie lub serwomechanizmy. Rzadziej używane są silniki krokowe (ciężko jest znaleźć lekkie silniki krokowe o dostatecznej mocy).

Gdy znajdziemy już odpowiednie silniki, musimy spowodować, aby wprawiły w ruch koła. Do tego służy układ przeniesienia napędu. W robotach minisumo najczęściej używane są gotowe przekładnie, wymontowane z wkrętarek akumulatorowych lub z różnego rodzaju sprzętu AGD i zabawek. Niezmiernie rzadko zdarza się, aby konstruktor robota zdecydował się na samodzielne zaprojektowanie i wykonanie takiej przekładni ze względu na wysoki stopień skomplikowania, a także wymaganą precyzję. Innym wyjściem jest zastosowanie serwomechanizmów – mają one wbudowane przekładnie, więc do wału serwa od razu mocujemy koło.

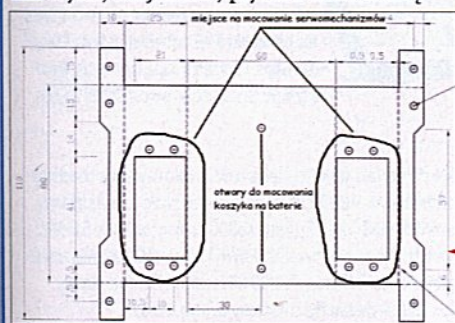
Powstaje jednak pytanie: jak dobrać odpowiedni silnik wraz przekładnią (tak zwany zespół napędowy)? O tym, jak to zrobić, przeczytamy w dokumencie PDF z płyty Eksperta.

i Skąd wziąć silnik

Wiemy już, że powinniśmy szukać silnika z gotową przekładnią. Prawdziwą kopalnią silników są zabawki. Aby znaleźć silnik, warto także zajrzeć do niepotrzebnych już czytnika płyt CD, napędu dyskiety i innych tego typu urządzeń. Jeśli wszystko zawiedzie, możemy udać się do sklepu modelarskiego lub sklepów z małą elektroniką.

JAK TO WSZYSTKO POŁĄCZYĆ

Element nośny, czyli korpus robota, służy do przymocowania zespołów napędowych, czujników, płytki z elektroniką

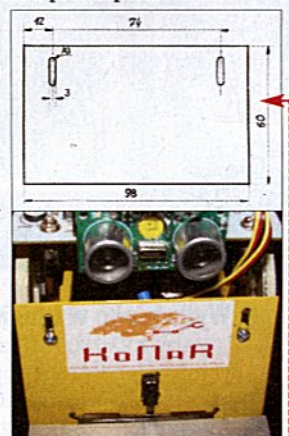


i źródła zasilania. Ponieważ nasz minirobot z założenia narażony jest na zderzenia, musimy zadbać, aby korpus był wytrzymały i odporny, a zarazem lekki, gdyż stanowi jeden z największych elementów konstrukcyjnych. Zwykle wykonywany jest więc z profili aluminiowych, cienkich blach lub laminatu. Tej części platformy należy poświęcić szczególnie dużo uwagi zarówno przy projektowaniu, jak i konstruowaniu – ewentualne poprawki mogą okazać się bardzo trudne lub wręcz niemożliwe do zrealizowania. Na rysunku ● pokazano projekt korpusu użytego w konstrukcji Eksperta. Jest on wykonany z blachy o grubości 1 mm.

ZBROJA WOJOWNIKA

Ważnym elementem konstrukcji mechanicznej robota jest jego obudowa. Jej zadaniem jest zabezpieczenie podzespołów robota

przed atakami przeciwnika oraz wprowadzanie w błąd jego sensorów. Z analiz wyników, że obudowa powinna mieć kształt ostrosłupa o możliwie najmniejszym kącie nachylenia ścian do podłoża. Dzięki temu fale emitowane przez sonary oraz promieniowanie z czujników podczerwonych odbijają się od naszego robota w taki sposób, że nie powrócą do przeciwnika. Unikniemy w ten sposób namierzenia. Tworząc obudowę, warto zastosować matowe (co powoduje tłumienie promieniowania podczerwonego) i miękkie (tłumi fale wysyłane przez sonar) pokrycie ścian.



Grabarz został wyposażony w prostą w konstrukcji osłonę przednią i tylną – obie wykonane według projektu ● z blachy o grubości 1 milimetra. Oczywiście zarówno obudowę, jak i osłony warto przyozdobić, czyniąc naszą maszynę niepowtarzalną

Nie każda obudowa jest niewidoczna dla czujników przeciwnika. Należy więc robota wyposażyć w broń – oczywiście tylko taką, która nie niszczy przeciwnika, a spowoduje jedynie utratę możliwości ataku, obrony czy ucieczki. Do tego celu najsukrotniejsze są różnego rodzaju pługi, podnośniki i psychacze, które choćby na chwilę zdestabilizują lub zdeorientują przeciwnika. Dzięki temu staje się on łatwym celem i można go w prosty sposób wypchnąć z ringu. Opisany przez Eksperta robot został wyposażony w pług wykonany z blachy aluminiowej o grubości 0,5 mm ● **PoIWR**



Autorzy artykułu (Mariusz Janiak, Janek Kędziński, Jakub Malewicz, Robert Muszyński oraz Karol Sydor) związani są z działającym przy Wydziale Elektroniki Politechniki Wrocławskiej Kołem Naukowym Robotyków „KoNaR”

Drogowskaz cyklu	Ekspert
Budowa robota minisumo	5/2007
Mechaniczna konstrukcja	6/2007
Sensory	7/2007
Sterownik napędów i zasilanie	8/2007
Mikrokontroler sterujący	9/2007
Algorytm sterowania	10/2007

Indeks	
Dalmierze ultradźwiękowe i optyczne	54
Czujniki zderzenia	55
Wykrywanie białej linii	56

Na CD i DVD
Zdjęcia z zawodów robotów minisumo



FOT.: SCREEN DVD VISION, MONTAŻ: KOMPUTER ŚWIAT, EKSPERT

Dobry wzrok robota

Bez zmysłów nasz robot będzie bezbronny. W tym odcinku kursu nauczymy się wszystkiego o sensorach optycznych i dźwiękowych

Z dwóch poprzednich artykułów o budowie własnego robota minisumo dowiedzieliśmy się, jak wykonać jego mechaniczną konstrukcję. Najwyższy czas uzbroić robota w czujniki. W tym artykule Ekspert pokaże, jakie sensory warto zainstalować w naszym urządzeniu. Poznamy zasadę działania różnych czujników oraz niezbędnych do ich funkcjonowania modułów.

Jakie czujniki wybrać

Głównym zadaniem robota klasy minisumo jest wypchnięcie przeciwnika poza ring. Nasz robot powinien więc wykrywać swojego rywala. Jest kilka typów sensorów, które można wykorzystać do tego celu. Z Ekspertem poznamy zasadę działania dalmierzy optycznych i sonarów ultradźwiękowych. Dowiemy się też, jak zamontować zwykły stycznik, który wykryje atak przeciwnika.

Patrz, myśl, działaj

Jan Kędzierski

Współautor



Algorytm zaimplementowany w robocie powinien sterować nim według zasady: patrz, myśl, działaj. Warto więc dobrze zastanowić się nad doborem odpowiedniego zestawu czujników, który pozwoli na realizację pierwszej fazy tej zasady. Dzięki właściwym sensorom i dobremu algorytmowi nasz robot może stać się bardzo sprytny i z łatwością wysłać swoich przeciwników poza ring.

i Powstało forum!

Budowa własnego robota to trudne i czasochłonne zadanie. Warto więc korzystać z doświadczeń innych konstruktorów i autorów tego artykułu. Ekspert zachęca do odwiedzania założonego przez jednego z Czytelników forum poświęconego budowie własnego robota minisumo.

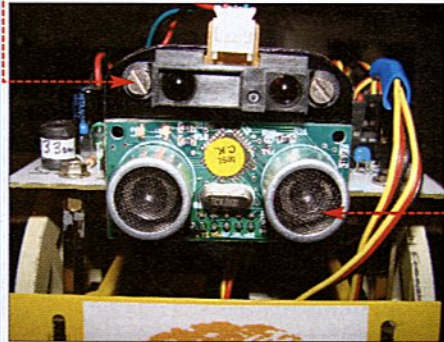
Oprócz namierzania oponenta nasz robot minisumo powinien również rozpoznawać oznaczony białą linią brzeg dohyo (ringu, na którym toczony są pojedynki robotów). Ekspert pokaże więc, jak zbudować czujniki wykrywające białą linię. Ponieważ są one dość proste w konstrukcji, bez większych trudności możemy je wykonać samodzielnie. O tym, jak to zrobić, dowiemy się z części Wykrywanie białej linii (strona 56).

i Odporne czujniki

Czujniki, w które wyposażymy naszego robota, powinny być nie tylko precyzyjne, ale również odporne na różne trudne do przewidzenia w fazie projektowania sytuacje. Często zdarza się, że doskonale sprawdzające się podczas domowych testów sensory w trakcie zawodów nie działają zgodnie z założeniami (przyczyny mogą być różne – o niektórych przeczytamy w części Czujniki zderzenia). Dobrym pomysłem jest więc zamocowanie dodatkowych zwykłych styczników wraz ze specjalnie przygotowanymi zderzakami, które niezależnie od warunków oświetleniowych wykryją atak przeciwnika. Warto także zainstalować czujniki nadmiarowe o różnych zasadach pomiarowych.

DALMIERZE ULTRA

Do namierzania przeciwnika najlepiej jest zastosować dalmierze o różnych zasadach pomiarowych – na przykład czujniki wykorzystujące ultradźwięki oraz promienie podczerwone. Ponieważ samodzielne wykonanie tego typu czujników jest kosztowne i czasochłonne, najlepiej skorzystać z gotowych rozwiązań. Ekspert podpowie, jakie dalmierze warto zastosować.



Sonar ultradźwiękowy

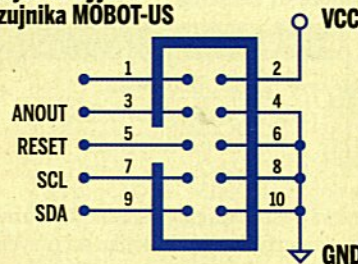
Ten typ czujników jest coraz częściej wykorzystywany w samochodach, w systemach pomagających kierowcom bezpiecznie zaparkować pojazd. Ze względu na niezbyt wygórowaną cenę (około 95 złotych) oraz dobre parametry Ekspert proponuje użyć czujnika MOBOT-US firmy Wobit.



Sonar MOBOT-US jest łatwy w obsłudze i dość szybki. Czas pojedynczego pomiaru wynosi około 30 milisekund – w jednej sekundzie można więc przeprowadzić 33 takie pomiary. Jest to wystarczająco często, aby skutecznie i precyzyjnie namierzyć przeciwnika

Jak wykorzystać czujnik w naszym robocie? Do komunikacji z sonarem MOBOT-US służy dwuprzewodowy interfejs I2C, często stosowany w urządzeniach automatyki przemysłowej. Transmisja danych przy użyciu interfejsu I2C jest typem transmisji szeregowej, czyli takiej, w której poszczególne bity są wysyłane jeden po drugim w takt zegara taktującego magistralę.

Wejścia/wyjścia czujnika MOBOT-US



DŹWIĘKOWE I OPTYCZNE

i Jak sterować sonarem

Dalmerz MOBOT-US obsługujemy w trzech fazach. W pierwszej nakazujemy czujnikowi wykonać pomiar. W tym celu wysyłamy do niego kolejne dwa bajty: **0x10** oraz **0x01**. Następnie czekamy około 30 milisekund (tyle zajmuje wykonanie pomiaru) i wysyłamy do dalmierza bajt o wartości **0x20**. Na koniec odczytujemy dwa kolejne bajty (**L** i **M**). Zmierzona przez sonar wartość uzyskujemy po wykonaniu prostego działania: $L + 256 * M$. Przedstawiony sposób pomiaru dotyczy trybu AUTO urządzenia. Sonar MOBOT-US pozwala jednak ustawić również różne parametry pomiaru. Aby na przykład dokonać pomiaru ze wzmocnieniem równym 30 (zwiększy to znacznie czułość odbiornika ultradźwięków), zamiast bajtu **0x01** wysyłamy do sonaru bajty: **0x02** i **0x1E** (wartość 30 dziesiętnej). Więcej o zaawansowanych możliwościach dalmierza MOBOT-US przeczytamy w dołączonej do niego dokumentacji.

Wykorzystanie dalmierza MOBOT-US sprwadza się do podłączenia napięcia zasilającego 5 V do linii **VCC**, masy do linii oznaczonej jako **GND** oraz do doprowadzenia sygnału z linii **SDA** i **SCL** do głównego sterownika naszego robota (o sterowniku przeczytamy w jednym z następných numerów Eksperta). Obsługa dalmierza jest stosunkowo prosta – opis wykonania przykładowego pomiaru znajdziemy w ramce Jak sterować sonarem.

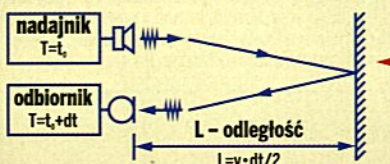
Kłopoty z sonarem

Dalmerz ultradźwiękowy nie jest doskonałym sensorem. Gdy przeciwnik będzie wyposażony w podobne urządzenie, systemy pomiarowe stojących naprzeciw siebie robotów będą się nawzajem zakłócać, a więc wyniki pomiarów nie będą zgodne

i Sonar ultradźwiękowy

Ultradźwięki to drgania sprężyste, których częstotliwość leży poza pasmem słyszalności człowieka. Stosowane często w medycynie, w defektoskopii materiałowej do wykrywania pęknięć i uszkodzeń struktury, a także w systemach alarmowych. Sonary ultradźwiękowe występują również w... przyrodzie – wyposażone są w nie nietoperze, delfiny oraz niektóre owady.

Zasada działania sonaru ultradźwiękowego



Pomiar odległości ultradźwiękami jest pomiarem pośrednim. Polega on na wyemitowaniu kilku impulsów o określonej częstotliwości, a następnie zmierzeniu czasu ich powrotu po odbiciu od napotkanego obiektu. Pomiar trwa stosunkowo długo – wynika to z prędkości rozchodzenia się fal ultradźwiękowych w powietrzu. Należy pamiętać, że większość z dostępnych na rynku gotowych urządzeń pracuje z taką samą częstotliwością 40 kHz – mogą się więc one wzajemnie zakłócać.



Robot **o** wdzięcznej nazwie Rambo bardzo dobrze radzi sobie z przeciwnikami – jego różowe futerko nie dość, że czyni go niewidzialnym dla dalmierzy ultradźwiękowych przeciwników, to jeszcze wielokrotnie pozwoliło mu zdobyć nagrodę publiczności na Otwartych Zawodach Robotów Minisumo we Wrocławiu

z prawdą. Ponadto, jeśli sygnał emitowany przez nadajnik dalmierza dotrze do płaskiej powierzchni nachylonej pod kątem na przykład w górę, nie powróci w pole widzenia odbiornika (dalmierz zachowa się tak, jakby nie wykrył żadnego obiektu). Dodatkowo ultradźwięki w dużym stopniu pochłaniane są również przez materiały gąbczaste, futra i różnego rodzaju materiały włochate.

Dalmerz optyczny

Aby wspomóc w działaniu sonar ultradźwiękowy, warto wyposażyć dodatkowo naszego robota w dalmierz o odmiennym

sposobie działania, na przykład optyczny czujnik odległości wykorzystujący promieniowanie podczerwone. Wśród konstruktorów robotów mobilnych dużą popularnością cieszą się zintegrowane dalmierze firmy Sharp (model GP2D12 można kupić w cenie 39 złotych). Zasada pomiaru odległości przez dalmierz optyczny tylko z pozoru podobna jest do czujnika ultradźwiękowego – emitowane fale podczerwone odbijają się od obiektu i powracają do sensora.

W tym wypadku nie mierzy się jednak czasu powrotu, ale kąt padania odbitych promieni (patrz ramka poniżej). Na wyjściu czujnika uzyskujemy wartość analogową napięcia, która w przybliżeniu jest proporcjonalna do odległości.

Dalmerz ma trzy piny, z których na jeden podajemy napięcie zasilania VCC równe 5 V, a na drugi – masę GND. Trzeci pin jest natomiast wyjściem, które podłączamy do głównego sterownika naszego robota. Więcej o czujniku przeczytamy na stronie 3.

i Jak działa sonar optyczny

Dioda sonaru optycznego emituje fale podczerwone, które po odbiciu się od obiektu trafiają do sensora. Ponieważ składa się on z kilku umieszczonych w jednym rzędzie miniaturowych czujników, możliwe jest zmierzenie kąta padania odbitego światła. Im kąt jest mniejszy, tym dalej od sonaru znajduje się obiekt.

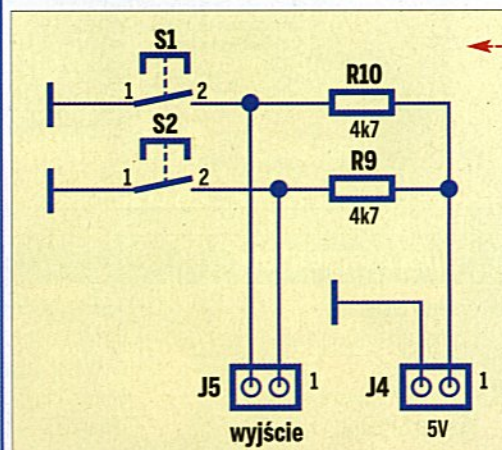
CZUJNIKI ZDERZENIA

Zamontowane na naszym robocie dwa dalmierze (jeden optyczny, drugi ultradźwiękowy) nie zawsze pozwolą namierzyć przeciwnika na ringu. Jeśli drugi robot nie będzie znajdował się w zasięgu czujników, nasz wojownik będzie musiał, obracając się, sprawdzić, gdzie jest przeciwnik – w tym czasie będzie zupełnie bezbronny. Poza tym, promieniowanie z dalmierzy naszego robota może odbić się od obudowy przeciwnika, nie powracając do nich. Może również zostać pochłonięte przez materiał, z którego będzie wykonana obudowa oponenta. Kłopoty mogą również wystąpić, jeśli przeciwnik naszego robota wyposażony jest w podobne do naszych dalmierze – w takiej sytuacji mogą się one wzajemnie zakłócać.



Co zrobić, aby uodpornić naszego robota na te wszystkie sytuacje? Dobrym pomysłem okazuje się zamocowanie na ściankach lub zderzakach naszego robota dodatkowo zwykłych styczników. Można je wykonać zgodnie ze

schematem. Działanie tak wykonanych czujników zderzenia jest bardzo proste – gdy styczniki zostaną zwarte, na wyjściu układu pojawi się sygnał.



Elementy S1 i S2 to zwykłe mikro styczniki podłączone przez rezystory do zasilania. Wyprowadzenia styczników możemy podłączyć bezpośrednio do głównego sterownika

WYKRYWANIE BIAŁEJ LINII

Wiemy już, za pomocą jakich sensorów sprawdzić, jak daleko od robota znajduje się przeciwnik. Naszego wojownika musimy jednak wyposażyć jeszcze w czujniki, które pozwolą wykryć brzeg ringu. W tym celu najlepiej zastosować cztery sensory koloru, po jednym w każdym rogu robota.

Budowa czujników koloru wykrywających białą linię jest stosunkowo prosta, dlatego najlepiej skonstruować je samodzielnie. Sensor koloru składa się z odbiornika, nadajnika oraz układu dopasowującego sygnał podawany do głównego sterownika. Odbiornikiem jest fototranzystor – element czuły na natężenie światła. Nadajnik to dioda LED emitująca promieniowanie podczerwone.

Napięcie uzyskane na wyjściu fototranzystora ma wartość proporcjonalną do jasności powierzchni, na którą jest on skierowany. Ponieważ robot będzie się poruszał po czarnej powierzchni, a brzeg ringu jest biały, musimy rozróżnić tylko dwa stany: po-



wierzchnia czarna oraz biała. Wystarczy więc, że do głównego sterownika robota dostarczymy logiczną wartość 1 lub 0. Aby tak się jednak stało, niezbędne jest wykonanie układu zamieniającego analogową wartość napięcia na jej cyfrowy odpowiednik. Jego schemat Ekspert zamieszcza poniżej.

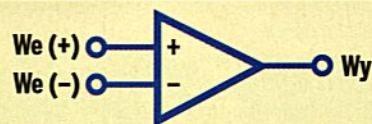
Montaż czujnika białej linii

Wszystkie użyte do budowy czujnika białej linii fototranzystory i diody nadawcze muszą być jednakowe. Najłatwiej uda nam się je zakupić w obudowach 3-milimetrowych przypominających zwykłą diodę świecącą LED. W trakcie montażu należy zwrócić uwagę na ich prawidłową polaryzację (plus i minus). Fototranzystory i diody mocujemy parami i umieszczamy na wysokości około 1 centymetra od powierzchni ruchu. Dodatkowo, zarówno na nadajniku, jak i odbiorniku z kawałka

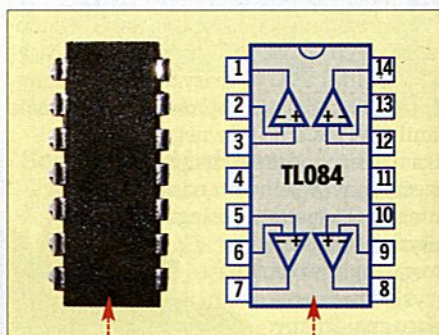


Wzmocniacze operacyjne

Jeżeli sygnał wejściowy zostanie doprowadzony do wejścia **We (-)** (nazywanego wejściem odwracającym) wzmacniacza operacyjnego, to na wyjściu pojawi się sygnał w fazie przeciwnej. Jeśli natomiast sygnał wejściowy zostanie doprowadzony do wejścia **We (+)** (wejście nieodwracające), na wyjściu pojawi się ten sam sygnał.



urki (na przykład czarnej koszulki termokurczliwej) wykonujemy osłonkę. Taki zabieg uodporni odbiornik na niekorzystne działanie światła zewnętrznego.



W module czujników kolorów najlepiej zastosować układ TL084 (lub TLC274), który w jednej obudowie kryje cztery wzmacniacze operacyjne.

1 Do budowy jednego czujnika białej linii potrzebujemy fototranzystor, diodę nadawczą, dwa rezystory (10 kiloomów oraz 330 omów) i koszulkę termokurczliwą

2 Najpierw osłaniamy diodę i fototranzystor za pomocą koszulki termokurczliwej

3 Następnie łączymy ze sobą diodę i fototranzystor. Zwracamy uwagę na właściwą polaryzację tych elementów – dłuższa nóżka diody to anoda (A), a dłuższa nóżka fototranzystora to emiter (E). Na koniec przyłutowujemy rezystory

4 Przyłutowujemy trzy przewody (zgodnie ze schematem) oraz izolujemy połączenia

5 Czujnik białej linii jest już gotowy. Po wykonaniu czterech takich elementów podłączamy je do modułu czujników koloru (patrz schemat obok)

Na wyjściu modułu czujników koloru uzyskamy napięcie 5V w przypadku czarnej nawierzchni i 0V, gdy czujnik wykryje białą linię. Poziom jasności powierzchni, przy którym powinien zadziałać czujnik, wyznaczamy, zmieniając oporność rezystora nastawczego R4.

Spis elementów

- **Gotowe elementy:**
 - sonar ultradźwiękowy MOBOT-US
 - dalmierz optyczny Sharp GP2D12
- **Kondensatory:**
 - 1x 100n
- **Elementy półprzewodnikowe:**
 - 4x fototranzystor 3 mm
 - 4x dioda nadawcza podczerwona 3 mm
 - Układ TL084 lub TLC274
- **Inne elementy:**
 - dwa mikrośczytniki
 - przewody
- 4x 10k
- 4x 5M2
- 3x 4k7
- 1x regulowany 20k

Autrzy (Mariusz Janiak, Janek Kedziński, Jakub Malewicz, Robert Muszyński oraz Karol Sydor) związani są z działającym przy Wydziale Elektroniki Politechniki Wrocławskiej Kołem Naukowym Robotyków „KoNaR”

Warto zajrzeć...

- Adresy WWW:**
- 1 <http://robot.alias.pl>
 - 2 www.wobit.pl
 - 3 www.konar.ict.pwr.wroc.pl/module.php?op=download&cmd=click&id=2

Drogowskaz cyklu	Ekspert
Budowa robota minisumo	5/2007
Mechaniczna konstrukcja	6/2007
Sensory	7/2007
Sterownik napędów i zasilanie	8/2007
Mikrokontroler sterujący	9/2007
Algorytm sterowania	10/2007



Cała naprzód!

Jeśli zbudowaliśmy już podstawowe części robota, najwyższy czas wprowadzić go w ruch. Zadbajmy więc o zasilanie i napęd

Z drugiej części cyklu Budujemy własnego robota minisumo (Ekspert 6/2007) dowiedzieliśmy się, jak zamontować do naszego wojownika silniki. Najwyższy czas wprawić więc koła w ruch.

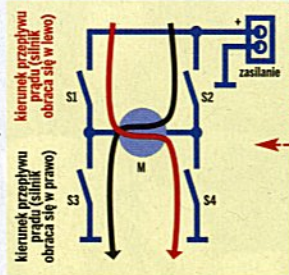
Z tego artykułu dowiemy się, jakie akumulatory zastosować w robocie. Nauczymy się również sterować pracą silników i zaprojektu-

jemy układ, który nam na to pozwoli. Poznamy też sposób na likwidację zakłóceń w układzie elektrycznym naszego urządzenia.

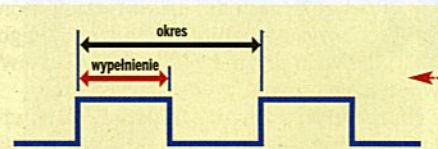
Z montażem przedstawionych w tym artykule układów warto się na razie wstrzymać. Z następnego numeru dowiemy się bowiem, jak wykonać płytkę drukowaną, na której zamontujemy całą elektronikę robota.

KONTROLA SILNIKA

Do sterowania pojedynczym silnikiem używa się tak zwanego mostka typu H, którego schemat ideowy wygląda tak ●. Aby silnik obrócił się w lewo, wystarczy załączyć styczniki **S1** oraz **S4**. Żeby natomiast zmienić kierunek obrotów, trzeba rozłączyć styczniki **S1** i **S4**, a następnie załączyć styczniki **S2** i **S3**. Mostek H daje nam również możliwość hamowania silnika – aby to zrobić, wystarczy załączyć styczniki **S3** i **S4** (lub **S1** i **S2**). Bardzo ważne jest, by w żadnym wypadku nie dopuścić do jednoczesnego załączenia styczników **S1** i **S3** lub **S2** i **S4** – spowoduje to zwarcie źródła zasilania i w konsekwencji nawet pożar!



Wiemy już, jak zmieniać kierunek obrotów silnika. Jak jednak zmieniać jego prędkość obrotu? Z pomocą przychodzi metoda sterowania powszechnie zwana PWM (ang. Pulse Width Modulation – modulacja szerokością impulsu). W metodzie tej wykorzystywany jest generator sygnału prostokątnego z możliwością zmiany częstotliwości (okresu) i wypełnienia ●. Tak się składa, że cewki silników wprawiających robota w ruch uśredniają zasilające je napięcie. Zwiększając zatem wypełnienie impulsów, zwiększamy również napięcie średnie, a w efekcie prędkość obrotową silnika.

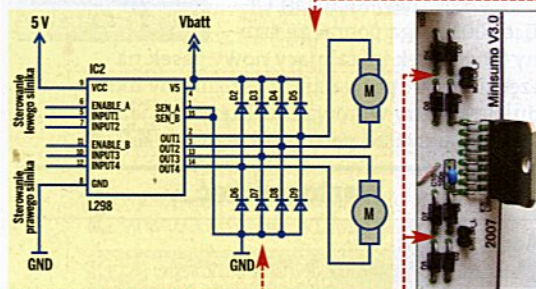


STEROWANIE SILNIKIEM W PRAKTYCE

Wiemy już, że do sterowania silnikami najlepiej wykorzystywać mostek typu H. Zamiast jednak samodzielnie go wykonywać, można użyć układu scalonego L298 ●, w którym są zintegrowane dwa takie mostki (po jednym na każdy silnik). Do poprawnej pracy układu L298 wymaga-

na jest niewielka liczba elementów dodatkowych, które musimy podłączyć zgodnie ze schematem ●.

Układ L298 wyposażony jest w sześć wejść (po trzy dla każdego mostka) oraz cztery wyjścia. Zasilany jest dwoma napięciami, w naszym przypadku o różnej wartości. Napięcie 5 V jest pobierane z tego samego źródła, które zasilają całą elektronikę robota. Mostki H podłączone są natomiast do źródła energii przeznaczonego do zasilania silników o odpowiednio dobranym napięciu (więcej o zasilaniu przeczytamy w części Źródło mocy dla robota). Wyjścia (oznaczone jako **OUT**) służą do podłączenia silników – pierwszy z nich korzysta z wyjść **OUT1** i **OUT2**, drugi zaś z **OUT3** i **OUT4**.



Moduł sterownika napędami L298 wymaga podłączenia tylko kilku diod ●. Ich zadaniem jest ochrona układu przed przepięciami indukowanymi w cewkach silników. Można użyć zwykłych diod prostowniczych

! Uwaga!
Należy zwrócić uwagę na odpowiedni dobór przekroju przewodów użytych w połączeniach pomiędzy sterownikiem a silnikami. Minimalna wartość to 1,5 milimetra kwadratowego.

ENABLE_A i **ENABLE_B** do załączania mostków. W tabeli poniżej Ekspert wyjaśnił znaczenie poszczególnych stanów wejść na przykładzie jednego mostka (więcej na ten temat przeczytamy w części dotyczącej sterownika naszego robota w następnym Ekspercie).

Sterowanie układem L298			
ENABLE_A	INPUT1	INPUT2	akcja
H	H	L	silnik obraca się w lewo
	L	H	silnik obraca się w prawo
L	INPUT1=INPUT2		silnik hamuje
	x	x	mostek wyłączony

H – stan wysoki, L – stan niski, x – stan obojętny

ŹRÓDŁO MOCY DLA ROBOTA

Niestety, nie ma jednej odpowiedzi na pytanie: Jaki typ akumulatora czy baterii powinniśmy wybrać do zasilania naszego minirobota? Każde rozwiązanie ma wady i zalety. Ekspert przedstawi więc najpopularniejsze sposoby zasilania robota minisumo.

Jaki akumulator wybrać

Na rynku dostępnych jest kilka typów akumulatorów o wymiarach odpowiednich do zasilania robota minisumo. Najstarsze z nich, wciąż stosowane między innymi w wkrętarkach, są akumulatory niklowo-kadmowe (Ni-Cd). Nieco nowszym wynalazkiem są akumulatory niklowo-wodorkowe (Ni-MH), zamienniki popularnych paluszków. Najnowocześniejsze, ale wymagające najbardziej skomplikowanego procesu ładowania, są akumulatory litowo-jonowe (Li-Ion) oraz litowo-polimerowe (Li-Pol) używane w telefonach komórkowych. Przed zakupem ogniwa do naszego robota musimy więc dobrze zastanowić się nad tym, jakiego powinno dostarczać napięcia i jaką mieć wydajność prądową oraz pojemność.

Proste ogniwa

Akumulatory Ni-Cd oraz Ni-MH najłatwiej kupimy w rozmiarze AA (popularne paluszki) i napięciu nieco poniżej 1,4 V na ogniwo. Niestety, taka wartość napięcia występuje jedynie przy pełnym naładowaniu. Jak już wiemy, prawie cała elektronika naszego minirobota zasilana będzie prądem o napięciu 5 V. Z tego wynika, że zastosowanie nawet czterech akumulatorów Ni-Cd czy Ni-MH nie daje gwarancji, że układy będą zasilane odpowiednim napięciem. Może się więc zdarzyć, że zbyt duży spadek napięcia przy zwiększonym wysiłku spowoduje zresetowanie się sterownika. Jeśli jednak nie dopuścimy do zbyt dużego rozładowania tego typu ogniwa, jest szansa, że poziom napięcia zasilającego będzie wystarczający. Bezpieczniej byłoby użyć pięciu akumulatorów wielkości paluszka i odpowiedniego stabilizatora (patrz część Redukcja zakłóceń). W takim wypadku możemy mieć jednak problem z wagą całego robota.

Niewątpliwą zaletą akumulatorów Ni-Cd oraz Ni-MH jest łatwy proces ładowania,

Groźne zakłócenia

Układy elektroniczne pobierają energię ze źródła zasilania impulsowo, stąd też pojawiać się będą pewne szybkozmienne wahania napięcia, zwane zakłóceniami. Ich głównym źródłem są impulsowe układy sterowania napędami PWM (również same silniki wprowadzają do układu spore zakłócenia). Zakłócenia są bardzo często przyczyną niepoprawnej pracy zastosowanego w robocie mikroprocesora. Jeśli więc ich nie zminimalizujemy, mikroprocesor będzie ciągle się zawieszał, resetował i działał w trudny do przewidzenia sposób. Aby tego uniknąć, musimy wykonać odpowiedni układ stabilizujący – więcej informacji w części Redukcja zakłóceń.



Wydajność prądowa akumulatorów Ni-Cd lub Ni-MH z pewnością wystarczy do wystereowania zaproponowanych przez Eksperta serwo-mechanizmów. Jeśli jednak zdecydujemy się zastosować mocniejsze silniki, napięcie zasilające może spaść poniżej dopuszczalnej wartości

mają one zwykle małą wydajność prądową oraz często zawierają wbudowaną elektronikę, znacznie utrudniającą ich podłączenie. Ogniwa Li-Ion

i Li-Pol są dużo lepsze od Ni-Cd i Ni-MH, a w dodatku można je obciążać znacznymi prądami. Ich typowe napięcie robocze stanowi wielokrotność wartości 3,7 V. W konstrukcjach minisumo najczęściej wykorzystuje się modele o napięciu 7,4 V i pojemności 1200 mAh

Ogniwa litowe

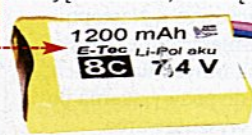
Akumulatory Li-Ion i Li-Pol możemy zakupić jedynie w sklepach modelarskich lub sklepach internetowych. Ekspert nie zaleca wykorzystywać baterii przeznaczonych do telefonów, gdyż

i Li-Pol są dużo lepsze od Ni-Cd i Ni-MH, a w dodatku można je obciążać znacznymi prądami. Ich typowe napięcie robocze stanowi wielokrotność wartości 3,7 V. W konstrukcjach minisumo najczęściej wykorzystuje się modele o napięciu 7,4 V i pojemności 1200 mAh

(dwa ogniwa). Niestety, ten typ zasilania będzie nas sporo kosztował. Zakup samej baterii to wydatek około 100 złotych. A ponieważ proces ich ładowania jest dość

skomplikowany, ładowarka kosztuje niemal drugie tyle. Ale to nie jedyna wada takiego rozwiązania. W przypadku ogniwa litowych całkowite rozładowanie prowadzi do trwałego uszkodzenia (wartość graniczna, przy której należy bezwzględnie rozpocząć ładowanie, to 2,7 V na jedno ogniwo!). Jeśli zdecydujemy się jednak na luksusową litowo-polimerową wersję zasilania, niezbędne będzie użycie

w układzie robota miniaturowego stabilizatora redukującego napięcie z 7,4 V do 5 V



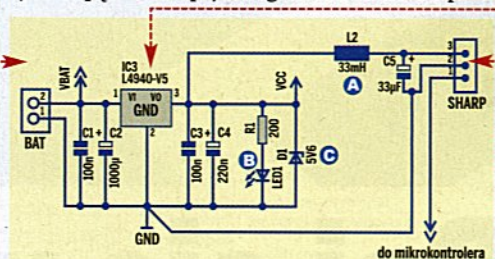
REDUKCJA ZAKŁÓCEŃ

Aby stłumić zakłócenia powstające w wyniku pracy silników i pozostałych układów elektronicznych, należy w zasilaczu silników zastosować w odpowiednich miejscach kondensatory oraz cewki.

Kondensator to element, który podobnie jak bateria magazynuje w sobie energię – tyle że zazwyczaj mało i na krótko. Jego podstawową zaletą jest to, że może on zmagazynowaną energię do układu oddać bardzo szybko (bateria jest przy kondensatorze wyjątkowo „powolna”). Dlatego też, jeśli użyjemy kondensatorów, w pierwszej chwili energia zostanie pobrana właśnie z nich, a gdy bateria zacznie przekazywać energię do układu, przy okazji będą ładowane puste kondensatory. Do poprawy jakości zasilacza stosuje się kondensatory ceramiczne oraz elektrolityczne.

Układ stabilizatora

W celu zredukowania napięcia akumulatora do 5 V najlepiej zastosować stabilizator L4940V5. Na jego wyjściu uzyskujemy stabilne napięcie (oznaczone jako VCC), którym zasilane będą wszystkie układy z wyjątkiem mostków sterujących napędami i optycznego dalmierza Sharp.



Do silników doprowadzamy napięcie bezpośrednio z akumulatora (VBAT), a do dalmierza, który wprowadza do układu spore zakłócenia, bezpośrednio ze stabilizatora (w tym wypadku stosujemy również dodatkowe elementy A). Tuż za stabilizatorem warto umieścić diodę LED B sygnalizującą działanie układu oraz diodę Zenera C, która zabezpiecza pozostałe moduły na wypadek uszkodzenia zasilacza.

Uwaga! W wersji z czterema „paluszkami” pomijamy układ stabilizatora L4940V5, łącząc z sobą nóżki 1 i 3. Układ zasilacza i stabilizatora zajmuje sporą część płytki drukowanej. O tym, jak wykonać taką płytkę i zmontować całą elektronikę, przeczytamy w następnym numerze Eksperta

Spis elementów

- **Rezystory:**
 - 1x 220
- **Kondensatory:**
 - 3x 100 nF, 1x 22 uF,
 - 1x 1000 uF, 1x 33 uF
- **Cewki:**
 - 1x 33 mH
- **Elementy półprzewodnikowe:**
 - 8x dioda HER108
 - układ L298
 - dioda LED świecąca
 - dioda Zenera 5V61W3
 - układ L4940V5

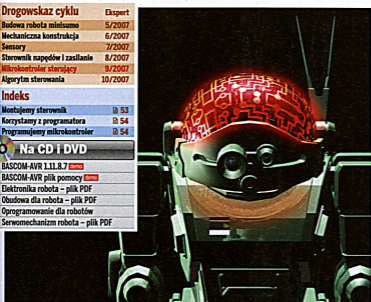
Drogowskaz cyklu	Ekspert
Budowa robota minisumo	5/2007
Mechaniczna konstrukcja	6/2007
Sensory	7/2007
Sterownik napędów i zasilanie	8/2007
Mikrokontroler sterujący	9/2007
Algorytm sterowania	10/2007

Indeks

Montujemy sterownik	53
Korzystamy z programatora	54
Programujemy mikrokontroler	54

Na CD i DVD

BASCOM-AVR 1.11.8.7	CD
BASCOM-AVR plik pomocy	CD
Elektronika robota – plik PDF	
Obwód dla robota – plik PDF	
Programowanie dla robotów	
Serwomechanizm robota – plik PDF	



Mózg robota

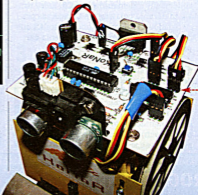
Centralny układ sterujący jest w robocie najważniejszy. To on decyduje o tym, czy całość będzie działać poprawnie

W tej części cyklu artykułów poświęconych budowie robota minisumo zajmujemy się mikroprocesorowym sterownikiem stanowiącym serce mechanicznego wojownika. Dzięki niemu nasza dwukółkowa konstrukcja stanie się prawdziwym zawodnikiem minisumo zdolnym do walki na ringu.

1 Po co sterownik

Do zadań sterownika należy gromadzenie oraz przetwarzanie danych sensorycznych, realizacja algorytmu sterowania oraz generowanie sygnałów dla układu napędowego.

Najpierw dowiemy się, jakiego układu elektronicznego najlepiej użyć jako sterownika. Następnie Ekspert pokaże, jak zbudowany jest sterownik. Na koniec nauczymy się go programować oraz poznamy podstawy obsługi środowiska BASCOM AVR.



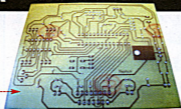
UKŁAD STEROWNIKA

Najistotniejszym elementem sterownika jest mikrokontroler. W robocie przedstawianym przez Eksperta zastosowano popularny, tani i niewielki mikrokontroler z rodziny AVR – Atmega48 firmy Atmel. Wyposażony jest on w 4 KB nieulotnej pamięci Flash (zapisane w niej informacje nie są tracone po odłączeniu zasilania). Do dyspozycji mamy także 512 bajtów pamięci RAM, którą wykorzystujemy do przechowywania danych operacyjnych.

Program dla mikrokontrolera opracujemy za pomocą dedykowanego środowiska programistycznego, działającego na zwykłym pecie. Gotowy program po skompilowaniu do postaci kodu binarnego rozumianego przez mikrokontroler umieścimy w pamięci Flash za pomocą programatora (patrz część Programator).

Montujemy sterownik i inne elementy na płycie drukowanej

W celu uproszczenia budowy części elektronicznej robota wszystkie układy pomocnicze opisane w poprzednich artykułach (wraz z obwodami mikrokontrolera) Ekspert zaleca umieścić na jednej płycie drukowa-



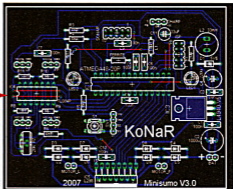
nej. Na krążku DVD/CD dołączonym do Eksperta znajdziemy PDF z opisem wykonania takiej płytki domowymi metodami.

Sterownik został wyposażony w złącza służące do podłączenia komponentów zewnętrznych, takich jak czujniki, silnik i akumulator. Ich opis i przeznaczenie znajdziemy w tabeli na następnej stronie.

Całą płytkę drukowaną możemy zmontować, korzystając ze schematu (w postaci pliku graficznego znajdziemy go również na płycie DVD/CD dołączonej do Eksperta). Do poprawnej pracy sterownika niezbędne jest użycie kilku dodatkowych, wcześniej nieopisanych, elementów (ich pełny spis znajdziemy na końcu artykułu).

1 Rezystory R3 i R4 znajdujące się przy złączu SONAR mają za zadanie podciągnięcie napięcia zasilania do linii SDA i SCL sonaru. Są one niezbędne do prawidłowej pracy magistrali I2C.

2 Odczyt pomiarów z dalmierza optycznego Sharp realizowany jest z wykorzystaniem przetwornika analogowo-cyfrowego zintegrowanego z mikrokontrolerem (zadaniem przetwornika jest zamiana wartości analogowej mierzonego sygnału na jej re-

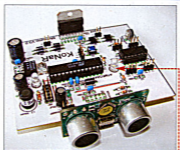


1 Złącza sterownika

Nazwa złącza	Opis złącza
BAT	złącze dwupinowe, służy do podłączenia pakietu akumulatora
LT, PT, PP, LP	cztery złącza trójpinowe, służą do podłączenia czterech czujników koloru (białej linii), LT – lewy tył, PT – prawy tył, PP – prawy przód, LP – lewy przód
SV1	złącze trójpinowe, do podłączenia czujnika zsterownego
SHARP	złącze trójpinowe, do podłączenia czujnika optycznego Sharp GP2D02
SONAR	złącze dziesięcpinowe, do podłączenia czujnika ultradźwiękowego Mobot-US
ISP	złącze dziesięcpinowe, do podłączenia programatora na czas programowania
MOTOR_P	złącze dwupinowe do podłączenia prawego silnika
MOTOR_L	złącze dwupinowe do podłączenia lewego silnika

prezentację cyfrową). Przetwornik ten jest zasilany przez filtr dolnoprzepustowy LC zbudowany z elementów L1 i C6, który usuwa zakłócenia pojawiające się podczas pracy układów cyfrowych. Kondensator C5 dodatkowo filtruje napięcie odniesienia dla przetwornika.

3 Koralik ferrytowy, oznaczony na schemacie jako FERRITE, tłumí zakłócenia pojawiające się od strony masy (można go znaleźć w starych komponentach komputerowych). W razie problemów z jego



Gotowa płytka drukowana zawiera całą elektronikę niezbędną do poprawnego działania robota minisumo

zdobyciem można go pominąć, zastępując zwykłą zwórką z drutu.

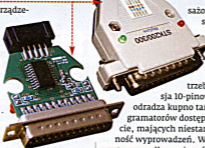
4 Elementy C7, C8, C9 i C10 to kondensatory odsprężające.

5 Dioda świecąca LED2 jest elementem pomocniczym, sterowanym bezpośrednio przez mikrokontroler. Płynący przez nią prąd ogranicza rezystor R2. Można ją wykorzystać w dowolny sposób, według własnego uznania – przykładowo do sygnalizacji wykrycia przeciwnika przez robota.

6 Rezystor R5 podciąga wejście RESET mikrokontrolera do zasilania.

PROGRAMATOR

Programator jest urządzeniem służącym do programowania wewnętrznej pamięci mikrokontrolera. Ekspert proponuje zastosować popularny programator stk200 – charakteryzuje się on prostą konstrukcją, niską ceną oraz zapewnianym wysokim poziomem bezpieczeństwa. Urządzenie to należy podłączyć do portu LPT naszego komputera oraz do złącza ISP na płytce drukowanej ze sterownikiem.



Skąd wziąć programator stk200? Można kupić gotowe urządzenie (cena około 25 złotych) lub wykonać go samodzielnie (o tym, jak to zrobić, przeczytamy na stronie 6). Przy zakupie gotowego programatora należy zwrócić uwagę na standard złącza, w jakie jest on wypo-

żony. Firma Atmel stosuje dwa standardowe złącze 6- lub 10-pinowe. Do programowania naszego robota potrzebna będzie wersja 10-pinowa. Ekspert odradza kupno tanich wersji programatorów dostępnych w internecie, mających niestandardową kolejność wyprowadzeń. W przypadku odwrotnego podłączenia taśmy lub podłączenia do standardowego złącza można sterownik doprowadzić do bardzo poważnych uszkodzeń – w takim wypadku konieczna będzie wymiana układu.

MOSI	1	2	VDD
LED	3	4	GND
RST	5	6	GND
SCK	7	8	GND
MISO	9	10	GND

Ekspert radzi

Nowe komputery, w tym laptopy, często nie są wyposażone w złącze LPT. W takim wypadku musimy zaopatrzyć się w programator USB. Jest on niestety droższy od wersji LPT i nie zawsze współpracuje z opisanym przez Eksperta środowiskiem programistycznym – do programowania warto więc wtedy wykorzystać środowiska dedykowane do zastosowanego programatora.

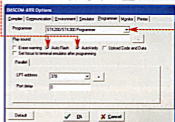
ŚRODOWISKO

W pierwszej części cyklu artykułów poświęconych budowie robota Ekspert wspomniał o środowisku programistycznym BASCOM AVR. Oprogramowanie to służy do edycji, kompilacji i programowania mikrokontrolerów AVR w systemie Windows i stanowi tak zwane środowisko IDE, czyli zintegrowane środowisko programistyczne. Dostępna za darmo wersja demonstracyjna tego oprogramowania pozwala na generowanie kodu o całkowitej długości ograniczonej do 4 KB, czyli dokładnie równej wielkości pamięci, w jaką wyposażony jest zastosowany w naszym robocie mikrokontroler.

Na płycie dołączonej do Eksperta, oprócz aplikacji BASCOM AVR, znajdziemy również polską wersję pomocy BASCOM Polish Language (nie jest ona w pełni aktualna, ale w większości wypadków może okazać się bardzo pomocna). Ekspert pokazuje, jak korzystać ze środowiska BASCOM AVR.

Instalacja i konfiguracja środowiska

Po zainstalowaniu środowisko należy odpowiednio skonfigurować. Jeśli będziemy programować za pomocą opisanego w części Programator urządzenia stk200, z menu **Options** wybieramy pozycję **Programmer** i na liście wyskakującej pozycję **STK200/STK300 Programmer**. Dodatkowo zaznaczamy opcję i ustawiamy adres portu LPT, do którego podłączony jest programator (w większości wypadków będzie to adres 378). Klikamy na przycisk **OK**.



! Uwaga!

Środowisko programistyczne BASCOM AVR do działania wymaga zainstalowanej w systemie dowolnej drukarki (wystarczy, że będzie zainstalowana, nie musi być nawet podłączona).

1 Kompletny spis elementów

Spis obejmuje tylko elementy znajdujące się na płycie drukowanej sterownika, nie obejmuje czujników koloru i zderzeniowych oraz włącznika zasilania i baterii. Należy też pamiętać o wtyczkach do złączy goldpin.

- **Rezystory:**
R1 200
R2 470
R3 R6 4,7k
R7 potencjometr 20k
R8-R11 1M
- **Kondensatory:**
C1, C3, C5- C10 100nF
C2 1000uF
C4 220uF
C11 33uF

PROGRAMISTYCZNE BASCOM AVR

Programowanie mikrokontrolera

Przed rozpoczęciem pisania algorytmu działania naszego robota powinniśmy poznać podstawy programowania mikrokontrolerów. Cała procedura jest stosunkowo prosta.

1 W pierwszej kolejności otwieramy w Bascomie przykładowy plik z programem (z płyty Eksperta), a następnie wciskamy klawisz **Q**. Spowoduje to skompilowanie programu do postaci binarnej.

2 Z menu **Program** wybieramy **Send to chip** i **Manual Program**, co spowoduje otwarcie okna **Widoczny jest w nim typ procesora** podłączonego do programatora oraz kod wynikowy programu.

Uwaga!

Jeśli na ekranie pojawi się powiadomienie **Could not identify chip with EEPROM**, oznacza to, że komunikacja z procesorem nie działa poprawnie. Należy wtedy sprawdzić podłączenie programatora oraz zasilanie układu (programator jest zasilany z docelowego urządzenia).

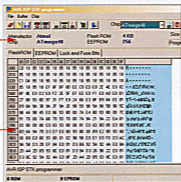
3 Przesłanie kodu programu do pamięci Flash mikrokontrolera wymaga wybrania z menu **Chip** polecenia **Auto program**. Pojawia się wtedy okno **pokazujące stan** procesu weryfikacji. Program jest najpierw wgrany do pamięci mikrokontrolera, a następnie odczytywany w celu weryfikacji.

4 Pomyślnie zakończony weryfikacji sygnałizowane jest pojawieniem się na dolnym pasku napisu **Verify OK**. Może zdarzyć się, że pomimo poprawnego działania interfejsu program nie zapisze się poprawnie – należy wtedy ponownie zaprogramować.

Testujemy robota

Wiemy już, w jaki sposób zaprogramować mikrokontroler. Możemy więc przejść do testowania naszego robota.

1 Do środowiska BASCOM AVR wczytujemy program **test_klaw.bas** (znajdziemy go na



płyce Eksperta). Jego zadaniem jest przetworzenie czujnika zderzeniowego oraz przycisku na płycie z elektroniką robota. Po zaprogramowaniu układu i wciśnięciu klawisza na płycie głównej lub czujnika zderzeniowego powinna zaświecić się dioda LED. Jeśli działa poprawnie, przechodzimy do kolejnych testów.

2 Wczytujemy teraz program **test_bw.bas**, programujemy mikrokontroler i sprawdzamy czujniki koloru. W tym celu stawiamy robota na czarnej nawierzchni i obserwujemy, czy świeci się dioda LED. Jeżeli

Ekspert radzi

Jeśli programujemy mikrokontroler pierwszy raz, należy ustawić kliknięciem myszy jego bity konfiguracyjne – Fusebit. W tym celu wybieramy zakładkę **Lock and Fuse Bits**, ustawiamy bity Fusebit High i Fusebit Low (klikamy na przycisk **Write FS**) i **Write FSH**. Nie należy zmieniać stanu bitów, których znaczenia nie znamy – możemy w ten sposób łatwo stracić komunikację z mikrokontrolerem, co spowoduje, że zastaniemy zmuszeni do zaprogramowania się w nowy układ.

Fusebits (F)	
Fusebit G	1 Divide Clock by 8 Disabled
Fusebit B	1 CLKOUT Output disabled
Fusebit KLAS87	1000T0 Inve. RC Osc. 8 MHz; Startup time
Fusebits High (H)	
Fusebit High K	1 PIVN PCB = RESET
Fusebit High J	1 debug/WIRE Disabled
Fusebit High I	0 SPI enabled
Fusebit High H	1 w/DTR enabled by w/DTR
Fusebit High G	1 Erase EEPROM when chip starts
Fusebit High DEF	101 Brown Out 2.7V

Programy testujące

Na płycie dołączonej do Eksperta znajdziemy programy testujące poszczególne elementy naszego robota. Zostały one napisane w języku MCS Basic, a ich kod źródłowy mają rozszerzenie **.bas**. Pamiętajmy, że za ich pomocą możemy przetestować tylko kompletny robot. Przed uruchomieniem testów powinniśmy się więc upewnić, że wszystkie elementy naszego wojownika są prawidłowo przyłączone we właściwych miejscach i w układzie sterownika niczego nie brakuje!

czujniki zostały wykonane prawidłowo, a dioda zaświeci się, musimy za pomocą potencjometru **ustawić próg** zadziałania czujnika. Kręcimy potencjometrem tak długo, aż dioda zgaśnie. Następnie przykładamy do każdego z czujników kawałek białej kartki i sprawdzamy, czy dioda się zapala.

3 Kolejnym testem jest sprawdzenie działania dalmierzy. W tym celu przygotowano dwa programy: **test_sharp.bas** i **test_sonar.bas**. Postępujemy podobnie jak w poprzednich testach. Pojawienie się przeszkody w odległości mniejszej niż 50 cm spowoduje zaświecenie się diody LED.

4 Na koniec testujemy napędy robota. Program **test_motor.bas** realizuje cykliczne wykonywanie ruchów w przód, w tył oraz skręt w prawo i w lewo. Każdy z nich trwa dokładnie jedną sekundę.

5 Jeżeli diagnostyka przebiegła pomyślnie, oznacza to, że podzespoły są sprawne. Możemy przejść do zaprogramowania robota do walki sumo. W tym celu wczytujemy program **sumo.bas** (**sumo bez sonaru.bas**, jeśli robot nie jest wyposażony w sonar, i **sumo bez sharpa.bas**, gdy nie ma dalmierza) i programujemy mikrokontroler. Pozostałe nam zainstalowanie przetwornika, z którym nasz robot zmierzy się na dohoyo.

Własny algorytm

Mamy już gotowego robota minimum, którego możemy wystawić do walki. Oprogramowanie robota zaprezentowane przez Eksperta jest jedynie propozycją – powinniśmy samodzielnie opracować algorytm walki. W ostatnim artykule z cyklu **Budujemy własnego robota** dowiemy się, jak zaprojektować taki algorytm. **PoWiR**

Autorzy (Mariusz Janiak, Janek Kędziński, Jakub Malewicz, Robert Muszyński oraz Karol Sydor) związani są z działającym przy Wydziale Elektroniki Politechniki Wrocławskiej Kółkiem Naukowym Robotyków „KoNaR”

Warto zaprzeć...

Adresy WWW:

- www.konar.ctp.wroc.pl/module.php?op=download&cmd=click&id=108
- www.zpcir.ctp.wroc.pl
- www.dioda.com.pl
- http://robot.bigbest.pl

niezbędnych do budowy płytki z elektroniką robota

- Półprzewodniki:**
 - LED1 Led 5mm biała
 - LED2 Led 5mm zielona
 - D1 dioda zenera 5,6 V
 - D2-D9 dioda HER108
 - IC1 ATmega48-20PI
 - IC3 LA940-5V
 - IC4 TL084P
 - IC5 L298N
- Różne**
 - L1 10 µH
 - L2 33 mH
 - FERRITE koralek ferrytowy
 - S1 mikrowłącznik
 - SHARP LP, PP, PT, LT – lista goldpin 3x1
 - SV1 – lista goldpin 4x1
 - ISP – podwójna lista goldpin 5x2
 - BAT, MOTOR_P, MOTOR_L – lista goldpin 2x1
 - SONAR – gniazdo goldpin żeńskie, kątowe 5x2 (montowane od strony druku)
 - Podstawki pod układy scalone 14 pin i 28 pin (wąska)

Indeks

Teoria sterowania	62
Trochę o sztucznej inteligencji	63
Układ sterowania robotem	63
Oprogramowanie mikrokontrolera	64

Na CD i DVD

Kompletne kody źródłowe

nazwa pliku
- plik znajduje się na krążku Eksperta

nazwa pliku
- plik w całości znajduje się na krążku, pokazany tu jest tylko jego fragment

BASCOM-AVR 1.11.8.7 demo

BASCOM-AVR plik pomocy freeware

Elektronika robota - PDF

Płytką drukowaną dla robota minisumo

Schemat modułów robota - PDF

Program w głowie

Fizyczna budowa robota to dopiero początek. Umysł, czyli program sterujący, jest najważniejszą bronią mechanicznego wojownika.

Z Ekspertem sami go stworzymy

W ostatniej części cyklu artykułów poświęconych budowie robota minisumo Ekspert przedstawi prosty algorytm sterowania naszym małym wojownikiem. Algorytm ten zapiszemy w języku programowania, zrozumiemy zasady działania algorytmu, Ekspert postara się przybliżyć niektóre zagadnienia z zakresu teorii sterowania i sztucznej inteligencji. Dodatkowo Ekspert opíše również inne stosowane architektury układów sterowania. Dzięki temu będziemy mogli sami eksperymentować ze zbudowanym robotem, tak by przekształcić go w niepokonanego zawodnika sumo.

KRÓTKO O STEROWANIU

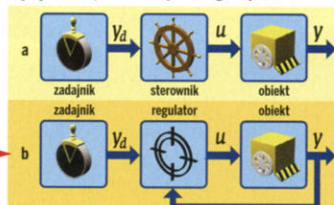
O tym, jak sterować różnymi obiektami, mówi teoria sterowania. Traktuje ona obiekt sterowania jako układ z zestawem wejść u i wyjść y .

Zadanie sterowania polega na takim oddziaływaniu na obiekt przez wejścia u , by uzyskać jego pożądane zachowanie, obserwowane zazwyczaj przez wyjścia y . By zrealizować to zadanie, wykorzystywana jest wiedza na temat zachowania obiektu, na przykład w postaci jego matematycznego modelu. Można też potraktować obiekt jako czarną skrzynkę i dopiero w trakcie sterowania identyfikować jej własności. W automatycznych układach sterowania zadanie sterowania jest realizowane za pomocą sterownika dołączonego do wejścia obiektu. Sterowanie może odbywać się w układzie otwartym lub zamkniętym.

W otwartym układzie sterowania wartości sygnałów podawanych na wejście u są wyznaczone jedynie na podstawie wartości zadanej y_d , bez wiedzy o aktualnym zachowa-

niu samego obiektu. W układzie zamkniętym pojawia się sprzężenie zwrotne, polegające na dostarczeniu do sterownika informacji o stanie wyjść y obiektu. Ze względu na własności praktycznie zawsze stosowane jest sterowanie ze sprzężeniem zwrotnym.

Umiejętne dobranie układu sterowania do obiektu i zadania sterowania jest naprawdę trudne. W robotyce opisane powyżej metody są najczęściej stosowane na niskim poziomie, na przykład do regulacji prędkości czy też położenia robota. Na wysokim poziomie przydatne będą metody sztucznej inteligencji.



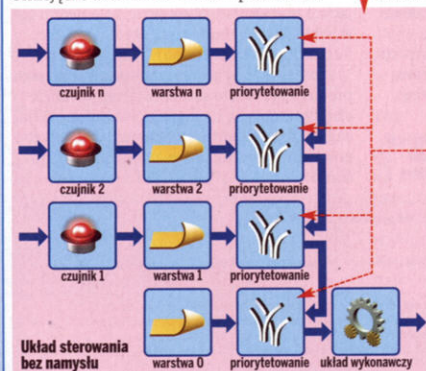
Typy architektury układów sterowania: a - w układzie otwartym, b - w układzie zamkniętym

TROCĘ O SZTUCZNEJ INTELIGENCJI

Robot-wojownik jest swego rodzaju aktorem występującym na scenie zwanej dohoy. Przeciwnik to nieznaną gracz, którego zachowania są trudne do przewidzenia, ruchy niemożliwe do odgadnięcia, a strategie znane tylko jego twórcy. Walka wymaga więc obserwacji, planowania i działania. Musimy także pamiętać, że każdy nasz ruch na pewno zostanie dostrzeżony przez przeciwnika i wpłynie na zmianę jego zachowania. Taki model opisu robota i otoczenia obejmuje skomplikowane zagadnienia planowania działań naszego wojownika, doboru strategii, pozyskiwania wiedzy i uczenia się.

Sterowanie bez namysłu

Znacznie prostsze jest tak zwane sterowanie odruchowe (bez namysłu). Polega ono na określeniu par typu bodziec-odruch, definiujących sposób działania całego systemu. Umiejętne dobranie odruchów pozwala na



zbudowanie robota, który mimo prostej konstrukcji układu sterowania będzie zachowywał się całkiem rozsądnie, a może nawet inteligentnie. Struktura układu sterowania oparta na tym schemacie wygląda następująco:

Dostarczające informacji z otoczenia sensory powiązane są z odpowiednimi warstwami sterownika, skojarzonymi z poszczególnymi reakcjami systemu. Warstwy działają współbieżnie, co oznacza, że wszystkie równocześnie generują poprawne sygnały sterujące. Z tego względu wymagany jest odpowiedni mechanizm priorytetywania (schematycznie przedstawiony na rysunku za pomocą zwrotnic), rozstrzygający, która reakcja jest ważniejsza w danym momencie. Warstwy wyższe mają zazwyczaj wyższy priorytet, dlatego umieszcza się w nich reakcje krytyczne. Zaletą układów sterowania tego typu jest ich względna prostota, łatwość implementacji, duża szybkość działania oraz krótki czas reakcji na zmiany otoczenia. Wadą zaś to, że działania nie opierają się na planie, nie ma w nich miejsca na uczenie się na błędach czy też na strategię pozwalającą zwyczaj wyższy priorytet, dlatego umieszcza się w nich reakcje krytyczne. Zaletą układów sterowania tego typu jest ich względna prostota, łatwość implementacji, duża szybkość działania oraz krótki czas reakcji na zmiany otoczenia. Wadą zaś to, że działania nie opierają się na planie, nie ma w nich miejsca na uczenie się na błędach czy też na strategię pozwalającą zwyczaj wyższy priorytet, dlatego umieszcza się w nich reakcje krytyczne.

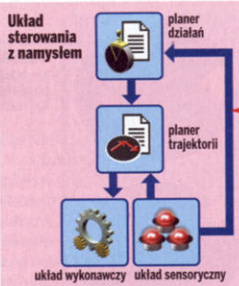
Sterowanie z namysłem

Innym rozwiązaniem jest sterowanie z namysłem. Realizowane jest ono w postaci procedury patrz-myśl-działaj. W tym przypadku rozbudowanie fazy myśli pozwala na zapewnienie naprawdę wysokiego poziomu inteligencji naszego robota. Układ sterowania tego typu przypomina w działaniu człowieka, który

podjmuje decyzje na podstawie wiedzy i planu, korzysta z metod dedukcji, logicznego myślenia, tworzy model otaczającego świata, potrafi ocenić, czy podejmowane akcje przyniosą pożądany skutek, wyciąga wnioski i na tej podstawie się uczy.

Układ sterowania z namysłem jest układem hierarchicznym. Sygnał sterujący, zanim dotrze do układu wykonawczego, musi przejść przez wszystkie warstwy układu sterowania. Poszczególne warstwy połączone są między sobą interfejsami, przez które komunikują się i wymieniają dane.

Planer działań, stanowiący najwyższą warstwę sterownika, jest swoistym centrum dowodzenia, dokonującym wyboru strategii, oceniającym skuteczność działania, a nawet czasami pozyskującym wiedzę. Bazując na posiadanej wiedzy i informacjach pochodzących z systemu sensorycznego, warstwa ta przygotowuje plan działania, który przekazywany jest warstwie niższej, realizującej planowanie trajektorii. Na podstawie planu działania planer trajektorii wyznacza sygnały sterujące przeznaczone dla układu wykonawczego. Opracowanie takiego typu sterownika jest niezwykle trudne i wymaga posiadania specjalistycznej wiedzy z różnych dziedzin. Dodatkowy koszt, jaki musimy ponieść w tym przypadku, wynika z gigantycznej złożoności obliczeniowej (pociągającej za sobą konieczność stosowania wydajnych procesorów z wielkimi zasobami pamięciowymi oraz dyskowymi) prowadzącej do długich czasów reakcji systemu.



UKŁAD STEROWANIA ROBOTEM

idealnym układem sterowania dla naszego robota wydaje się układ, który pozwoliłby nam stworzyć myślącego wojownika, kierującego się sprytem i przebiegłością, postępującego zgodnie z ustaloną strategią. Nasz robot nadzorowany jest jednak przez mikrokontroler o niewielkich zasobach i możliwościach, dlatego Ekspert zdecydował się na zastosowanie prostego, wymagającego małej mocy obliczeniowej rozwiązania, realizującego układ sterowania bez namysłu (patrz ramka Trochę o sztucznej inteligencji).

Najniższa warstwa układu sterowania - **szukaj 1** odpowiada za błądzenie robota po dohoy. Dzięki niej robot będzie penetrował ring w poszukiwaniu przeciwnika. Nad warstwą **szukaj 1** znajduje się warstwa odpowiedzialna za prowadzenie ataku - **atak 2**. Dalmierze oraz czujniki zderzeniowe dostarczają tej warstwie informacji o odległości do przeciwnika. Gdy przeciwnik nie znajduje się w polu widzenia, warstwa **atak** pozostaje nieaktywna.

Zadaniem najwyższej warstwy o nazwie **biała linia 3** jest czuwanie nad tym, by ro-

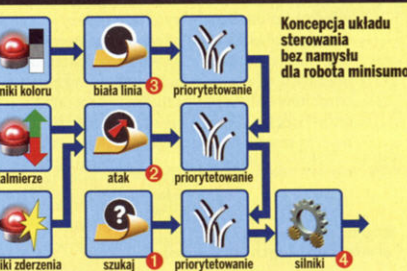
bot nie wyjechał z dohoy. Do wykrywania białej znajdującej się na jego kręwdzi białej linii służą czujniki koloru podłączone do wejścia opisywanej warstwy. Warstwa pozostaje w spoczynku do czasu wykrycia przez czujnik koloru brzożingu.

Uaktywnienie się jednej z wyższych warstw powoduje odcinanie warstw niższych i przejęcie przez nią w całości kontroli nad układem wykonawczym, oznaczonym jako **silniki 4**.

Zastosowany przez Eksperta układ sterowania działa na zasadzie odruchów. W układzie tym każdej akcji przyporządkowana jest pewna reakcja. Musimy zdefiniować minimalny zestaw odruchów pozwalający uzyskać w miarę inteligentne zachowanie robota.

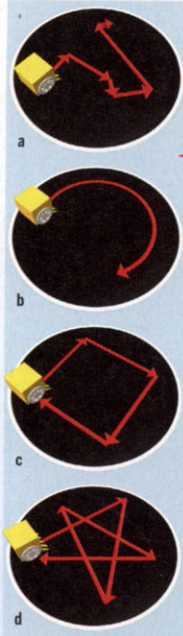
Poszukiwanie przeciwnika

Zaczynamy od warstwy **szukaj 1**. Robot w tym trybie powinien przemieszczać się w poszukiwaniu przeciwnika. Należy się spodziewać, że przeciwnik nie będzie czekał bezczynnie, aż go znajdziemy, i zapewne również



będzie krążył po ringu. W takiej sytuacji dobrać i wyjątkowo prostą metodą postępowania jest tak zwane błądzenie losowe, polegające na wykonywaniu ciągu losowych przemieszczeń i obrotów.

Przykładowy ruch tego typu może składać się z sekwencji: runda do przodu przez pewien czas, obrót o pewien kąt w losowym kierunku, dalsza jazda na wprost i tak dalej. Do realizacji takiego ruchu potrzebny jest jednak generator liczb losowych. Inna, wymagająca mniejszych nakładów obliczeniowych metoda może polegać na systematycznej jeździe po



Poszukiwanie przeciwnika:
a – błądzenie losowe, b – ruch po okręgu, c – ruch po kwadracie, d – ruch po gwiazdzie

okręgu, kwadracie, gwiazdzie – lub też innej dowolnej figurze. W trybie szukania nie ma sensu zapewniania dużej dokładności. Wszelkie niedokładności zwiększą tylko prawdopodobieństwo znalezienia przeciwnika.

Atak

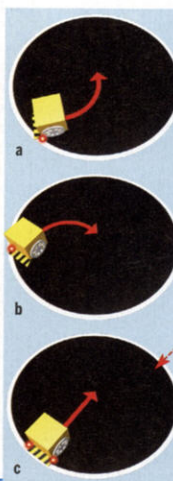
Warstwa ataku aktywuje się w momencie wykrycia przeciwnika przez dalmierz lub czujniki zderzeniowe. Gdy przeciwnika wykryły dalmierze, należy najpierw doprowadzić do zwania. Rozsądny wydaje się ruch cała naprzód. Należy jednak pamiętać, że szybki robot przy pełnej prędkości może nie być w stanie zatrzymać się na ringu po wykryciu

białej linii. Znajdzie się wówczas poza przepisowym polem walki i przegra pojedynek. Prędkość pościgu należy więc dobrać tak, by do tego nie doszło. Pełną moc powinniśmy włączyć w momencie, gdy przeciwnik znajduje się naprawdę blisko

lub gdy jesteśmy z nim w zwarciu, o czym poinformują nas czujniki zderzeniowe.

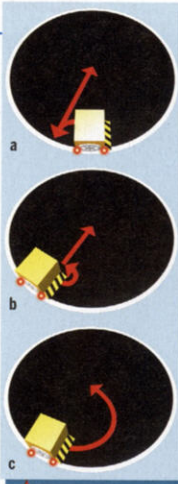
Uwaga na linię

Zostały nam jeszcze do opracowania odrucho dla warstwy biała_linia. Warstwa ta aktywuje się, gdy czujnik koloru wykryje brzeg ringu oznakowany białą linią. W takiej sytuacji robot powinien wykonać manewr pozwalający na oddalenie się na bezpieczną odległość od brzegu dohody najlepiej w kierunku środka ringu. Jest to najbezpieczniejsze miejsce na polu walki.



Rozpatrzmy sytuację, jakie mogą się zdarzyć podczas walki. Zbudowany przez nas robot wyposażony jest w cztery czujniki koloru, które rozmieszczone są symetrycznie w rogach jego obudowy. Najechanie na białą linię najczęściej uaktywnia w danej chwili tylko jeden z nich. W takiej sytuacji Ekspert proponuje wykonać jeden z manewrów

Sposób ucieczki z białej linii:
a – aktywne lewy czujnik koloru, b – aktywne prawy czujnik koloru,
c – aktywne oba przednie lub tylne czujniki koloru



Ucieczka z białej linii, gdy aktywne są przednie i tylne czujniki koloru: a – obrót przez koło i ucieczka, b – obrót w miejscu i ucieczka, c – ucieczka po ostrym łuku

Znacznie rzadziej może zdarzyć się, że aktywne są jednocześnie dwa czujniki koloru. W przypadku, gdy robot najechał na białą linię przodem lub tyłem, warto wykonać manewr. Bardzo rzadko natomiast dochodzi do sytuacji, gdy aktywne są jednocześnie przedni i tylny czujnik koloru – oba lewe lub oba prawe. W takiej sytuacji znajdziemy się raczej tylko wtedy, gdy będziemy pchani przez przeciwnika. Warto wtedy zastosować jeden z następujących manewrów: obrót przez koło (jedno koło zablokowane) i ucieczka, obrót w miejscu i ucieczka lub ucieczka po ciasnym łuku

Przedstawiony zestaw zachowań jest tylko propozycją. Warto samemu przeprowadzić eksperymenty, a następnie, wzorując się na sugestiach Eksperta, dobrać najbardziej odpowiednio według nas reakcje robota minisumo.

```

' Czy minal czas potrzebny na pomiary?
If Timermeasure = 0 Then

' Inicjalizacja pomiaru czasu
Timermeasure = Measure_period
    
```

```

' Odczyt sensorów
Stop Timer0 'na czas pomiarow zatrzymujemy Timer0
Ir_dist = Getadc(adc_channel) 'odczyt przetwornika AC
Us_dist = Us_read() 'odczyt sonaru
Call Us_measure(us_gain) 'inicjalizacja nowego pomiaru sonarem
Whiteline.0 = Not W1f1 'odczyt stanu czujnikow bialej linii
Whiteline.1 = Not W1r1
Whiteline.2 = Not W1r2
Whiteline.3 = Not W1r3
Switch = Sw1 'odczyt stanu przycisku na plytce
' sterownika
Touch = Sp2 'odczyt stanu czujnika dotykowego
Start Timer0 'koniec pomiarow - timer start
    
```

```

' Przetwarzanie danych pomiarowych (generacja zdarzen)
If Whiteline <> 0 Then
Call Finitestatemachine(event_whiteline)
End If
If Us_dist < Us_dist_thr Then 'fuzja sygnalow z sonaru i SHARPA
Call Finitestatemachine(event_us) 'najprostszsze rozwiazanie
End If
If Ir_dist < Ir_dist_thr Then 'traktujemy je rownowaznie
Call Finitestatemachine(event_ir)
End If
If Touch = 0 Then
Call Finitestatemachine(event_touch)
End If
If Switch = 0 Then
Call Finitestatemachine(event_switch)
End If
End If
    
```

```

' Czy minal czas akcji elementarnej?
If Timeraction = 0 Then
Incr K 'nastepny krok elementarny realizowanego odruchu
If K <= Length Then 'czy wykonano wszystkie ruchy elem.
Timeraction = Timetab(k)
Vel1 = Lookup(elmetab(k), Vel1tab)
Velr = Lookup(elmetab(k), Velr1tab)
Dir1 = Lookup(elmetab(k), Dir1tab)
Dirr = Lookup(elmetab(k), Dirr1tab)
Else
Call Finitestatemachine(event_endaction)
' TimerAction = TimeTab(1)
End If
End If
Loop 'koniec petli glownej
    
```

STRUKTURA OPROGRAMOWANIA

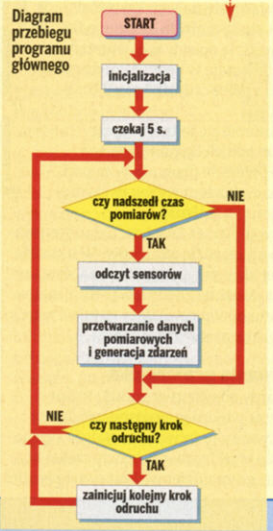
Znając architekturę układu sterowania oraz planowaną strategię działania, możemy przystąpić do stworzenia algorytmu sterowania robotem i niezbędnych do tego struktur danych. W tym celu należy uwzględnić kilka aspektów procesu sterowania: odczyt danych sensorycznych, zadawanie ruchu robota, realizację zaplanowanych odruchów i obsługę zmian stanu robota. Program podzielimy na dwie części: część główną oraz zestaw procedur obsługi przerwań.



W programie głównym umieszczone zostały procedury inicjalizacji sprzętu i struktur danych oraz nieskończona pętla główna, realizująca algorytm.

Procedury obsługi przerwań, skojarzone z układami peryferyjnymi mikrokontrolera, służą do obsługi specjalnych zdarzeń w systemie. Tutaj może być realizowany odczyt czujników pomiarowych, zadawanie prędkości ruchu, odmierzenie czasu. W przykładowej

Eksperta w przerwiach będziemy dokonywali jedynie pomiaru czasu i zadawali prędkości ruchu robota. Wszystkie pozostałe elementy zostaną umieszczone w programie głównym, którego diagram możemy zobaczyć na rysunku





```
Sub Finitestatemachine (byval Event As Byte)
Select Case State
Case State_search: 'szukaj
Case Event_touch:
Case Event_whiteline:
Case Event_attack:
Case Event_us_To_Event_ir:
Case Event_chase:
Case Event_init_action:
Case Event_prepare_attack:
Case Else:
Case Event_prepare_search:
End Select
```

```
Case State_attack: 'atak
Select Case Event
Case Event_touch:
Case Event_whiteline:
Case Event_prepare_attack:
Case Event_init_action:
Case Event_prepare_search:
Case Event_endaction:
End Select
```

```
Case State_whiteline: 'biała linia
Select Case Event
Case Event_whiteline:
Case Event_prepare_whiteline:
Case Event_init_action:
End Select
```

```
Case Event_endaction:
Case Event_prepare_search:
Case Event_init_action:
End Select
End Sub
```

1 Zaczniemy od zaimplementowania automatu skończonego, odwiedzającego strukturę naszego układu sterowania. W automacie wyróżnimy trzy stany, które odpowiadają trzem warstwom sterownika.

Umieszczone na rysunku okręgi – wierzchołki automatu – symbolizują stany systemu. Strzałki natomiast symbolizują zdarzenia. Do obsługi naszego automatu użyjemy funkcji **FiniteStateMachine(event)**. Jest ona wywoływana za każdym razem, gdy w systemie pojawi się zdarzenie. Zadaniem funkcji jest wyliczenie wartości zmiennej globalnej **state**, w której będziemy przechowywać informację o stanie, w jakim obecnie znajduje się automat. Jedynym pojawieniem się zdarzenia w systemie może powodować zmianę jej wartości. Nowa wartość funkcji zależy od aktualnej wartości oraz od zaobserwowanego zdarzenia. W przypadku zmiany stanu funkcja **FiniteStateMachine(event)** dodatkowo inicjuje realizację odruchu odpowiadającego nowemu stanowi.

2 Zastanówmy się teraz, w jaki sposób zapisać w postaci programu komputerowe-

```
'Tablica ruchow elementarnych lewego kola (byte)
VelTab: 'predkosci
Data 140, 250, 250, 60, 250, 250, 250, 250, 0
```

```
DirLTab: 'kierunek 1-przod, 0-tyl
Data 1, 1, 1, 1, 0, 1, 0, 1, 1
```

```
'Tablica ruchow elementarnych prawego kola (byte)
VelTab: 'predkosci
Data 140, 250, 60, 250, 250, 250, 250, 0, 250
```

```
DirRTab: 'kierunek 1-przod, 0-tyl
Data 1, 1, 1, 1, 0, 0, 1, 1, 1
```

go zestaw wcześniej zdefiniowanych odruchów. Ważne jest, by zapis nie ograniczał późniejszej rozbudowy czy zmiany repertuaru zachowań. Wszelkie modyfikacje powinny być możliwe do wprowadzenia w prosty i intuicyjny sposób.

Zaczynamy od zdefiniowania akcji elementarnych realizujących podstawowe ruchy: jazda prosto, po łuku, obrót w miejscu i obrót przez koło. Każdy ruch elementarny jest charakteryzowany przez cztery wartości: prędkość lewego koła, prędkość prawego koła, kierunek obrotów koła lewego, kierunek obrotów koła prawego. Wartości te dla poszczególnych akcji należy dobrać eksperymentalnie. Następnie zapisujemy je w programie w postaci tablic:

VelTab, VelRTab, DirLTab oraz DirRTab.

3 Teraz zdefiniowanie odruchu wymaga jedynie podania ciągu indeksów wskazujących na ruchy elementarne wraz z czasami ich trwania. Ciągi te przechowamy w odpowiednich tablicach.

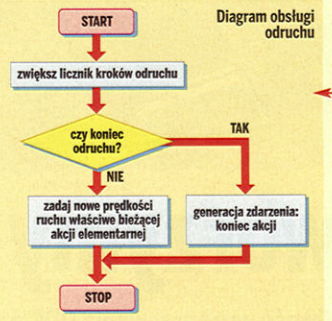
Na przykład informacje o odruchu realizowanym w stanie **szukaj** zapamiętujemy w tablicach **ElemTab_Search** (indeksy akcji elementarnych) i **TimeTab_Search** (czasy ich trwania). Liczba elementów tych tablic pamiętana jest jako **Length_Search**. Podobne tablice tworzymy dla pozostałych odruchów.

4 Wykonanie wybranego odruchu ma w przykładzie prezentowanym przez Eksperta miejsce w pętli głównej programu i polega na kolejnym przepisywaniu wartości z wybranych tablic do zmiennych **VelL, VelR, DirL, DirR**, gdzie przechowywane są odpowiednio aktualne prędkości lewego i prawego

```
'Odruch szukania (byte, word, byte)
ElemTab_search:
Data 0, 7
TimeTab_search:
Data 100%, 200%
Length_search:
Data 2
'Odruch poscigi (byte, word, byte)
ElemTab_chase:
Data 1
TimeTab_chase:
Data 500%
Length_chase:
Data 1
```

koła oraz kierunki ich ruchu. Proces ten możemy zaobserwować na rysunku. Na tym kończymy cykl dotyczący robota. Dzięki wiedzy nabytej podczas budowy własnego robota wielu Czytelników będzie mogło wyposażać swojego joyownika w inne rozwiązania niż prezentowane. Przeciwnie odpowiedzialności konstrukcji może zagwarantować jej zwycięstwo w robotycznych potyczkach. Dodatkowo

informacje i materiały znajdziemy na stronie: www.konar.ict.pwr.wroc.pl/kse Najbliższa okazja do zmierzenia się z przeciwnikami będzie miała miejsce w V Otwartych Zawodach Robotów Minisumo, które odbędą się na Politechnice Wrocławskiej w grudniu tego roku (więcej informacji na ten temat na witrynie www.konar.ict.pwr.wroc.pl).



```
If Timeraction = 0 Then
Incr K 'nastepny ruch elementarny realizowanego odruchu
If K <= Length Then 'czy wykonano wszystkie ruchy elementarne
Timeraction = Timetab(K)
VelL = Lookup(elemtab(K), VelLtab)
VelR = Lookup(elemtab(K), VelRtab)
DirL = Lookup(elemtab(K), DirLtab)
DirR = Lookup(elemtab(K), DirRtab)
Else
Call Finitestatemachine(event_endaction)
Timeraction = Timetab(1)
End If
End If
```

Autorzy (Mariusz Janiak, Janek Kędziński, Jakub Malewicz, Robert Muszyński oraz Karol Sydor) związani są z działającym przy Wydziale Elektroniki Politechniki Wrocławskiej Kółem Naukowym Robotyków „KoNaR”

1 Trudne terminy

- » **automat skończony** – (ang. finite state machine, FSM) to abstrakcyjny, matematyczny, iteracyjny model zachowania systemu dynamicznego oparty na tablicy dyskretnych przejść między jego kolejnymi stanami.
- » **sprzężenie zwrotne** – (ang. feedback) wzajemne oddziaływanie między dwoma układami polegające na uzależnieniu oddziaływania na pewne zjawisko od zmian zachodzących w tym zjawisku. Wyróżnia się dodatnie oraz ujemne sprzężenie zwrotne. Dodatnie stosowane jest powszechnie w układach generatorów drgań, ujemne zaś w układach sterowania.
- » **sztuczna inteligencja** – dziedzina badań naukowych informatyki, której celem jest konstruowanie maszyn i programów komputerowych zdolnych do realizacji wybranych funkcji umysłu i ludzkiej myślności nieopodporowanych się prostej numerycznej optymalizacji.
- » **teoria sterowania** – dział nauki i techniki zajmujący się zachowaniem układów dynamicznych w czasie. Sterowanie to celowe oddziaływanie na przebieg zachowania tych układów.