

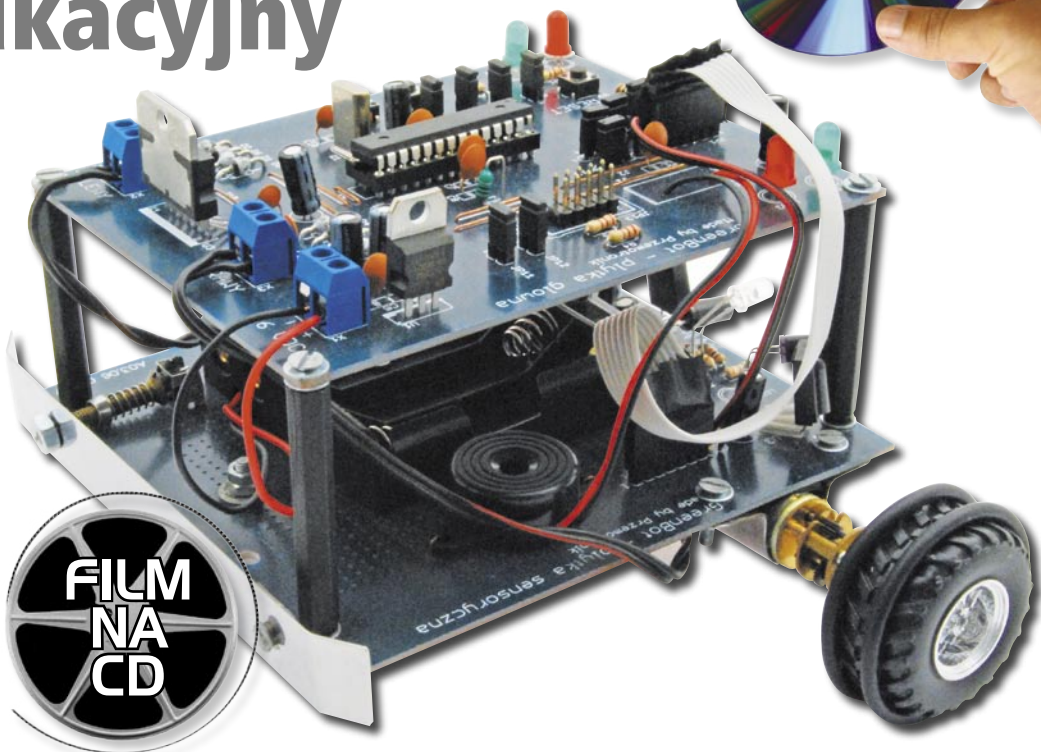
# GreenBot (1)

## Robot edukacyjny



Bogaty w możliwości rozbudowy i ilość różnorodnych funkcji, prosty robot skierowany do wszystkich chcących zająć się robotyką, lub mających pewne doświadczenie w budowie i programowaniu robotów. Projekt może być zarówno bazą do zabawy, jak i testowania zaawansowanych algorytmów: Line Follower, Light Follower, Maze Mouse i innych.

**Rekomendacje:** robot do samodzielnej budowy, dla pasjonatów robotyki na różnych poziomach zaawansowania. Polecany szczególnie jako projekt edukacyjny.



### Genesis

Liczba osób, chcących praktycznie zająć się robotyką, jest duża, na co wskazują liczne dyskusje internetowe. Padają w nich prośby o opisy przykładowych konstrukcji lub o ogólne porady. Już pobieżna analiza potrzeb w tej dziedzinie wykazuje, że największym zainteresowaniem cieszą się roboty, które można zaliczyć do kilku wyraźnie określonych grup.

Planując ogólną koncepcję budowy oraz projektując układy elektroniczne robota, postawiono na mnogość funkcji. W ten sposób skonstruowano urządzenie, które powinno zaspokoić potrzeby wielu entuzjastów. GreenBot to pewna „baza”, podstawa, która jest w gruncie rzeczy swego rodzaju „platformą ewaluacyjną” dla potrzeb robotyki. Konstrukcja jest otwarta i elastyczna, dzięki czemu rozbudowa urządzenia o dodatkowe funkcje jest dużo łatwiejsza, niż w innych tego typu robotach. Z pewnością dla bardzo wielu osób istotny będzie fakt, że

robota wykonano w oparciu o łatwo dostępne, popularne, tanie elementy.

Jedną z najistotniejszych zalet prezentowanego rozwiązania jest fakt, że może ono służyć do bardzo wielu testów, w różnych konfiguracjach sprzętowych oraz – co najważniejsze – z różnymi programami dla mikrokontrolera. Główne zastosowania robota to budowa:

- Nadzwyczaj popularnego w ostatnim czasie „Waldka Światłoluba”, inaczej zwanego „Light followerem”, to jest robota poruszającego się w kierunku najsilniejszego źródła światła.
- Cieszącego się coraz większą popularnością „Line followera”, to jest robota poruszającego się wzdłuż wyznaczonej na podłożu linii.
- Zdalnie sterowanego za pomocą pilota podczerwieni pojazdu wykorzystującego kodowanie RC5.
- Robota omijającego przeszkody a postępującego się prostym radarem podczerwieni oraz czujnikiem stykowym umieszczonymi „na pokładzie” robota.

Programy dla robota można pisać z użyciem dowolnego kompilatora przeznaczonego dla mikrokontrolerów AVR, jednak autor projektu, ze względu na popularność i prostotę wykorzystał Bascom AVR. Umożliwia on napisanie programu, jego symulację, a także załadowanie do pamięci mikrokontrolera. Wersja demonstracyjna Bascom-a, z ograniczeniem długości kodu do 4 kB, w zupełności zaspokaja podstawowe potrzeby, związane z wykorzystywaniem robota.

## AVT-5176

W ofercie AVT:  
AVT-5176A – płytka drukowana

### PODSTAWOWE PARAMETRY

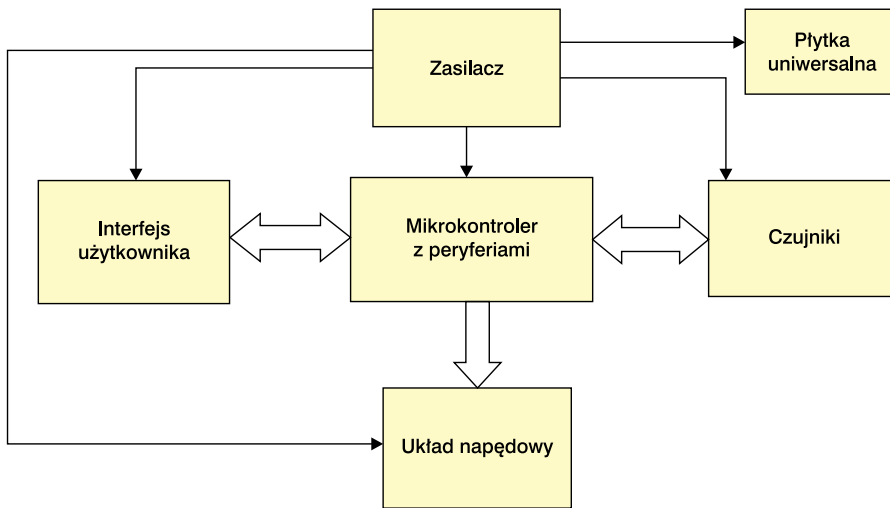
- Płytki jednostronne o wymiarach: 93,6×121,3 mm (główna) i 112,7×120,3 mm (sensoryczna)
- Zasilanie z baterii lub akumulatorów (6 VDC)
- Mikrokontroler ATmega8L (lub podobny)
- Konstrukcja otwarta do rozbudowy
- Napęd: 2 silniki DC lub przerobione serwomechanizmy
- Czujniki: 2× fototranzystory, 2× czujniki linii, 3× radary podczerwieni, 1× czujnik stykowy
- Interfejs użytkownika: 2× programowalne microswitch-e, przycisk reset, diody LED (2× zielona, 2× czerwona), 1× buzzer piezo
- Dostępne zestawy funkcji: light follower, line follower, robot zdalnie sterowany (RC5), omijanie przeszkód etc.
- Wbudowana płytka uniwersalna
- Możliwość wysterowania silników do 2 A
- Programowanie przez złącze ISP Kanda



### PROJEKTY POKREWNE

wymienione artykuły są w całości dostępne na CD

Tytuł artykułu	Nr EP/EdW	Kit
Pojazd zabawka	EP 1/2009	AVT-5165
Zabawka – programowany pojazd	EP 2-3/2002	AVT-5051
Robot (a właściwie jego ręka)	EP 7-9/1999	AVT-821
Zabawka – pojazd z radarem	EP 7/1998	AVT-429
ROBORobak	EP 7-8/2005	----



Rys. 1. Schemat blokowy GreenBota

## Opis schematu blokowego

Schemat blokowy układu elektronicznego przedstawiono na rys. 1. GreenBot składa się z 6 bloków, których krótki opis funkcjonalny przedstawiono niżej.

### Blok mikrokontrolera

Najważniejszym elementem GreenBota jest mikrokontroler wraz z podstawowymi peryferiami, spełniający rolę „mózgu” robota. Procesor zawiaduje całością, przechowuje i wykonuje program, podejmuje decyzje. Do procesora podłączono wszystkie pozostałe bloki. W tym momencie warto wspomnieć, że cały układ elektroniczny, (oprócz układu sensorycznego, silników i zasilania), zamontowano na górnej płytce głównej.

### Blok zasilania

Zadaniem tego bloku jest dostarczenie stabilizowanego napięcia zasilania do wszystkich elementów robota. Do prostego układu elektronicznego, dającego na wyjściu napięcie odpowiednie do zasilania całego robota, podłączono akumulatory, znajdujące się na dolnej płytce sensorycznej.

### Blok napędowy

Umożliwia on GreenBotowi poruszanie się. Zastosowana konstrukcja mechaniczna układu napędowego jest bardzo prosta: to typowa konfiguracja trójkołowa z dwoma kołami napędowymi (niezależne koła prawe i lewe oraz tylne koło Kastora, spełniające rolę trzeciego punktu podparcia). Oprócz silników (na płytce dolnej) składają się na niego także odpowiednie układy elektroniczne służące do bezpośredniego ich wysterowania.

### Blok interfejsu użytkownika

Prosty, a jednocześnie bardzo potrzebny moduł, dzięki któremu możliwe jest odczytywanie informacji o stanie robota podczas pracy. Zawiera również przyciski, które pozwalają w pewnym stopniu sterować robotem, np. przełączać zaprogramowane wcześniej opcje. Istnieje również możliwość rozbudowania tego bloku (podobnie, jak i pozostałych) przez wykorzystanie kolejnego modułu, jakim jest *Blok płytki uniwersalnej*.

### Blok płytki uniwersalnej

To rodzaj płytki uniwersalnej w centralnej części płytki dolnej. Umożliwia ona rozbudowę robota w dowolny sposób, poprzez wzbogacanie go o dodatkowe elementy stosownie do własnych potrzeb. Możliwe jest na przykład wlutowanie całkowicie nowych czujników, dodanie elementów wykonawczych lub kilku diod LED, wyświetlacza LCD itp. W celu ułatwienia tego zadania, skrajne ścieżki połączone zostały z szynami zasilania.

Wiele spośród wyprowadzeń mikrokontrolera podłączonych jest do reszty układu za pomocą goldpinów, na które można nakładać zworki (jumpery) i przewody połączeniowe, w celu dołączenia ich do procesora i układów peryferyjnych. Dzięki takiemu rozwiązaniu, doskonale znanemu chociażby z zestawów ewaluacyjnych dla mikrokontrolerów, otwierają się przed Czytelnikami ogromne możliwości wypróbowania własnych pomysłów.

### Blok sensoryczny

Jest to najbardziej rozbudowana i jednocześnie najbardziej elastyczna część układu. Zawiera ona wszystkie potrzebne czujniki oraz elementy dodatkowe potrzebne do ich prawidłowej pracy. Całość umieszczono na tzw. płytce sensorycznej, zamontowanej pod płytką główną. Sensoryka robota została zaprojektowana w sposób ergonomiczny i bardzo oszczędny, o czym świadczy np. zastosowanie tych samych czujników zarówno do odbierania promieniowania IR z pilota RC, jak i do konstrukcji radaru.

## Budowa płyty głównej

Schemat ideowy płytki głównej przedstawiono na rys. 2. Sercem układu jest bardzo popularny mikrokontroler AVR typu ATmega8L (U2) pracujący przy obniżonym w stosunku do wersji standardowej napięciu zasilania. Wybór taki został podyktowany warunkami zasilania robota. Jest to urządzenie mobilne, toteż do jego zasilania używane są baterie lub akumulatory. Po dłuższej eksploatacji rozładują się a napięcie spada poniżej nominalnego. Dzięki zastosowaniu wersji „L” mikrokontrolera moż-

liwa jest praca z napięciem zasilania 2,7 V, co wydłuża czas eksploatacji. Dodatkowo, w razie potrzeby możliwa jest wymiana kontrolera na inny, gdyż w urządzeniu zastosowano precyzyjną podstawkę pod procesor.

Obwód generatora sygnału zegarowego dla mikrokontrolera wykorzystuje oscylator kwarcowy Q1 oraz kondensatory C9 i C10. Jego zadaniem jest dostarczanie stabilnego sygnału o częstotliwości 8 MHz, taktującego mikrokontroler.

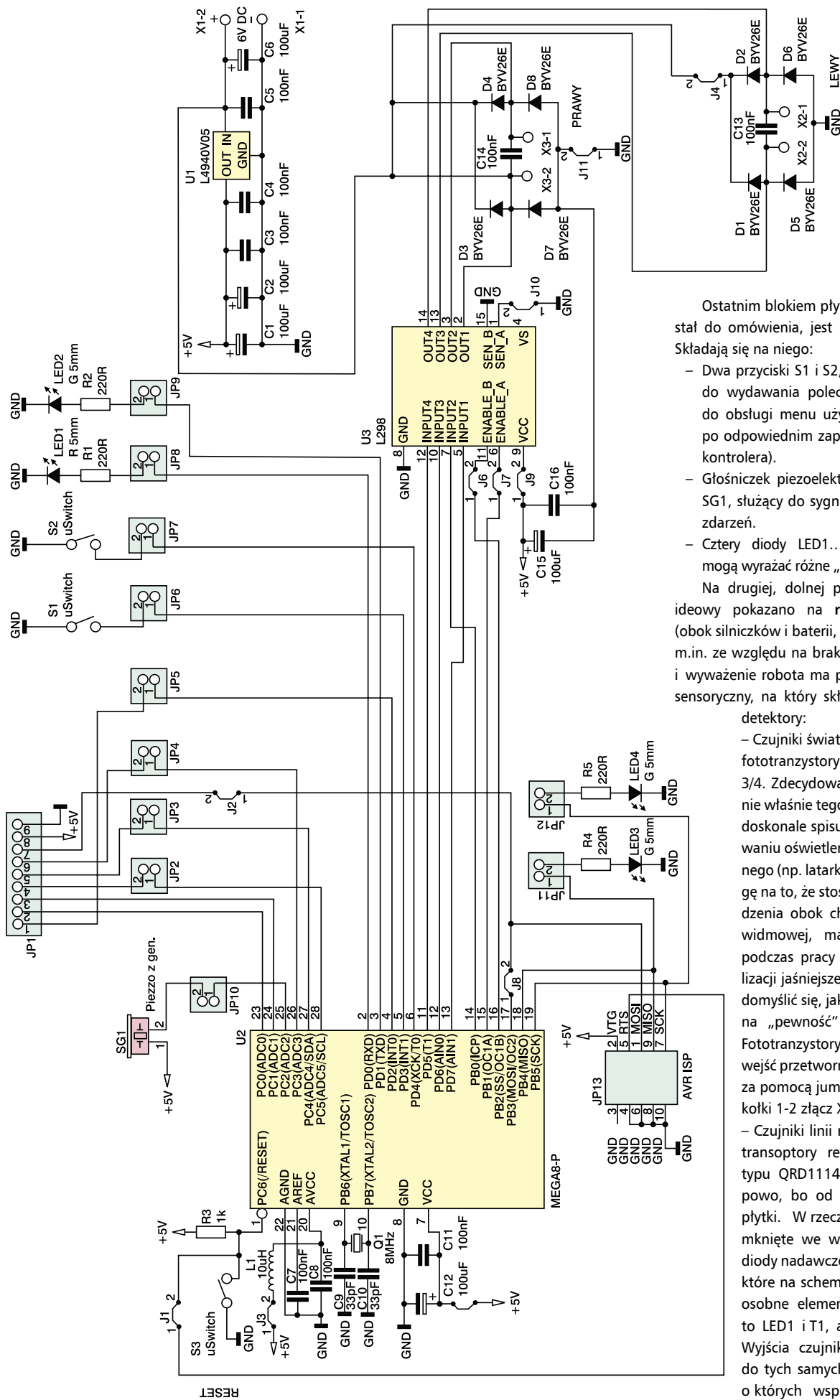
Elementy C8, L1, C7 tworzą filtr zasilania przetwornika analogowo-cyfrowego. Kondensator C7 zalecany jest przez producenta. Służy do filtrowania ewentualnych zakłóceń napięcia zasilającego AREF. Do wyprowadzenia zerowania mikrokontrolera podłączono przycisk S3 i rezystor R3. Dzięki temu, jeśli zajdzie taka potrzeba, można zrestartować procesor. To cecha bardzo użyteczna podczas eksperymentowania, gdy często nie do końca wiadomo, dlaczego program przestaje funkcjonować.

Kolejnym blokiem jest blok zasilacza. Jak pokazano na rys. 2, zawiera on złącze śrubowe X1, służące do podłączenia baterii (lub akumulatorów). Do zasilania GreenBot-a można zastosować cztery baterie alkaliczne LR6 (paluszki), lub jeszcze lepiej – zestaw akumulatorów. Nie jest polecane zamienne zastosowanie czterech akumulatorów NiMH lub NiCd w rozmiarze LR6, gdyż ich napięcie znamionowe jest niższe, niż baterii i wynosi 1,2 V (a nie 1,5 V, jak w baterii). Można natomiast zastosować zestawy modelarskie o napięciu 7,2 V lub 5 akumulatorów.

Napięcie z baterii, wstępnie odfiltrowane przez kondensatory C5 i C6, jest podawane na wejście stabilizatora U1. Jest to stabilizator typu LDO (*Low Drop Output*) zastosowany z tego powodu, że standardowe stabilizatory, jak np. LM7805, wymagają do poprawnej pracy dużej różnicy pomiędzy napięciem wejściowym, a wyjściowym. Dla układu LM7805 jest to minimum 2 V. Stabilizatory LDO potrzebują znacznie mniej – np. zastosowany w GreenBocie układ L4940V05 wymaga jedynie 0,45 V. Jak łatwo się domyślić, zastosowanie standardowego układu uniemożliwiłoby pełne wykorzystanie pojemności baterii.

Umieszczone w odpowiednich punktach płytki drukowanej kondensatory C1...C4, podobnie jak C11 i C12 tuż obok mikrokontrolera oraz C15 i C16 obok układu U3, dodatkowo filtrują napięcie zasilania zapobiegając przypadkowym oscylacjom.

Kolejną część układu to blok napędowy, którego głównym elementem jest popularny, scalony mostek H typu L298 (U3). Z mostkiem współpracuje osiem szybkich diod regeneracyjnych D1...D8. Zastosowano szybkie diody typu BYV26E o maksymalnym prądzie przewodzenia 1 A. Złącza X2 i X3 służą do przyłączenia lewego i prawego silniczka, a umieszczone tuż przy nich kondensatory ceramiczne C13 i C14 dodatkowo zmniejszają zakłócenia impulsowe pochodzące od silników napędzających.



Rys. 2. Schemat ideowy płyty głównej

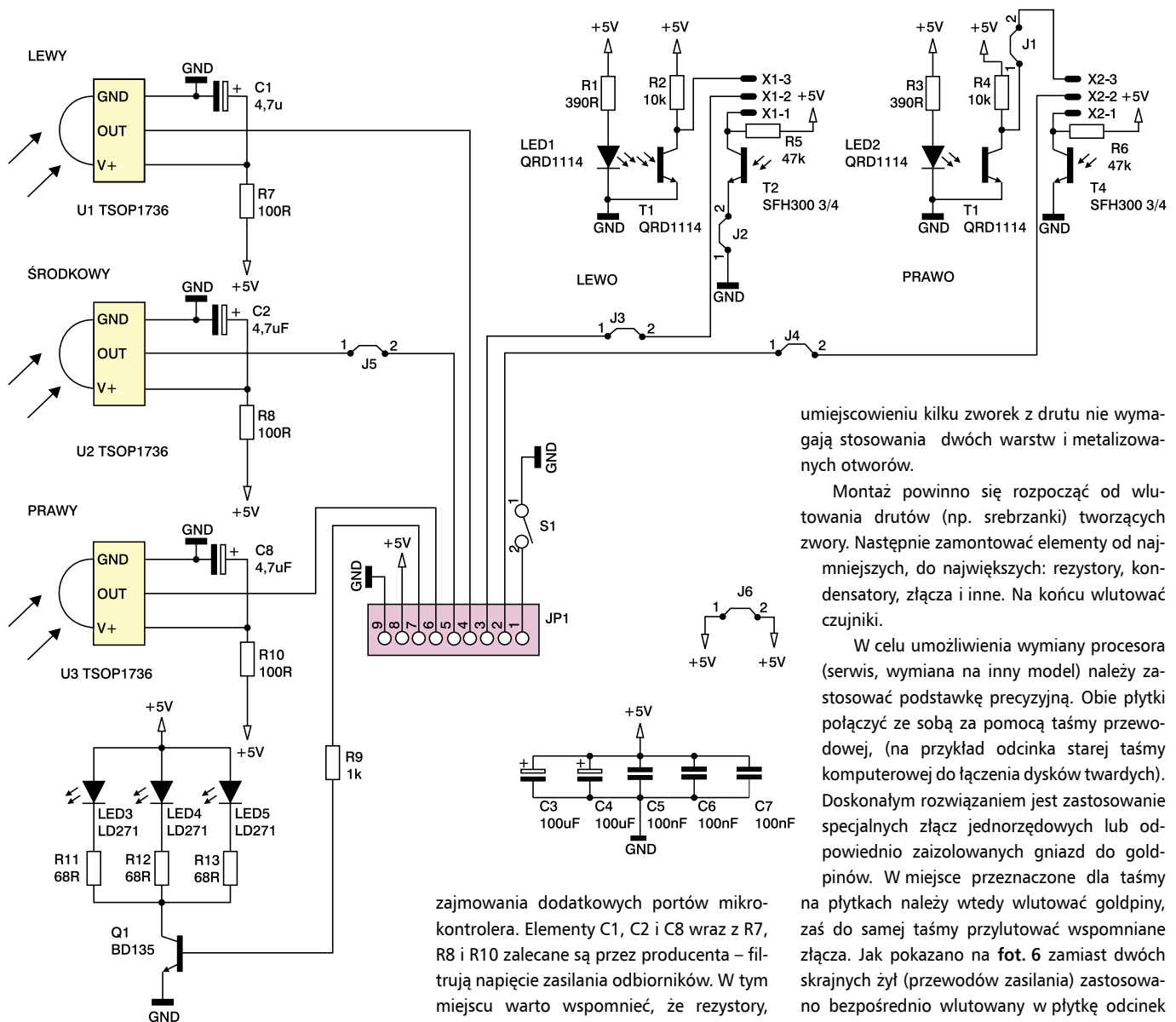
Ostatnim blokiem płytki głównej, jaki pozostał do omówienia, jest interfejs użytkownika. Składają się na niego:

- Dwa przyciski S1 i S2, które mogą posłużyć do wydawania poleceń robotowi, a także do obsługi menu użytkownika (oczywiście po odpowiednim zaprogramowaniu mikrokontrolera).
- Głośniczek piezoelektryczny z generatorem SG1, służący do sygnalizowania wybranych zdarzeń.
- Cztery diody LED1...LED4, które również mogą wyrażać różne „stany psychiki” robota.

Na drugiej, dolnej płytce, której schemat ideowy pokazano na rys. 3, swoje miejsce (obok silniczków i baterii, które znalazły się tutaj m.in. ze względu na brak przestrzeni na górze) i wyważenie robota ma przede wszystkim blok sensoryczny, na który składają się następujące detektory:

- Czujniki światła zewnętrznego: dwa fototranzystory T2 i T4, typu SFH300 3/4. Zdecydowano się na zastosowanie właśnie tego typu czujników, gdyż doskonale spisują się one przy wykrywaniu oświetlenia dziennego i sztucznego (np. latarki). Warto zwrócić uwagę na to, że stosunkowo wąski kąt widzenia obok charakterystyki czułości widmowej, ma kolosalne znaczenie podczas pracy robota w trybie lokalizacji jaśniejszego punktu. Nietrudno domyślić się, jak kąt optyczny wpływa na „pewność” zachowania robota. Fototranzystory można podłączyć do wejść przetwornika (ADC1 oraz ADC0) za pomocą jumperów, nałożonych na kołki 1-2 złącz X1 i X2.

- Czujniki linii na podłożu: tworzą je transistory refleksyjne (odbiciowe) typu QRD1114, zamocowane nietypowo, bo od strony ścieżek dolnej płytki. W rzeczywistości są to zamknięte we wspólnych obudowach diody nadawcze oraz fototranzystory, które na schemacie narysowano jako osobne elementy (jeden transopator to LED1 i T1, a drugi to LED2 i T3). Wyjścia czujników można dołączyć do tych samych wejść przetwornika, o których wspomniano w poprzed-



Rys. 3. Schemat ideowy płyty z sensorami

nim akapicie, zwiernając kołki 2-3 złącz X1 i X2.

- Czujniki podczerwieni: układy o szerokim zastosowaniu, oparte na popularnych kostkach typu TFMS5360. W układzie zastosowano ich zamienniki TSOP1736. Elementy te pozwolą robotowi odbierać paczki impulsów podczerwieni o częstotliwości nośnej 36 kHz, na przykład od pilota RC5 albo wewnętrzznego nadajnika z trzema diodami IR (LED3...LED5). Nadajniki (diody LD271) sterowane są za pomocą tranzystora Q1. Warto dodać, że ich dobór również nie był przypadkowy. Kąt połowicznej jasności wynosi 25°, co jest niewątpliwą zaletą, ponieważ zbyt wąski kąt świecenia powodowałby trudności z wykrywaniem wąskich i wysokich przeszkód, zaś zbyt szeroki – niepotrzebne „zazębianie się” zakresów detekcji. Zastosowanie tych samych detektorów pozwoliło nie tylko zaoszczędzić miejsce na płycie, ale także umożliwiło uniknięcie

zajmowania dodatkowych portów mikrokontrolera. Elementy C1, C2 i C8 wraz z R7, R8 i R10 zalecane są przez producenta – filtrują napięcie zasilania odbiorników. W tym miejscu warto wspomnieć, że rezystory, zwiększające stałą czasową filtrów RC, mają wartości nieco mniejsze, niż zalecane w katalogach, a to ze względu na możliwy duży spadek napięcia zasilania.

- Czujnik stykowy. Zamontowano go z tyłu dolnej płytki. Tworzy go zaledwie jeden element – pionowy microswitch S1. Pozwoli on uniknąć zderzeń z przeszkodami podczas cofania. Podłączono go do wejścia mikrokontrolera bez dodatkowych elementów, gdyż potrzebną histerezę zapewnia port mikrokontrolera, a ewentualne „odklócanie” przebiegu z nieuniknionych drgań zestyku można wyeliminować programowo.

Na dolnej płycie umieszczono wspomnianą wcześniej płytkę uniwersalną, która jest polem do realizacji własnych pomysłów. Dwie skrajne ścieżki poprowadzono równoległe do dłuższych boków płytki i podłączono do szyn zasilania. Dzięki temu zasilanie jest blisko miejsca, gdzie może być potrzebne.

### Montaż układu

Schemat montażowy płytki głównej pokazano na rys. 4, a płytki sensorycznej na rys. 5. Jak widać, są to płytki jednostronne, które dzięki

umiejscowieniu kilku zworek z drutu nie wymagają stosowania dwóch warstw i metalizowanych otworów.

Montaż powinno się rozpocząć od wlotowania drutów (np. srebrzanki) tworzących zwory. Następnie zamontować elementy od najmniejszych, do największych: rezystory, kondensatory, złącza i inne. Na końcu wlotować czujniki.

W celu umożliwienia wymiany procesora (serwis, wymiana na inny model) należy zastosować podstawkę precyzyjną. Obie płytki połączyć ze sobą za pomocą taśmy przewodowej, (na przykład odcinka starej taśmy komputerowej do łączenia dysków twardych). Doskonałym rozwiązaniem jest zastosowanie specjalnych złącz jednorzędowych lub odpowiednio zaizolowanych gniazd do goldpinów. W miejsce przeznaczonych dla taśmy na płytkach należy wtedy wlotować goldpiny, zaś do samej taśmy przylutować wspomniane złącza. Jak pokazano na fot. 6 zamiast dwóch skrajnych żył (przewodów zasilania) zastosowano bezpośrednio wlotowany w płytkę odcinek kabla o większej średnicy. Jest to konieczne, ponieważ cały układ sensoryczny, a głównie diody nadawcze podczerwieni, pobiera dość znaczny prąd zasilania. Jeśli zsumuje się on z prądem pobieranym przez dodatkowe układy podłączone na płytce uniwersalnej (główne przyłącze zasilania jest przecież wspólne), to może się okazać, że natężenie prądu będzie znacznie większe, niż dopuszczalne dla zastosowanej taśmy.

Jeśli stabilizator lub mostek będą się przegrzewać, to należy zastosować odpowiednie radiatory. Można wykorzystać np. pochodzące z demontażu urządzeń lub wykonane z blachy aluminiowej o grubości kilku milimetrów. Należy przy tym podkreślić, że silniki, które zostaną zastosowane, nie mogą pobierać (również przy zablokowanej mechanicznie osi) więcej niż 2 A. Taki jest maksymalny, dopuszczalny prąd stały obciążenia, który może przewodzić bez ryzyka uszkodzenia zastosowany układ scalony.

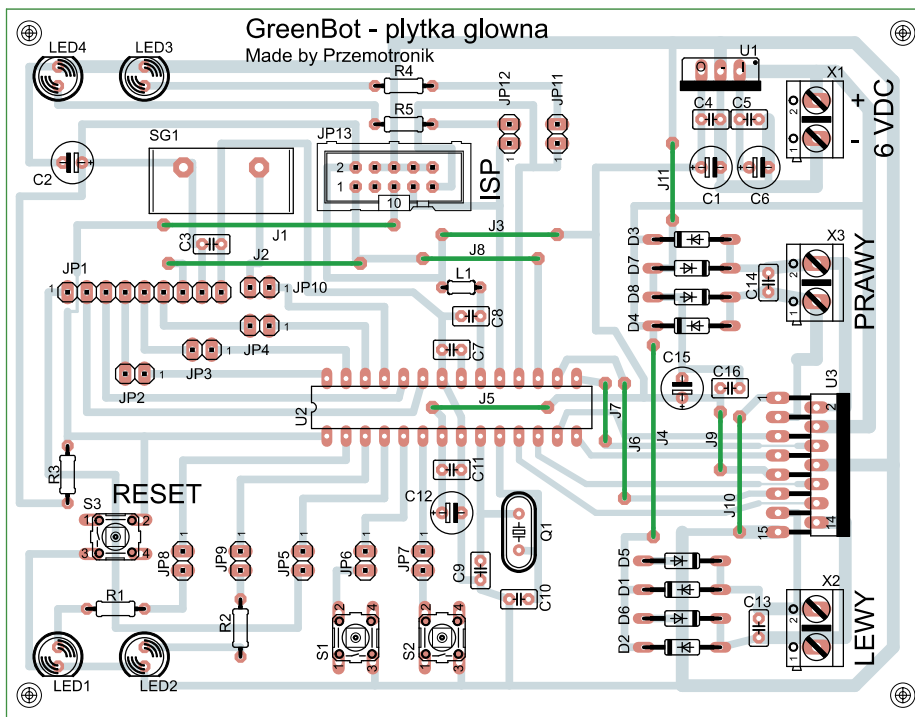
Następnym etapem montażu robota jest wlotowanie czujników podczerwieni U1...U3. Najpierw należy zamontować odbiorniki prawy i lewy, a potem środkowy. Testy prototypu wykazały wyraźną zależność między „jakością” detekcji przeszkód, a położeniem czujników. Czuj-

niki powinny być nieco wysunięte do przodu, przed linię padów lutowniczych (fot. 7). Dzięki takiemu ustawieniu, sensory lepiej zachowują się przy wykrywaniu przeszkód położonych pod różnymi kątami względem robota. Słupki dystansowe nie powinny zasłaniać podłużnych soczewek układów scalonych oraz soczewek diod. Odbiorniki należy zamontować na wysokości kilku milimetrów ponad powierzchnią płytki, na jak najdłuższych końcówkach. Umożliwi to późniejsze, dokładne wyregulowanie ich położenia.

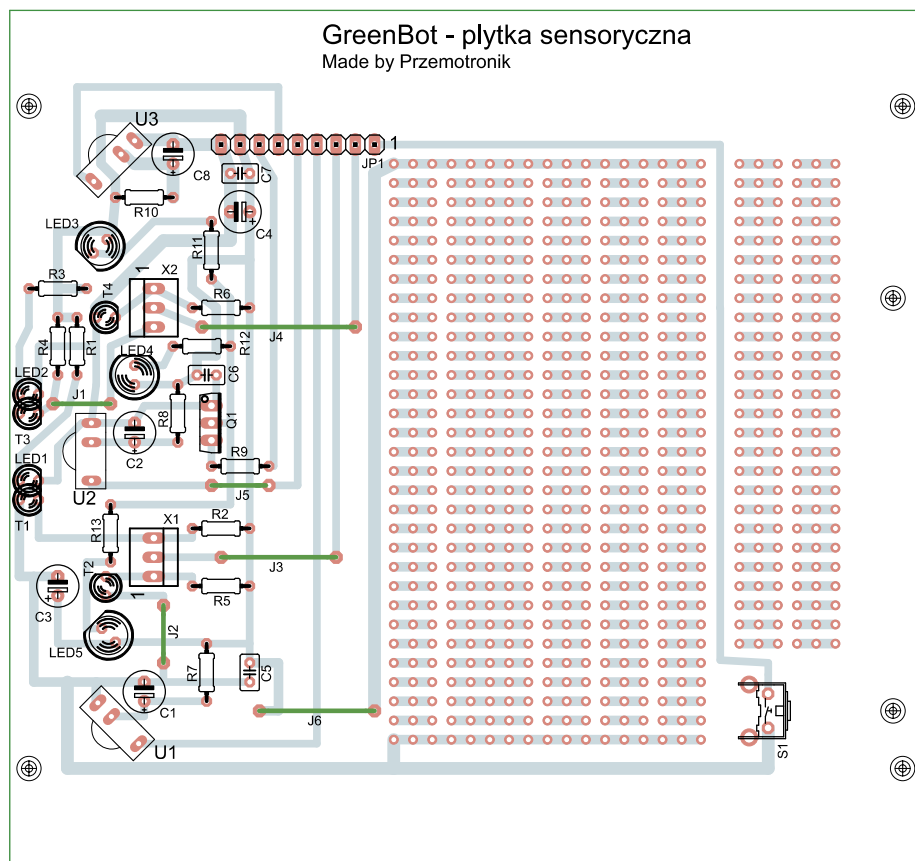
Diodom nadawczym trzeba wygiąć końcówki i wlutować je w podobnej pozycji, lecz kilka milimetrów wyżej. Emitery IR również przesunięte są przed linię swoich punktów lutowniczych tak, by ominęły słupki dystansujące. Środki obudów diod powinny znajdować się na wysokości nieco ponad środkiem obudowy odbiornika. Podczas testów wyszło na jaw, że bardzo czule odbiorniki muszą być dokładnie osłonięte z tyłu i z boków od promieniowania diod; w modelu prototypowym użyto w tym celu czarnej taśmy izolacyjnej. Podobnie należy osłonić same diody – w tym celu możliwe jest użycie ciemnej (najlepiej czarnej) rurki lub ostatecznie odcinka taśmy. Krawędź takiej osłony diody powinna kończyć się mniej więcej na poziomie czoła jej soczewki lub jeden, dwa milimetry przed nim. Przed wlutowaniem można nałożyć na końcówki czujników krótkie odcinki rurek termokurczliwych. Jest to ważne ze względu na bliskość tychże wyprowadzeń i innych, niez izolowanych elementów dolnej płytki. Dodatkowego komentarza wymaga natomiast montaż czujników, od niego zależy bowiem jakość późniejszej detekcji.

Zastosowane czujniki QRD1114 mają cztery wyprowadzenia. Ich montaż jest dosyć nietypowy. Transoptory na warstwie opisowej płytki reprezentuje obrys dwóch elementów: diody LED i fototranzystora. Ich końcówki położone od wnętrza płytki (czyli wyprowadzenia przy „ściętych” krawędziach obrysów) to odpowiednio: dla diody katoda, natomiast dla fototranzystora – kolektor. Widok zamontowanych czujników pokazano na fot. 8.

Wątpliwości może budzić fakt, iż katalogowa zalecana odległość pracy wynosi około 1,3 mm, natomiast w modelu odległość ta jest wyraźnie większa. Wynika to z prototypowego charakteru urządzenia na fotografiach. W finalnym układzie warto zamontować je niżej, tuż nad powierzchnią podłoża. Choć czujniki w większej odległości spełniają swoje zadanie, to jednak nie należy zapominać o ogromnym (nieliniowym) wpływie odległości od podłoża na ich czułość. W praktyce warto poczekać z montażem transoptorów aż do momentu, gdy zostaną zainstalowane silniki wraz z kołami oraz koło tylne, gdyż dopiero wtedy możliwe będzie jednoznaczne określenie potrzebnej odległości przy znanej już geometrii robota. Konieczny jest tutaj pewien kompromis. Nisko położone czujniki linii niejako wyznaczają warunki pracy robota, ograniczając je do równych, gładkich powierzchni. Bardzo ważne jest, aby krótko obciąć końców-



Rys. 4. Schemat montażowy płytki głównej



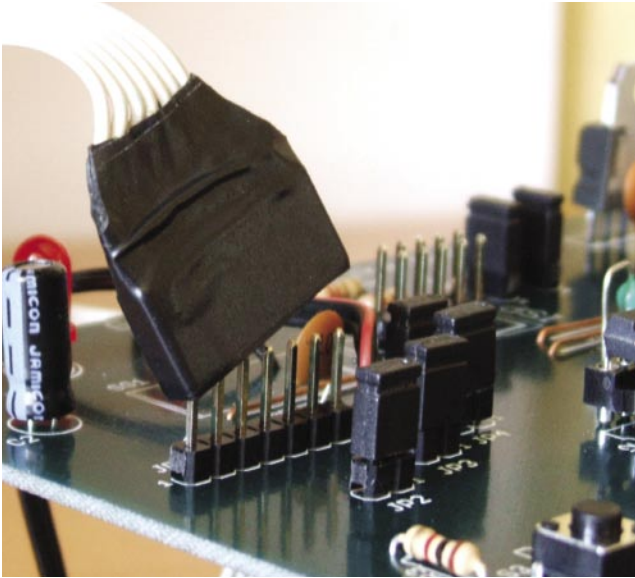
Rys. 5. Schemat montażowy płytki z sensorami

ki czujników linii, które wystawać będą nieco ponad górną powierzchnię płytki sensorycznej. Zapobiega to wspomnianym w poprzednim akapicie zwarciom z końcówkami czujników podcierwieni.

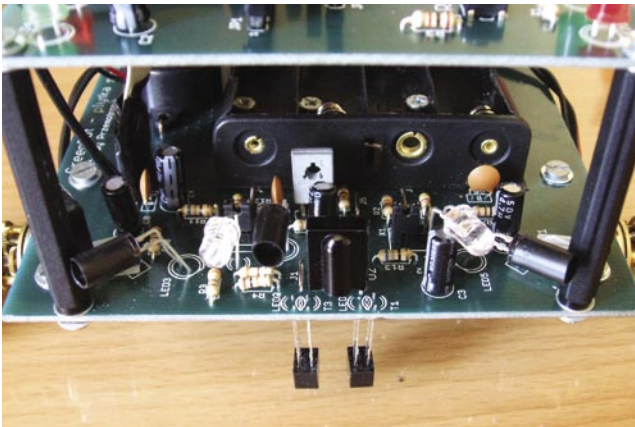
Z montażem fototranzystorów nie powinno być natomiast większych problemów. Trzeba je po prostu zamontować jak najwyżej (nie skracając końcówek), lekko pochylone w prawo i w lewo.

Należy jedynie uważać, by soczewki nie były całkowicie zasłonięte przez górną płytkę. Prawidłowy układ przednich czujników pokazano na fot. 7.

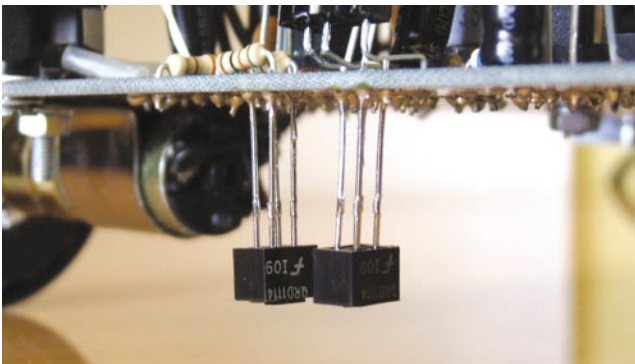
Przycisk (tylny czujnik zderzeniowy) montowany jest pionowo w płytce. Do pracy konieczny jest jeszcze zderzak. W modelu prototypowym wykonano go z kawałka odpowiednio ukształtowanej blachy, którą przymocowano następnie do dolnej płytki za pomocą kątownika, osadzonego na śru-



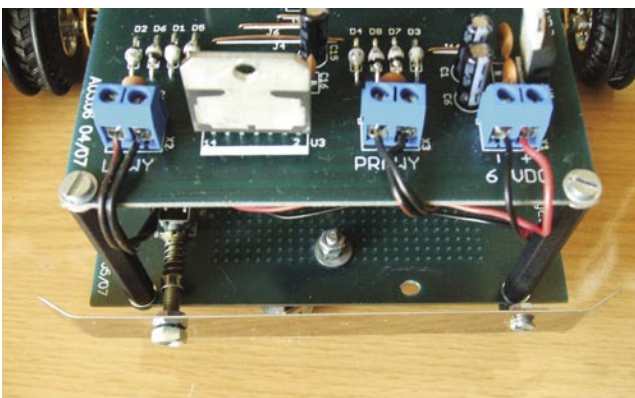
Fot. 6. Sposób wykonania wtyku kabla połączeniowego



Fot. 7. Sposób montażu czujników czolowych



Fot. 8. Sposób montażu czujników refleksyjnych



Fot. 9. Sposób wykonania zderzaka

bie, przytrzymującej tylny słupek dystansowy (fot. 9).

Kolejny element to koszyczki dla baterii (lub pakiet akumulatorów). Warto wypróbować różne własne koncepcje montażu zasilania. Może dzięki temu uda się zaoszczędzić miejsce na płytce. Brzęczyk piezoelektryczny można zamontować tak, jak będzie najwygodniej. W modelu z fotografii użyto dwóch druków wlotowych w płytce. Ostatnim etapem montażu jest umieszczenie w podstawce mikrokontrolera. Trzeba przy tym pamiętać o standardowych środkach ostrożności ESD.

Montaż silniczków zależy od tego, jakiego rodzaju napędy zostaną zastosowane.

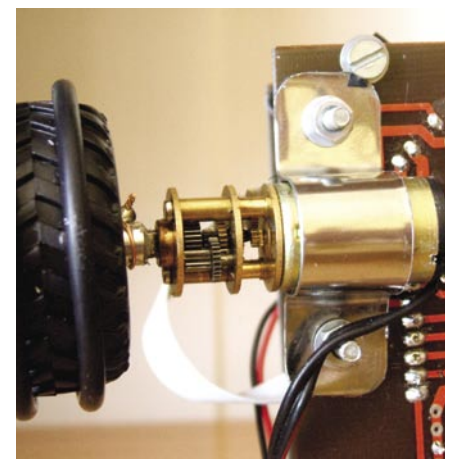
Najwygodniejsze do zamontowania są silniki o spłaszczonym przekroju obudowy. Małe napędy DC można znaleźć m.in. w różnych zabawkach, jednak często okazują się one zbyt słabe do poruszenia cięższej konstrukcji. W prototypie wykorzystano: dwa wysokiej jakości silniki z metalowymi przekładniami. Mają one dużo większy moment obrotowy, jednak stosunkowo małą prędkość obrotową, co mimo wszystko w wielu zastosowaniach okazuje się zaletą. Można również użyć przerobionych serwo-mechanizmów. Zmiana polega na usunięciu blokady mechanicznej (na jednym z kół przekładni znajduje się wypustka, którą trzeba delikatnie odpiłować), odłączeniu płytki z elektroniką sterującą oraz wyprowadzeniu przewodów bezpośrednio do silnika. Efektem tak wykonanej przeróbki jest doskonały napęd, bardzo wygodny w montażu i o dużym momencie obrotowym (patrz EP 1/2009, artykuł pt. „Pojazd integracyjny”).

W modelu zamontowano silniczki bliżej przedniej krawędzi płytki, za pomocą odpowiednio wygiętych blaszek, które przykręcono do dolnej płytki śrubami M3. Konieczne są podkładki, które zabezpieczą płytkę przed wyłamaniem śrub. Doskonałe efekty dają włożone pomiędzy silniczki a płytkę paski gumy: tłumią drgania, likwidują naprężenia i nie dopuszczają do zwarcia poprzez kontakt silniczków z płytką. Należy uważać, aby zbyt mocne dokręcenie silników na podkładkach gumowych nie spowodowało skrzywienia płytki i przez to pęknięcia ścieżek. Prawdopodobnie zamontowany silnik pokazano na fot. 10. Na specjalnie poszerzonych brzegach płytki nie zaznaczono otworów – należy wykonać zgodnie z własnymi potrzebami.

Kółka zastosowane w modelu pochodzą z zabawki. Połączenia zostały dodatkowo wzmocnione za pomocą kleju oraz cienkich druków, które owinięto wokół tulejek kółek i zlutowano. Ponieważ solidne, precyzyjne osadzenie kół na wałkach napędów jest koniecznością, do konstrukcji robota również doskonale nadają się gotowe, ogumione koła o średnicy kilkudziesięciu milimetrów, dostępne w sklepach modelarskich.

Ostatnim elementem, który pozostaje do zamontowania na płytce dolnej, jest kółko podporowe (Kastora). W roli tego elementu można użyć rolki w metalowej obejmie – takiej jakiej stosowane są w magnetofonach do prowadzenia taśmy (na obydwu brzegach kasety). Ze względu na wielkość i geometrię GreenBota typowe, małe rolki mogą nie zdać egzaminu i dlatego potrzebną obejmę wykonano samodzielnie ze stalowej blachy. Samo kółko kupiono w sklepie z częściami elektronicznymi. Ma ono średnicę nieco większą, niż zwykłe rolki (13 mm). Rolę osi rolki spełnia cienki wkręt. Trzeci otwór, wykonany w górnej jej części, służy do zamocowania całości na pionowej osi. Konieczne jest użycie podkładek (pomiędzy obejmą i płytką oraz po obu stronach kółka), a także zablokowanie nakrętki (np. drugą nakrętką lub lakierem), tak by nie odkręcała się podczas jazdy.

Szczególną uwagę należy zwrócić na to, czy podczas pełnego docelowego obciążenia bateriami i pozostałymi elementami konstrukcji, kółko



Fot. 10. Sposób mocowania silnika i koła

nie ociera o powierzchnię dolnej płytki. Schemat konstrukcji prostego koła Kastora pokazano na rys. 11a (z przodu) oraz rys. 11b (z boku). Zwieńczeniem prac montażowych będzie zmontowanie całości. W tym celu należy zastosować plastikowe lub metalowe, gwintowane słupki dystansowe. Ich długość zależy głównie od tego, jak zamontowane zostanie zasilanie oraz ile potrzeba miejsca pomiędzy płytkami. W prototypie zastosowano tulejki o długości 45 mm i gwincie M3. Pod śruby mocujące słupki dobrze jest dać podkładki, które nie dopuszczają do wylamania delikatnych rogów płytek.

Płytki drukowane zaprojektowano tak, by całą planowaną konstrukcję można było wykonać bez obudowy. Nic nie stoi jednak na przeszkodzie, aby zamontować płytki w inny sposób, umieszczając je w obudowie z tworzywa sztucznego. Ilość dostępnego w niej miejsca na dodatkowe moduły i rozbudowę mogłaby być w takim przypadku znacznie większa. Dobrze przemyślane ułożenie najczęściej używanych elementów obsługi (przycisków, diod LED, złącza ISP oraz ewentualnego głównego wyłącznika zasilania), a także większych części (baterii, napędów) może znacznie zwiększyć trwałość konstrukcji oraz wygodę jej użytkowania. Wyprowadzenie LEDów oraz mikroprzycisków na zewnątrz i odpowiednie ich opisanie dodatkowo zwiększyłyby estetykę urządzenia. Należy zwrócić baczną uwagę na położenie punktu ciężkości robota. Ogólna zasada jest taka, że im bliżej podłoża się on znajduje, tym lepiej. Dlatego ciężkie elementy, takie jak baterie zasilające powinny być zamontowane jak najniżej. Nieprzeważające wyważenie całości może spowodować wywracanie się robota podczas manewrów.

W czasie długotrwałych testów można zasilić robota przewodowo z zasilacza sieciowego o napięciu 6 V doprowadzając napięcie do złącza X1. I jeszcze jedna bardzo ważna uwaga związana pośrednio z montażem. Przy pierwszym uruchomieniu programu można natknąć się na problem niezgodności czasowej podczas realizacji bascomowych poleceń opóźnienia. Wiąże się to z koniecznością odpowiedniego zaprogramowania Fuse Bit-ów mikrokontrolera.

**Programowanie robota**

W tab. 1 przedstawiono opis funkcjonalny wyprowadzeń mikrokontrolera. W drugiej kolumnie pokazano nazwę danej końcówki w kontekście używanej funkcji a w ostatniej nazwy (alias), które będą używane zamiast oryginalnych.

Pierwsze programy umożliwiają sprawdzenie poprawności połączenia wszystkich elementów robota oraz komunikacji pomiędzy komputerem (programatorem), a mikrokontrolerem. Najprostszy program zawarty jest w pliku List01.bas. Po podaniu standardowych informacji dla kompilatora (deklaracja częstotliwości rezonatora kwarcowego oraz pliku rejestrów), następuje konfiguracja portów I/O. Następnym elementem programu jest

**Tab.1. Opis funkcji wyprowadzeń mikrokontrolera**

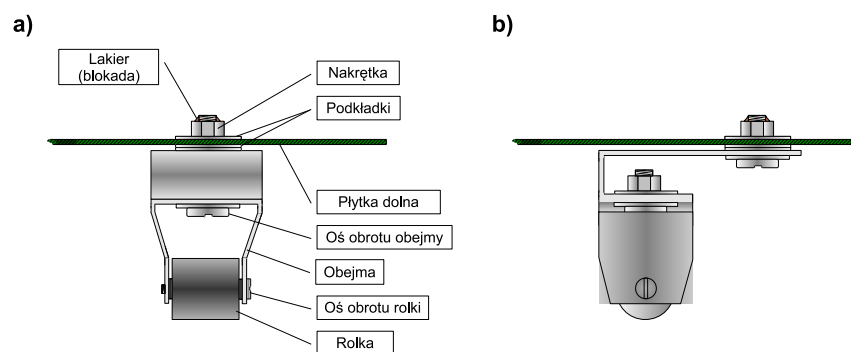
Port	Używana funkcja	Połączono z...	Kierunek	Alias w programie
PORTB0	Portb.0	INPUT2 mostka H	Wyjście	InputH_2
PORTB1	OC1A	ENABLE A mostka H	Wyjście	(używamy nazwy Pwm1a)
PORTB2	OC1B	ENABLE B mostka H	Wyjście	(używamy nazwy Pwm1b)
PORTB3	OC2/(MOSI)	LED nadawcze IR/(programator)	Wyjście	(używamy nazwy OCR2)
PORTB4	Portb.4/(MISO)	Dioda LED3/(programator)	Wyjście	LED3
PORTB5	Portb.5/(SCK)	Dioda LED4/(programator)	Wyjście	LED4
PORTB6	XTAL1	Oscylator kwarcowy	--	(nie używamy)
PORTB7	XTAL2	Oscylator kwarcowy	--	(nie używamy)
PORTC0	ADC0	Prawy fototranzystor/prawy czujnik linii	Wejście analogowe	(nie używamy)
PORTC1	ADC1	Lewy fototranzystor/lewy czujnik linii	Wejście analogowe	(nie używamy)
PORTC2	Portc.2	Brzęczyk piezozo	Wyjście	Buzzer
PORTC3	Pinc.3	Prawy czujnik podczerwieni	Wejście	PrawylR
PORTC4	Pinc.4	Środkowy czujnik podczerwieni	Wejście	SrodkowyIR
PORTC5	Pinc.5	Lewy czujnik podczerwieni	Wejście	LewylR
PORTC6	RESET	S3-przycisk reset/(programator)	Wejście reset	(nie używamy)
PORTD0	Portd.0	LED1	Wyjście	LED1
PORTD1	Portd.1	LED2	Wyjście	LED2
PORTD2	Pind.2/INT0	Czujnik zderzeniowy (tylny)	Wejście	Zderzak
PORTD3	Pind.3/INT1	S1	Wejście	S1
PORTD4	Pind.4	S2	Wejście	S2
PORTD5	Portd.5	INPUT4 mostka H	Wyjście	InputH_4
PORTD6	Portd.6	INPUT3 mostka H	Wyjście	InputH_3
PORTD7	Portd.7	INPUT1 mostka H	Wyjście	InputH_1

pętlą główną, w której mikrokontroler kolejno zaświeca i gasi wszystkie diody LED (LED1... LED4), a następnie uruchamia na chwilę brzęczyk piezozo.

W pliku List02.bas przedstawiono ogólny szkielet kolejnych programów dla robota. Jest to baza do tworzenia kolejnych programów zawierająca elementy takie, jak konfiguracja portów, licznika/timera1 oraz deklaracja

i treść procedury ułatwiającej sterowanie silnikami (poprzez podanie wartości do jedynie dwóch zmiennych).

W kolejnym programie testowym (plik List03.bas) procesor steruje silnikami robota w następujący sposób: robot jedzie do przodu przez 1 sekundę, potem obraca się dookoła przez 2 sekundy i porusza się do tyłu przez 1 sekundę.



**Rys. 11. Budowa koła Kasfora**



Na CD karty katalogowe i noty aplikacyjne elementów oznaczonych na Wykazie Elementów kolorem czerwonym

## WYKAZ ELEMENTÓW

### Płytki z sensorami

#### Rezystory

R1, R3: 390 Ω  
R2, R4: 10 kΩ  
R5, R6: 47 kΩ  
R7, R8, R10: 100 Ω  
R11, R12, R13: 68 Ω  
R9: 1 kΩ

#### Kondensatory

C1, C2, C8: 4,7 μF/16 V  
C3, C4: 100 μF/16 V  
C5, C6, C7: 100 nF

#### Półprzewodniki

Q1: BD135  
T1-LED1, T3-LED2: QRD1114  
U1, U2, U3: TSOP1736

T2, T4: SFH300  
LED3, LED4, LED5: LD271

#### Inne

S1: μswitch kątowy  
JP1: goldpin 1×9 plus gniazdo  
X1, X2: goldpin 1×3 plus 2× jumper  
Koszyk na baterie

### Płyta główna

#### Rezystory

R1, R2, R4, R5: 220 Ω  
R3: 1 kΩ

#### Kondensatory

C1, C2, C6, C12, C15: 100 μF/16 V  
C3, C4, C5, C7, C8, C11, C13, C14, C16:  
100 nF  
C9, C10: 33 pF

#### Półprzewodniki

U1: L4940V05  
U2: ATmega8  
U3: L298  
D1...D8: BYV26E  
LED1, LED3: Czerwona 5 mm  
LED2, LED4: Zielona 5 mm

#### Inne

S1, S2, S3: μswitch  
JP1: goldpin 1×9 plus gniazdo  
JP2-JP12: goldpin 1×2 plus jumpery  
JP13 ISP: goldpin 2×5  
Q1: kwarc 8 MHz  
Przewód: 9 żył ok. 150 mm  
Podstawka DIL28  
Brzęczyk z generatorem  
ARK2: 3 szt.

Po pierwszych, wstępnych ćwiczeniach i testach można przesłać do procesora najprostszy program, wykorzystujący czujniki. Jak wspomniano we wcześniejszej części artykułu, warto zająć się budową „Light followera”. W tym celu trzeba najpierw skonfigurować przetwornik analogowo-cyfrowy, a potem zadeklarować zmienne do obsługi danych z ADC. Program umieszczono w pliku `List04.bas`. Rozpoczyna się od krótkiego sygnału dźwiękowego, następnie wykonywana jest pętla, w której sprawdzane jest czy i z której strony dobiega silniejsze światło. W razie stwierdzenia silniejszego oświetlenia z jednej ze stron, robot blokuje leżący z tej strony silnik a drugi włącza z pełną prędkością. W przypadku wystąpienia identycznych sygnałów zarówno z prawej, jak i z lewej strony, mikrokontroler załącza oba silniki z połową prędkości do przodu.

Zgodnie z przewidywaniami autora artykułu, program „Light follower” dobrze ilustrujący zasadę jego działania, nie działa bezbłędnie. Niski stopień skomplikowania programu został okupiony błędem, wynikającym ze zbyt dużej dokładności pomiarów przetwornika. Oprócz tego, że wyniki, docierające z obydwu kanałów zawsze będą nieco inne (nawet w hipotetycznym przypadku identycznego poziomu oświetlenia), co wynika m.in. z różnic parametrów zastosowanych komponentów, to również poziom oświetlenia nigdy nie będzie identyczny z obu stron. Niewielkie różnice w natężeniu światła, które występowały podczas poruszania się robota, powodowały drgania i jego „niepewne” zachowanie się. Dopiero dodanie kodu pozwalającego na wykonanie pomiaru różnicowego (ściślej – porównanie zmierzonych

wartości z użyciem predefiniowanego progu liczbowego, wyrażonego w kwantach) umożliwiło prawie całkowite wyeliminowanie tej wady. Spowodowało także wytworzenie się charakterystycznej „czułości”. Przy dużej wartości progu robot nie reaguje na mniejsze różnice, więc jego działanie koncentruje się wokół naprawdę jasnych źródeł oświetlenia, a ignoruje np. światło dzienne. Dzięki takiej metodzie pomiaru robot interpretuje różnicę w oświetleniu mniejszą niż zadany próg (w przykładzie ustalono wartość 20) jako jednakowe oświetlenie z obu stron. Program wykorzystujący omówiony wyżej algorytm działania, zapisano w pliku `List05.bas`.

Przemysław Musz  
[www.przemotronik.pl](http://www.przemotronik.pl)

R E K L A M A M A

# Uniwersalna karta portów na USB

## AVT414



**Dostępne wersje:**  
A - płytki drukowane: 34zł  
B - komplet elementów: 67zł  
C - układ zmontowany: 130zł

**www.sklep.avt.pl**



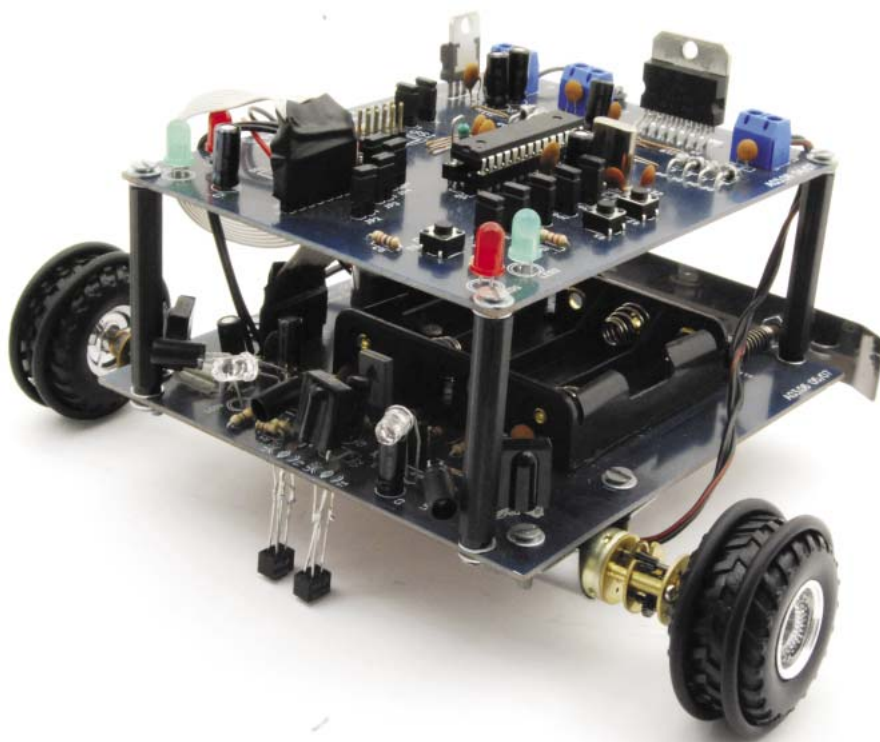
# GreenBot (2)

## Robot edukacyjny – programowanie



W EP 3/2009 rozpoczęliśmy publikację projektu niezwyklego robota edukacyjnego o nazwie GreenBot. W pierwszej części opisaliśmy budowę robota oraz proste programy służące do jego przetestowania. W tym odcinku opisujemy bardziej zaawansowane algorytmy: pojazdu zmierzającego w kierunku najsilniejszego źródła światła, jadącego wzdłuż linii namalowanej na podłożu, pojazdu zdalnie sterowanego, omijającego przeszkody.

**Rekomendacje:** robot do samodzielnej budowy, dla pasjonatów robotyki na różnych poziomach zaawansowania. Szczególnie polecany jako projekt edukacyjny.



### Światłolub

Kolejny program, zawarty w pliku *list06.bas* to zmodyfikowany algorytm „Waldka Światłoluba”, w którym zasada przeprowadzania obliczeń jest taka sama, jak poprzednio. Jeżeli jednak bezwzględna różnica pomiaru jest niższa od zadane-

go progu, robot zaczyna cofać się z prędkością równą połowie szybkości całkowitej (PWM=127). Gdy w czasie jazdy do tyłu robot napotka na przeszkodę, tylny czujnik stykowy podaje stan niski na odpowiednie wejście mikrokontrolera. Fakt ten zostanie wykryty za pomocą instrukcji *If..then*, umieszczonej w głównej pętli programu, zaraz po sprawdzaniu stanu fototranzystorów. Robot zmieni wtedy kierunek jazdy i będzie poruszał się przed siebie ze zwiększoną prędkością przez ok. 0,5 sekundy. Dzięki temu opóźnieniu możliwe będzie oddalenie się na większą odległość od przeszkody, co sprawi, że robot uniezależni się od drgań zestyku (o ile tylko czujnik będzie odpowiednio zainstalowany i drgania w czasie jazdy i nie będą powodowały fałszywych alarmów). GreenBot reaguje natychmiast na najkrótsze nawet zwarcie styków. Wprowadzenie opóźnienia to jeden ze sposobów na programowe odkłócanie przebiegów z elementów stykowych, o którym była mowa we wcześniejszej części artykułu. W pętli głównej umieszczono polecenie opóźnienia o 100 ms, co eliminuje ewentualne drgania styków czujnika.

Alternatywnym sposobem na ciągłe sprawdzanie stanu ważnego czujnika jest wykorzystanie jednego z przerwań. Prosta metoda *polling*,

» część pierwsza tego artykułu dostępna jest na CD

## AVT-5176

### PODSTAWOWE PARAMETRY

- Płytki jednostronne o wymiarach: 93,6×121,3 mm (główna) i 112,7×120,3 mm (sensoryczna)
- Zasilanie z baterii lub akumulatorów (6 VDC)
- Mikrokontroler ATmega8L (lub podobny)
- Konstrukcja otwarta do rozbudowy
- Napęd: 2 silniki DC lub przerobione serwomechanizmy
- Czujniki: 2× fototranzystory, 2× czujniki linii, 3× radary podczerwieni, 1× czujnik stykowy
- Interfejs użytkownika: 2× programowalne microswitch-e, przycisk reset, diody LED (2× zielona, 2× czerwona), 1× buzzer piezo
- Dostępne zestawy funkcji: light follower, line follower, robot zdalnie sterowany (RC5), omijanie przeszkód etc.
- Wbudowana płytka uniwersalna
- Możliwość wysterowania silników do 2 A
- Programowanie przez złącze ISP Kanda

W ofercie AVT:  
AVT-5176A – płytka drukowana

» PROJEKTY POKREWNE wymienione artykuły są w całości dostępne na CD

Tytuł artykułu	Nr EP/EdW	Kit
Pojazd zabawka	EP 1/2009	AVT-5165
Zabawka – programowany pojazd	EP 2-3/2002	AVT-5051
Robot (a właściwie jego ręka)	EP 7-9/1999	AVT-821
Zabawka – pojazd z radarem	EP 7/1998	AVT-429
ROBORobak	EP 7-8/2005	----

zastosowana w poprzednim programie, wiąże się z zajmowaniem niewielkiej części mocy obliczeniowej procesora, a więc przeznaczeniem pewnego czasu na realizację operacji, w czasie którego mikrokontroler mógłby zająć się realizacją ważniejszych zadań. Znacznie lepszą wydaje się metoda kontrolowania czujnika tylko wtedy, gdy ten zadziała (asynchronicznie). W tym celu podłączono mikroprzełącznik zderzaka do linii PORTD.2, to jest INTO. Program, wykorzystujący funkcję INTO, zapisano w pliku *list07.bas*. Warto wypróbować działanie obydwu wersji programu, najlepiej odpowiednio je modyfikując, na przykład zmieniając wartość opóźnienia na końcu pętli głównej, lub zupełnie usuwając to polecenie. W prostszych realizacjach można z powodzeniem zastosować poprzedni kod, natomiast w aplikacjach wymagających dużej szybkości wykonywania pętli głównej, warto wykorzystać program wykorzystujący przerwania.

## Line Follower

Kolejnym przykładem wykorzystania robota jest budowa Line Follower-a. O ogromnej i stale rosnącej popularności takich konstrukcji może świadczyć fakt, że oraz częściej organizowane są konkursy dla tego typu robotów. Ogólna zasada działania jest bardzo prosta: robot ma uważać, by nie zjechać z trasy wyznaczonej przez linię na podłożu. Przy zastosowaniu dwóch czujników: prawego i lewego, realizacja programowa takiego zadania wydaje się nie sprawiać większych problemów. Jedną z możliwości, opierającą się o pomiar różnicowy, jest prawie identyczna, jak w przypadku omówionego wcześniej Waldka Światłoluba. Podobieństwo zawiera się w sposobie obsługi czujników. Program należy jednak przerobić w taki sposób, by przy sygnale (białym kolorze) z danej strony, robot skręcił w stronę przeciwną. Oznacza to, że jeśli robot wypadnie z trasy i prawy czujnik wykryje jaśniejszy kolor po swojej stronie, to ma skręcić w lewo, by znów linia znalazła się pomiędzy czujnikami. Jeżeli obydwa czujniki w takiej konfiguracji nie wykrywają linii (zwracają niskie napięcie), to robot może np. cofać się aż do momentu powrotu na trasę. Algorytm tego typu zapisano w pliku *list08.bas*. Dodano w nim kolejną instrukcję warunkową: bez niej robot cofałby się, ilekroć obydwa czujniki znalazłyby się na linii. Próg (w przykładzie równy 600) należy dobrać we własnym zakresie metodą prób i błędów lub za pomocą pomiarów napięcia na kolektorze fototranzystora. Należy pamiętać, by przed uruchomieniem programu przestawić zworki na płytce dolnej w celu podłączenia czujników linii do wejść ADC. Warto wypróbować różne ustawienia czujników, ponieważ nieznaczne przybliżenie lub oddalenie od siebie sensorów może dać wymierne korzyści. Zależy to oczywiście od wielu czynników, przede wszystkim od odległości czujników od podłoża, szerokości linii oraz najmniejszego promienia skrętu trasy (przy wąskiej linii należy zbliżyć do siebie czujniki).

Już po pierwszej serii testów na przygotowanej wcześniej trasie, można dostrzec pewne charakterystyczne cechy robota. Platforma jest zwrotna, a program doskonale spisuje się na większości odcinków drogi. Zastosowanie dwóch czujników, co z resztą bardzo upraszcza konstrukcję i program, może jednak powodować pewne problemy przy pokonywaniu ostrych zakrętów trasy. Mają one również związek z dość dużymi wymiarami całej konstrukcji. Kolejny przykład programu Line Followera zapisano w pliku *list09.bas*. Główna modyfikacja, to wprowadzenie innego sposobu skręcania. Poznana i szeroko stosowana we wcześniejszych programach metoda z blokowaniem jednego koła oraz włączaniem drugiego nie zawsze umożliwia pełne wykorzystanie kinematycznych możliwości pojazdu. Drugi sposób pojawił się po raz pierwszy w programie 3, gdzie robot obracał się dookoła. Realizacja takiego zachowania za pomocą włączenia obydwu napędów w przeciwnych kierunkach (niekoniecznie z jednakowymi prędkościami), pozwala uzyskać dużo szybsze i pewniejsze reakcje. Ponadto, w programie zastosowano sygnalizację stanu czujników za pomocą diod LED. Zaświecenie diody czerwonej po danej stronie robota pokazuje, który czujnik jako ostatni „zgiął” linię. Porównanie działania obydwu programów pozwala wybrać wersję najbardziej odpowiadającą własnym potrzebom, a także zwrócić uwagę na najważniejsze czynniki, które trzeba uwzględnić podczas tworzenia własnego algorytmu.

Choć kwestię wykorzystania biernych czujników światła oraz sensorów linii omówiono przy okazji przedstawionych właśnie programów, możliwości ich wykorzystania są w rzeczywistości jeszcze większe. We własnym zakresie warto spróbować konstrukcji obdarzonych większą „inteligencją”, dodając na przykład procedury powodujące uciekanie Waldka przed silniejszym światłem czy rozbudowując algorytmy poszukiwania zgubionej linii.

## Pojazd zdalnie sterowany

Kolejnym sposobem wykorzystania GreenBota jest budowa zdalnie sterowanego pojazdu. W robocie zastosowano trzy scalone czujniki podczerwieni, które przystosowane są do odbierania przebiegu impulsowego o częstotliwości nośnej równej 36 kHz. W związku z tym, że taka częstotliwość jest również wykorzystywana w systemach zdalnego sterowania urządzeń audio-video, możliwe jest wydawanie odpowiednich poleceń robotowi za pomocą typowego nadajnika podczerwieni (pilot) przeznaczonego do sprzętu RTV. Można przy tym wykorzystać tylko jeden z czujników. Praktyka pokazuje bowiem bardzo dobrą czułość odbiorników zarówno na promieniowanie bezpośrednie, jak i odbite. Tę samą cechę wykorzystano przy wykrywaniu przeszkód, co będzie omówione w dalszej części artykułu.

Prosty program, wykorzystujący gotową funkcję BASCOM-a *GetRC5()*, zapisano w pliku *list10.bas*. Funkcję wywołuje dwukrotne naciś-

nięcie na pilocie dowolnego przycisku z części numerycznej. Na przykład wybranie 0 powoduje wygenerowanie sygnału dźwiękowego oraz włączenie dwóch diod LED. Aby wyłączyć sygnał wystarczy ponownie nacisnąć 0. Pozostała część klawiatury numerycznej służy do sterowania ruchem robota. Klawiszom przyporządkowano następujące funkcje:

- 2 – ruch do przodu;
- 8 – do tyłu;
- 5 - zatrzymanie silników;
- 1,4,7 – różne sposoby skręcania w lewo;
- 3,6,9 – różne sposoby skręcania w prawo.

W programie wykorzystano wszystkie poznane dotychczas sposoby napędu. Warto zwrócić uwagę, że program jest niezależny od adresu urządzenia. Nie powinno więc być problemu z pilotami, zarówno tymi, przeznaczonymi wyłącznie do telewizorów, jak i z uniwersalnymi. W przypadku nieznanymi dokładnych numerów rozkazów, ale także w celu zwiększenia możliwości sterowania robotem, warto zastosować rozbudowany program z *list11.bas*. Przedstawia on program z możliwością nauki kodów klawiszy, dzięki czemu możliwe jest jak najwygodniejsze przyporządkowanie funkcji klawiszom pilota. Każde naciśnięcie przycisku S1 powoduje zwiększanie stanu programowego licznika odpowiedzialnego za numer aktualnie ustawianej opcji. Po tym następuje wygenerowanie serii krótkich pisków brzęczyka, których liczba odpowiada numerowi aktualnie programowanej opcji. Następnie robot oczekuje na odebranie kodu z pilota. Gdy ten zostanie odebrany, to po prawidłowym odczytaniu kodu naciśniętego przycisku (wybór jest całkowicie dowolny, można używać również pilotów z przyciskami wyboru urządzenia), następuje krótki błysk diody LED1, co sygnalizuje zapisanie kodu w pamięci EEPROM mikrokontrolera. Po tej procedurze można przejść do zapisania kolejnego przycisku. Gdy stan licznika przekroczy wartość maksymalną, to zaświecą się obydwie zielone diody. Wówczas można zresetować robota i rozpocząć testy. Przycisk S2 umożliwia cofnięcie licznika o 1 w przypadku, gdy podczas „nauki” omyłkowo naciśnięto niepożądany przycisk i chce się zmienić poprzednio ustawiony kod.

Warto zwrócić uwagę na algorytmy, zastosowane w tym programie. Ze względu na naj-

wyższy stopień skomplikowania programu ze wszystkich przedstawionych do tej pory, omówione zostaną zastosowane w nim kluczowe procedury. Przyciski obsługiwane są pojedynczo. Zabezpieczenie przed wielokrotnym odczytaniem raz wciśniętego przycisku zrealizowano z pomocą zmiennych bitowych *Ena* i *Ena2*. Naciśnięcie S1 lub S2 powoduje odpowiednio inkrementowanie lub dekrementowanie programowego licznika (zmienna *Licznik*). Następnie sprawdzana jest zawartość zmiennej i jeśli jest o 1 większa od liczby elementów tablicy (programowanych opcji, w przykładzie 12), zaświecane są zielone diody LED2 i LED4, a programowemu licznikowi nadawana jest wartość 1. Potem w programie następuje pętla *For...Next*, której zadaniem jest generowanie sygnałów dźwiękowych z brzęczyka. Linie programu:

```
Bitwait Srodkowyir , Reset
Getrc5(adres , Rozkaz)
```

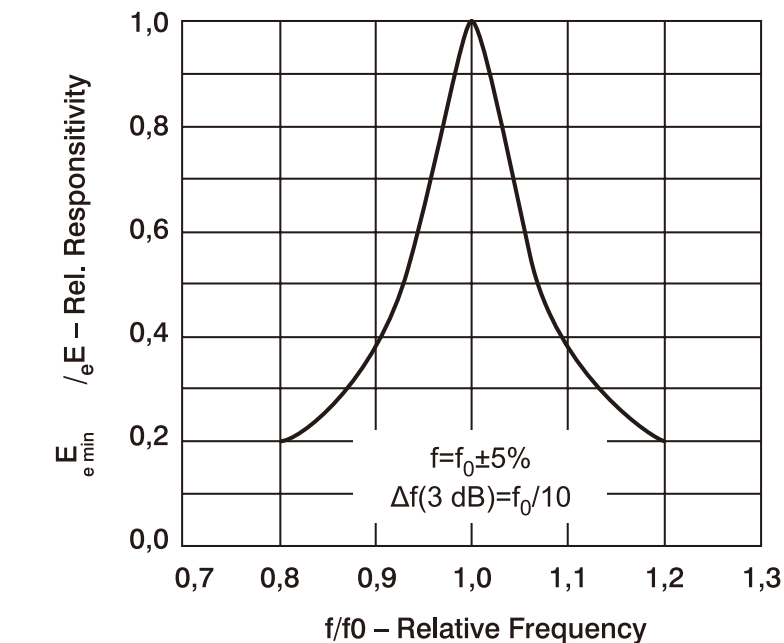
odpowiadają za oczekiwanie, a następnie odbiór kodu z pilota. Po odebraniu i zdekodowaniu sygnału (otrzymane wartości znajdują się w zmiennych *Rozkaz* i *Adres*), następuje zapisywanie sekwencji kodowej do postaci typu *Word*. Warto przyrzeć się temu dokładniej. Za opisane wyżej czynności odpowiadają poniższe linie programu:

```
Temp = Adres
Shift Temp , Left , 8
Temp = Temp + Rozkaz
Kod(licznik) = Temp
```

Zmiennej *Temp* nadawana jest wartość adresu odebranej komendy. Jest to zawsze wartość o wiele mniejsza, niż 255 (standardowo RC5 obsługuje 32 adresy), więc z powodzeniem mieści się w pierwszych ośmiu najmłodszych bitach zmiennej. Wszystkie bity są następnie przesuwane w lewo, a do wolnych, mniej znaczących bitów, przepisywana jest zawartość zmiennej *Rozkaz*. Celem takiego zabiegu jest uproszczenie przeszukiwania pamięci EEPROM.

Po zapisaniu 16-bitowego kodu, zaświeca się na chwilę dioda LED1. Wszystkie kody zapisywane są w komórkach zmiennej tablicowej typu *Word* (zmienna *Kod(12)*). Ilość komórek uzależniona jest od woli Czytelnika. Należy zadeklarować tablicę o tylu elementach, ile opcji ma być wywoływanych z poziomu pilota.

Jeżeli teraz użytkownik wprowadzi w taki sam sposób drugi (następny) kod, procedura powtórzy się dla następnej komórki chyba, że bezpośrednio po wprowadzeniu komendy zostanie naciśnięty S2, co spowoduje powrót do poprzedniej wartości licznika. Jeżeli przycisk S1 nie zostanie naciśnięty, to GreenBot przechodzi do trybu wykonywania rozkazów. Warto przed tym zresetować mikrokontroler za pomocą przycisku S3, by omyłkowo nie przeprogramować którejś z ustawionych poprzednio opcji. Robot odbiera i dekoduje komendy z pilota za pomocą funkcji *GetRC5()* i konwertuje je na postać 2-bajtową. Następnie przeszukiwana jest pamięć EEPROM, w której zapisane są komendy. Odpowiada za to następujący fragment programu:



Rys. 1. Charakterystyka zastosowanych czujników podczerwieni  $f_0=36$  kHz

```
For I2 = 1 To 12
  If Temp2 = Kod(i2) Then
    Wybor = I2
  Exit For
End If
Next I2
```

Zmienna *I2*, automatycznie zwiększana o 1 w każdej iteracji pętli wyznacza, która komórka tablicy ma być porównana z kodem zapamiętanym w zmiennej *Temp2*. Gdy wartości będą równe, to odpowiedni indeks zostanie przekazany do zmiennej *Wybor*. Ta z kolei zostanie sprawdzona za pomocą procedury *Select*, znanej z poprzedniego programu. Przy ilości opcji innej niż 12, należy oczywiście odpowiednio zmienić ilość możliwości *Case* oraz odpowiednie wartości w pozostałej części programu.

Na sterowaniu za pomocą pilota RC5 nie kończą się, oczywiście, możliwości GreenBota.

### Samodzielne omijanie przeszkód

Kolejnym sposobem wykorzystania GreenBta jest budowa inteligentnego pojazdu, który potrafi samodzielnie omijać przeszkody i wybierać właściwą trasę. W związku z tym, że robot nie posiada żadnych możliwości bezwzględnego pozycjonowania, a także nie zawiera enkoderów lub innych czujników położenia stosowanych w metodach odometrycznych, sterowanie nim musi odbywać się będzie w zupełnie inny sposób. Jako że jest to konstrukcja mobilna poruszająca się w jednej płaszczyźnie, musimy zastosować inne algorytmy sterowania. Prosta kontrola za pomocą opóźnień nie zdałaby tutaj egzaminu, więc bez orientacji w płaszczyźnie kartezjańskiej (wyznaczonej przez podłoże) wykorzystamy sterowanie w czasie rzeczywistym jedynie za pomocą odpowiednio analizowanych sygnałów, pochodzących z radarów podczerwieni oraz tylnego zderzaka. W bardziej zaawansowanych i złożonych algorytmach można pokusić się o wykorzystanie pozostałych czujników.

Prosty program, wykorzystujący czujniki i nadajniki IR w roli radarów, zapisano w pliku *list12.bas*. Do obsługi diod nadawczych IR wykorzystano Licznik/Timer2, który w przeciwieństwie do Timera0, zawiera obwody generowania PWM, z wyjściem na pinie PORTB.3. W programie zintegrowano procedurę wykrywania przeszkód z właściwym programem interpretującym ich położenie oraz ustalającym odpowiednią reakcję na sygnały z nich pochodzące. Najważniejszym blokiem programu jest ten odpowiedzialny za generowanie modulowanego promieniowania podczerwonego. Timer/Counter2 pracuje w trybie porównania. W linii *Ocr2=30* następuje zapisanie wartości do rejestru, z którym porównywana będzie zawartość licznika T/C2. Istotny jest fakt, że odpowiednie dobranie wartości tego rejestru pozwala dostosować nadajniki do pracy z odpowiednią czułością oraz częstotliwością odbiornika. Dla zastosowanych odbiorników najwyższą czułość można uzyskać z wartościami zbliżonymi do 14. Próg zadziałania scalonych odbiorników IR podwyższa się w bardzo szerokim zakresie, wraz z oddalaniem się częstotliwości pracy od częstotliwości środkowej odbiornika, a zmiany te są bardzo nieliniowe. Główną częścią programu, odpowiedzialną za nadawanie, jest następujący blok:

```
Incr Count
If Count = 250 Then Count = 0
If Count = 109 Then
  Start Timer2
  Akcja = 1
Else
  Stop Timer2
  Akcja = 0
End If
Waitms 1
```

W dalszej części programu następuje procedura wyliczania odpowiedniego kierunku jazdy w danych warunkach. Warto wspomnieć, że z opisanym programem skuteczność wykrywania przeszkód przez GreenBota jest bar-

dzo duża, mimo że nie jest wykorzystywany środkowy czujnik podczerwieni. Jego stan jest jedynie sygnalizowany za pomocą diody LED2. W programie można zauważyć puste miejsce za ostatnią instrukcją *else*. Warto wypróbować we własnym zakresie możliwości wykorzystania tej opcji. Można spróbować zastosować tam instrukcje, które nie mogą być użyte w przypadku, gdy aktywne są radary. Może to być na przykład wykorzystanie czujników światła, odbiór kodu RC5 z pilota lub zastosowanie dodatkowego czujnika, który mógłby być zakłócany przez nadajniki IR robota.

Maksymalny zasięg, uzyskiwany przez czujniki podczerwieni jest w stosunku do wymiarów robota bardzo duży i wynosi kilkadziesiąt centymetrów. Testy wykazały, że ręka, ustawiona prostopadle do czujnika, była skutecznie wykrywana już przy odległości 80...90 cm. Taki zasięg jest jednak możliwy do uzyskania tylko po zasłonięciu dwóch pozostałych czujników. Gdy odsłonięto wszystkie diody nadawcze, to czujniki wzajemnie zakłócały się uniemożliwiając poprawną pracę. Okazało się również, że jeśli częstotliwość pracy diod IR jest bardzo zbliżona do częstotliwości środkowej czujnika, to konieczne jest szczególnie dokładne osłonięcie diod. O ile bowiem rozwiązanie, pokazane w pierwszej części artykułu, wystarczy dla małego zasięgu czujników, o tyle w dla wartości OCR2 bliskich 14, lepiej spisywały się osłonki

dłuższe, obejmujące diody łącznie z kołnierzem obudowy. Warto wypróbować we własnym zakresie różnorodne rozwiązania.

Na rys. 1 pokazano wykres zależności czułości odbiornika TSOP1736 od częstotliwości padającej fali podczerwieni. Charakterystyka ma strome zbocza, co informuje nas o znacznym spadku czułości odbiornika, gdy częstotliwość odbieranego sygnału odbiega od 36 kHz. Zależność tę można wykorzystać do prostego, zgrubnego pomiaru odległości przeszkód od robota (określenie blisko/daleko/poza zasięgiem, ew. z poziomami przejściowymi). Wystarczy w tym celu zmieniać zawartość rejestru OCR2 i odczytywać przy tym stan czujników.

W artykule przedstawiono szerokie możliwości wykorzystania robota w celach edukacyjnych oraz eksperymentalnych. Rozwiązania konstrukcyjne dopasowano do potrzeb bardzo różnorodnych zastosowań, jednak nic nie stoi na przeszkodzie, by rozbudować warstwę sprzętową GreenBota. Podobna sytuacja ma miejsce z programowaniem. Warto spróbować zaimplementować w robocie bardziej zaawansowane algorytmy, np. prostą sieć neuronową. Ciekawy byłby również program hybrydowy, łączący w sobie jednocześnie obsługę radarów IR i/lub kodu RC5, czujników światła oraz zderzaka.

**Przemysław Musz**  
[www.przemozonik.pl](http://www.przemozonik.pl)

R E K L A M A

