

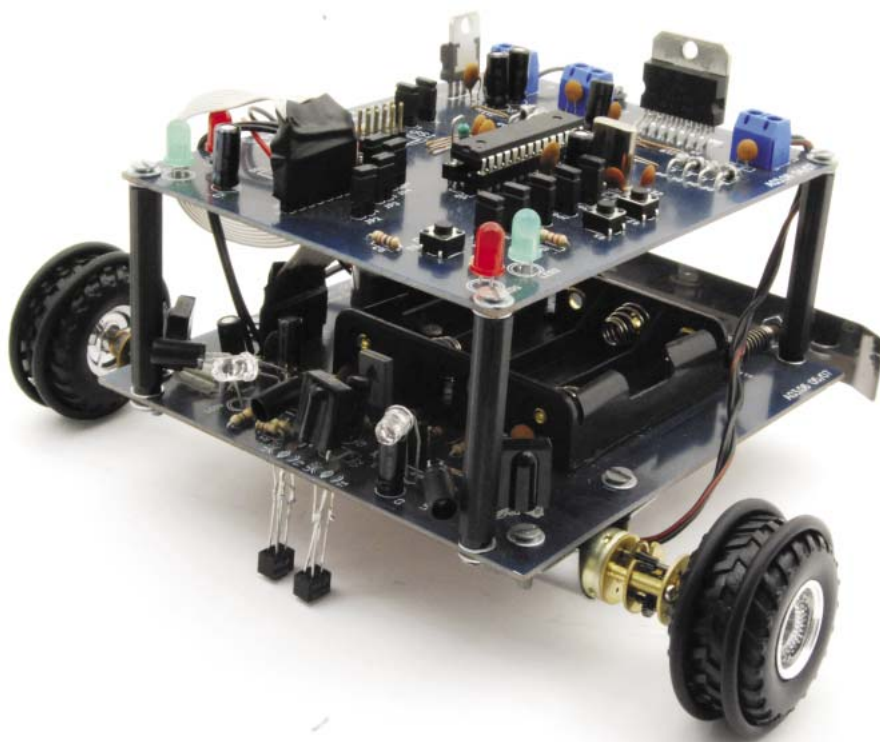
# GreenBot (2)

## 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 przeszkodę.

**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)
```

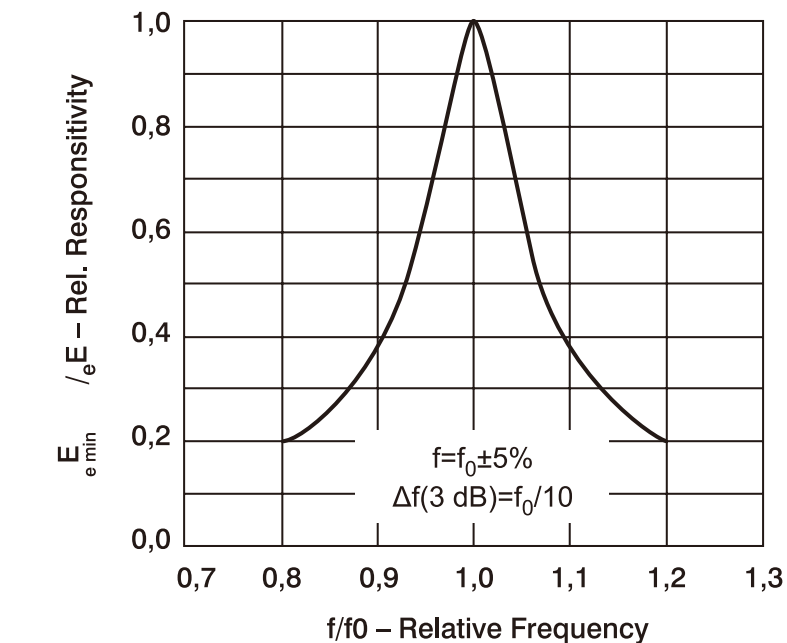
odpowiadają za oczekiwanie, a następnie odbiór kodu z pilota. Po odebraniu i zdekodowaniu sygnału (otrzymane wartości znajdują się w zmiennych *Rozkaz* i *Adres*), następuje zapisywanie sekwencji kodowej do postaci typu *Word*. Warto przyrzeć się temu dokładniej. Za opisane wyżej czynności odpowiadają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



## ZN350M

Uniwersalny moduł prostowniczy

Uniwersalny moduł prostowniczy przeznaczony jest do urządzeń, których moc nie przekracza 300 W. Przystosowany został do pracy ciągłej z napięciem wejściowym 15 – 50 VAC.

Wykorzystanie w module bezpiecznika polimerowego pozwala na chwilowy wzrost wydajności, przez co dedykowany jest do urządzeń pobierających w trakcie startu zwiększony pobór prądu np. sterowniki do silników DC. Bezpiecznik polimerowy nie ulega uszkodzeniu (działa termicznie). Po przeciążeniu odłącza zasilany obwód i załącza go ponownie po zmniejszeniu obciążenia.

### Właściwości:

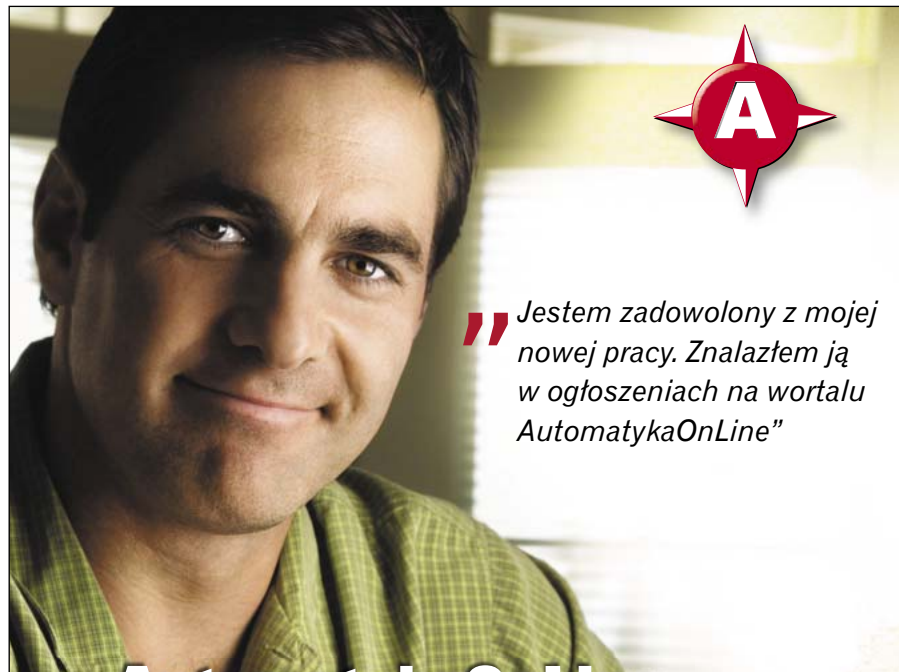
- Mostek prostownika o zwiększonej obciążalności,
- Małe rozmiary,
- Obudowa otwarta na profilu Alu,
- Mocowanie do standardowej szyny 35mm,
- Listwy rozłączne typu Combicon,
- Bezpiecznik polimerowy PTC
- Dioda LED sygnalizująca zasilanie,
- Znamionowy prąd: 6 A
- Znamionowe napięcie: 50 V AC
- Moc maksymalna 300 W
- Bez trafo



www.wobit.com.pl

PPH. WObit Witold Ober  
61-474 Poznań, ul. Gruszkowa 4  
tel. +48 61 8350 -620, -800  
fax +48 61 8350 -704, -804

R E K L A M A



„Jestem zadowolony z mojej nowej pracy. Znalazłem ją w ogłoszeniach na portalu AutomatykaOnLine”

www. **AutomatykaOnLine**.pl  
WORTAL AUTOMATYKI PRZEMYSŁOWEJ

**Wortal AutomatykaOnLine** jest źródłem cennych informacji z zakresu automatyki. Codziennie aktualizowane wiadomości gospodarcze. Nowinki techniczne. Baza wiarygodnych podwykonawców. Informacje o produktach. Ogłoszenia pracodawców i poszukujących pracy. Forum wymiany doświadczeń. Rozwiązania techniczne. Twój partner w biznesie.

**Wortal AutomatykaOnLine**  
ul. Puławska 303, 02-785 Warszawa, tel./fax: 046 857 73 72, e-mail: redakcja@automatykaonline.pl