

Peripheral Pin Select

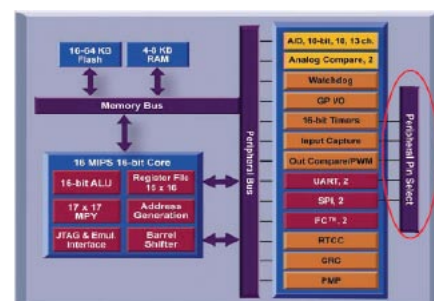
Konfiguracja wyprowadzeń mikrokontrolera w aplikacji

Zazwyczaj przy ocenie mikrokontrolera zwykle się porównywać szybkość (wydajność) rdzenia, wielkość pamięci programu i pamięci danych. Dla doświadczonego konstruktora, oprócz tych dość znaczących cech ważna jest też ilość i jakość zaimplementowanych układów peryferyjnych. Jeżeli takich układów jest wiele, to ich linie muszą być jakoś wyprowadzone na zewnątrz. Stosowane tu są dwa rozwiązania.

Pierwsze z nich i chyba najbardziej popularne, to przypisanie jednemu wyprowadzeniu obudowy wielu funkcji. Na przykład w układzie dsPIC30F6014 linia portu PORTF RF6 jest multiplexowana z zegarem portu SPI1, wejściem przerwania INTO i linią zegarową trzeciego kanału emulatora ICD. Uaktywnienie modułu SPI1 powoduje, że to wyprowadzenie staje się wyjściem zegarowym SCK1. Rozwiązanie takie ma jednak wadę: przypisanie wyprowadzenia do modułu peryferyjnego powoduje, że inny moduł przypisany do tego wyprowadzenia nie może być już w tej aplika-

cji użyty. Rozwiązaniem tego problemu może być takie zwiększenie liczby wyprowadzeń tak, aby wszystkie moduły miały własne linie wejścia/wyjścia. Przykładem jest mikrokontroler PIC24FJ128 umieszczony w obudowie 100-nóżkowej. Układy peryferyjne zajmują ok. 80 linii, a reszta to zasilanie, oscylator i linie portów. Jednak większa obudowa to wyższe koszty układu i płytki drukowanej.

W nowych mikrokontrolerach rodziny PIC24FJ64GA004 zastosowano rozwiązanie, które umożliwia wykorzystanie wielu układów peryferyjnych w mikrokontrolerach



Rys. 1. Budowa mikrokontrolera rodziny PIC24FJ64GA004

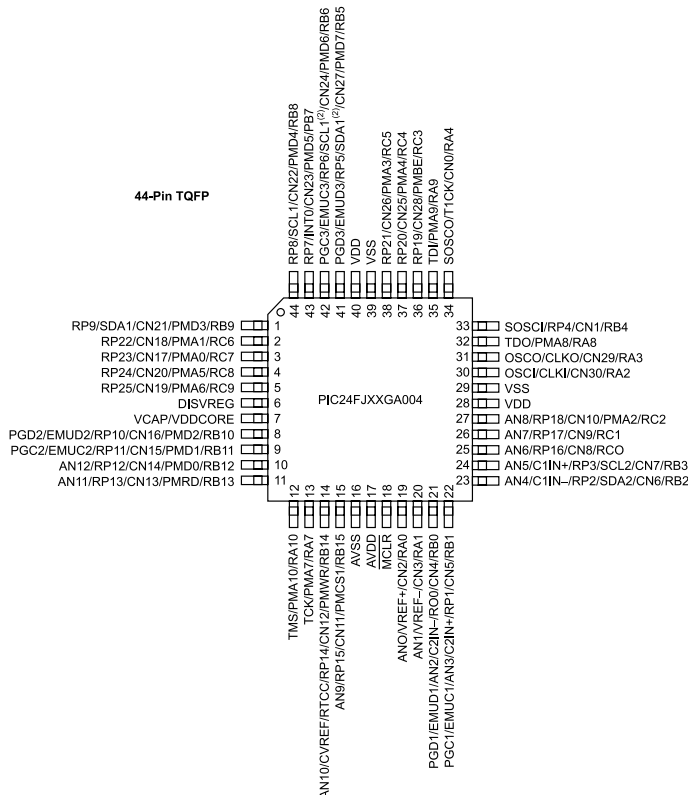
umieszczonych w obudowach z relatywnie małą liczbą nóżek. To rozwiązanie nazywa się Peripheral Pin Select (PPS). Podstawowe właściwości mikrokontrolera PIC24FJ64GA004 tej rodziny pokazano w tab. 1.

Mikrokontroler wyposażony jest w wiele układów peryferyjnych i umieszczone w obudowie 44-nóżkowej. PIC24FJ64GA002 ma taką samą ilość układów peryferyjnych, ale obudowę z tylko 28 wyprowadzeniami. Z konieczności, każde z wyprowadzeń musi być w jakiś sposób przyporządkowane do linii układów peryferyjnych. W klasycznym rozwiązaniu każde z wyprowadzeń miałyby na sztywno przypisane wiele funkcji. Zaimplementowanie mechanizmu PPS powoduje, że użytkownik może elastycznie przypisywać linie układów peryferyjnych do dowolnego (z pewnymi ograniczeniami) wyprowadzenia układu. Istnieje też możliwość dynamicznej zmiany przyporządkowania w trakcie pracy układu. Zmiana przypisania odbywa się programowo i nie wymaga użycia programatora. Mechanizm PPS jest zabezpieczony sprzętowo przed przypadkowym przeprogramowaniem w trakcie normalnej pracy. Na rys. 1 pokazano blokowo budowę mikrokontrolera rodziny PIC24FJ64GA004. Jeżeli chcielibyśmy, aby każde z peryferii z rys. 1 miało swoje własne wyprowadzenie, to obudowa układu musiałaby mieć, tak jak PIC24FJ128, około 100 wyprowadzeń.

Zastosowanie PPS umożliwia sensowne wykorzystanie układów peryferyjnych zależnie od potrzeb aplikacji. Na przykład, kiedy projektujemy coś w rodzaju sterownika komunikacyjnego, to możemy wykorzystać wszystkie porty USART, SPI i I²C, a zrezygnować z układów PWM, Capture itp.

Tab. 1. Podstawowe właściwości mikrokontrolera PIC24FJ64GA004

Wyposażenie	Ilość
UART ze wsparciem LIN i IRDA i 4 poziomowym FIFO	2
SPI z 8 poziomowym FIFO	2
I ² C Master/Slave	2
OUTPUT COMPARE/PWM	5
16BIT TIMERS	5
INPUT CAPTURE	5
10 BIT ADC 500ksamp/sek	13 (bez PPS)
Linie portów I/O	35
Komparatory analogowe	2
PMP Parallel Master Port 8 bitowa szyna adresowa i 8/16 szyna danych	
Sprzętowy RTC, kalendarz i alarm	
Właściwości	
Pamięć programu FLASH	48 kB
Pamięć danych SRAM	8 kB
Max prędkość CPU	16 MIPS
Napięcie zasilania	2.0...3,6 V
Oscylator FRC+PLL, HS+PLL	
Sprzętowe mnożenie 17x17 w jednym cyklu	
Sprzętowe dzielenie liczb 32 bitowych	
Programowanie ICSP	
Interfejs JTAG (Boundary scan + programowanie FLASH)	
Wbudowany stabilizator LDO	
Układ nadzoru zegara: pozwala na wyłączenie rdzenia kiedy zegar taktujący nie jest prawidłowy	
Watchdog z oddzielnym oscylatorem RC	
Układ oszczędzania energii NanoWatt	



Rys. 2. Wyprowadzenia mikrokontrolera w obudowie 44-nóżkowej

Jak się łatwo domyśleć się, układ PPS jest programowanym cyfrowym multiplekserem. Jego działanie podlega kilku ograniczeniom. Wejścia układów analogowych: przetwornika i komparatorów nie mogą być przypisywane mechanizmami PPS. Producent tłumaczy to zwiększonym kosztem multipleksera mającego możliwość jednoczesnego przełączania

sygnałów analogowych i cyfrowych. Nie jest też możliwa zmiana wyprowadzeń cyfrowego portu PMP (*Parallel Master Port*). Tu z kolei trzeba by było przełączać wiele sygnałów (16 lub 24) w niewielkiej obudowie. Zysk wynikający z zastosowania PPS przy przełączaniu wielu linii tego portu byłby znikomy. Kolejnym ograniczeniem jest wyłączenie linii interfejsu I²C (dwukierunkowa linia danych), oraz interfejsów USB i LAN (tutaj nie stosowanych), wejść przerwań od zmiany stanu linii portów (rezystory podciągające).

Wszystkie linie interfejsów niepodlegających mechanizmowi PPS (ADC, PMP, I²C) domyślnie są przypisane do linii I/O mikrokontrolera, tak jak to było dotychczas. Uaktywnienie takiego interfejsu powoduje przejęcie domyślnie na stałe mu przypisanego mu wyprowadzenia I/O. Inaczej się ma

sprawa z liniami interfejsów, których linie mogą być przypisywane do wyprowadzeń układu przez PPS. Przed ich użyciem trzeba je jawnie przypisać do wybranego wolnego wyprowadzenia mikrokontrolera.

Na rys. 2 pokazano wyprowadzenia mikrokontrolera w obudowie 44-nóżkowej. Jako domyślne zostały tu zaznaczone: linie portów I/O, wejścia przetwornika ADC i komparatorów, 2 interfejsów I²C, linie interfejsu PMP i wejść przerwań od zmiany stanów na liniach portów (CN0...CN30). Pozostałe interfejsy, jeżeli mają być użyte, to muszą być przypisane do jednego z wyprowadzeń oznaczonych RP0...RP25.

Wszystkie linie mają priorytety. Najwyższy mają linie domyślnie przypisane do wyprowadzeń (ADC, I²C itp.). Jeżeli taki interfejs zostanie uaktywniony, to przejmuje linię i nie będzie możliwe przyłączenie jej przez PPS. Z tego powodu, kiedy wykorzystywany jest na przykład wewnętrzny moduł ADC, to od niego należy zacząć przyporządkowywanie wyprowadzeń mikrokontrolera. Średni priorytet mają linie interfejsów przypisywane przez PPS. Linie nieużywane przez żaden z interfejsów, mogą być liniami I/O (mają najniższy priorytet).

Przypisaniem linii układów peryferyjnych do wyprowadzeń mikrokontrolera przez Peripheral Pin Select sterują 2 rejestry obszaru SFR. Pierwszy z tych rejestrów nazywany RPINRx przypisuje wejścia peryferii (na przykład RxD interfejsu UART) do wyprowadzenia. Drugi rejestr RPORx przypisuje wyjścia peryferii do wyprowadzenia (na przykład TxD interfejsu UART)

Po wyzerowaniu mikrokontrolera każde wejść układów peryferyjnych (podlegających mechanizmowi PPS) jest podłączone do RP0, bo wszystkie bity rejestrów RPINRx są wyzerowane. Przypisanie wyprowadzenia do wejścia układu peryferyjnego polega na wpisaniu rejestru RPINx 6-bitowej liczby *n* określającej numer wyprowadzenia opisanego jako RPN. W tab. 2 pokazano przypisanie bitów rejestrów RPINRx do linii wejściowych peryferii. Na przykład, kiedy chcemy do RP8 przypisać wejście przerwania INT4, to do rejestru RPIN2 wpisujemy 00001000B. Taki mechanizm pozwala na przypisanie kilku wejść do jednego wyprowadzenia. Daje to dodatkowe możliwości konfiguracji układowej już po zaprojektowaniu płytki drukowanej

Każde wyzerowanie mikrokontrolera powoduje, że do rejestrów RPORx są wpisywane zera i żadne z wyjść układów peryferyjnych nie jest połączone z wyprowadzeniami RPx. Taka linia staje się uniwersalna linia I/O. W tab. 3 pokazano przyporządkowanie bitów rejestrów RPORx wyjściom układów peryferyjnych, a tabeli 4 przyporządkowanie rejestrów RPORx wyprowadzeniom RPN

Programowa obsługa PPS zaczyna się od inicjalizacji po zerowaniu mikrokontrolera.

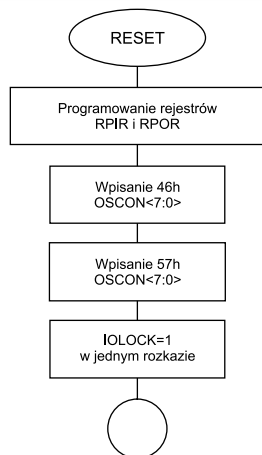
Tab. 2. Przypisanie wejść układów peryferyjnych do rejestrów RPINRx			
Nazwa wejścia	Funkcja	Bit rejestru	Bit konfiguracji
External Interrupt 1	INT1	RPINR0<13:8>	INT1R<5:0>
External Interrupt 2	INT2	RPINR1<5:0>	INT2R<5:0>
External Interrupt 3	INT3	RPINR1<13:8>	INT3R<5:0>
External Interrupt 4	INT4	RPINR2<5:0>	INT4R<5:0>
Timer2 External Clock	T2CK	RPINR3<5:0>	T2CKR<5:0>
Timer3 External Clock	T3CK	RPINR3<13:8>	T3CKR<5:0>
Timer4 External Clock	T4CK	RPINR4<5:0>	T4CKR<5:0>
Timer5 External Clock	T5CK	RPINR4<13:8>	T5CKR<5:0>
Input Capture 1	IC1	RPINR7<5:0>	IC1R<5:0>
Input Capture 2	IC2	RPINR7<13:8>	IC2R<5:0>
Input Capture 3	IC3	RPINR8<5:0>	IC3R<5:0>
Input Capture 4	IC4	RPINR8<13:8>	IC4R<5:0>
Input Capture 5	IC5	RPINR9<5:0>	IC5R<5:0>
Output Compare Fault A	OCFA	RPINR11<5:0>	OCAFR<5:0>
Output Compare Fault B	OCFB	RPINR11<13:8>	OCFBR<5:0>
UART1 Receive	U1RX	RPINR18<5:0>	U1RXR<5:0>
UART1 Clear To Send	U1CTS	RPINR18<13:8>	U1CTSR<5:0>
UART2 Receive	U2RX	RPINR19<5:0>	U2RXR<5:0>
UART2 Clear To Send	U2CTS	RPINR19<13:8>	U2CTSR<5:0>
SPI1 Data Input	SDI1	RPINR20<5:0>	SDI1R<5:0>
SPI1 Clock Input	SCK1	RPINR20<13:8>	SCK1R<5:0>
SPI1 Slave Select Input	SS1	RPINR21<5:0>	SS1R<5:0>
SPI2 Data Input	SDI2	RPINR22<5:0>	SDI2R<5:0>
SPI2 Clock Input	SCK2	RPINR22<13:8>	SCK2R<5:0>
SPI2 Slave Select Input	SS2	RPINR22<5:0>	SS2R<5:0>

Tab. 3. przyporządkowanie bitów rejestrów RPORx wyjściom układów peryferyjnych

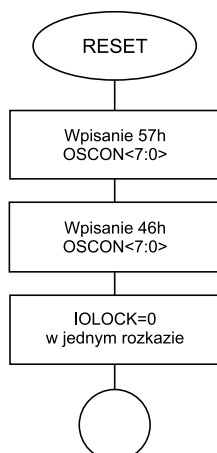
Funkcja	RPnR<5:0>	Nazwa wyjścia
NULL	0	Port I/O
C1OUT	1	RPn ---> Wyjście komparatora 1
C2OUT	2	RPn ---> Wyjście komparatora 2
U1TX	3	RPn ---> UART1 Transmit
U2RTS	4	RPn ---> UART1 Ready To Send
U2TX	5	RPn ---> UART2 Transmit
U2RTS	6	RPn ---> UART2 Ready To Send
SDO1	7	RPn ---> Wyjście danych SPI1
SCK1OUT	8	RPn ---> Wyjście zegara SPI1
SS1OUT	9	RPn ---> Wyjście sygnału Slave Select SPI1
SDO2	10	RPn ---> Wyjście danych SPI2
SCK2OUT	11	RPn ---> Wyjście zegara SPI2
SS2OUT	12	RPn ---> Wyjście sygnału Slave Select SPI2
OC1	18	RPn ---> Output Compare 1
OC2	19	RPn ---> Output Compare 2
OC3	20	RPn ---> Output Compare 3
OC4	21	RPn ---> Output Compare 4
OC5	22	RPn ---> Output Compare 5

Tab. 4. Przyporządkowanie rejestrów RPORx wyprowadzeniom RPn

Pin	Rejestr	Bity powiązane
RP0	RPO0<5:0>	RPOR<5:0>
RP1	RPO0<13:8>	RP1R<5:0>
RP2	RPO1<5:0>	RP2R<5:0>
RPn	RPOn/2<5:0>	RPnR<5:0>
RPn+1	RPOn/2<13:8>	RPn+1R<5:0>



Rys. 3. Blokowanie układu PPS. Uwaga: PPS nie może być ponownie odblokowany kiedy IO1LWAY=1



Rys. 4. Odblokowanie układu PPS

Pierwszym problemem, z którym się musimy zmierzyć jest określenie, w jaki sposób linie będą przypisywane po zerowaniu mikrokontrolera. Pierwsza możliwość, to jednokrotne przypisanie wybranych przez nas układów peryferyjnych do wybranych wyprowadzeń bez możliwości modyfikacji w trakcie działania programu. Druga możliwość, to dynamiczne przeprogramowywanie przypisywania wejść i wyjść w trakcie działania aplikacji.

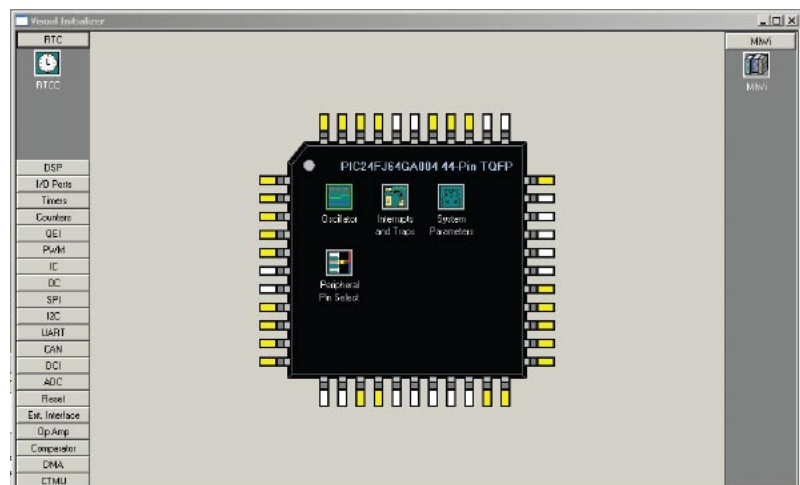
Do sterowania blokowaniem układu PPS wykorzystywane są 2 bity: IOLOCK i IOL1WAY. IOLOCK jest bitem rejestru OSCCON umieszczonego w obszarze rejestrów SFR. Bit IOL1WAY jest umieszczony w rejestrze konfiguracyjnym CW2 (jego modyfikacja jest możliwa tylko w trakcie programowania mikrokontrolera).

Ponieważ jak już powiedzieliśmy wszystkie rejestry RPnR i RPOR są po zerowaniu mikrokontrolera wyzerowane, to przed użyciem peryferii musimy je przeprogramować. Na rys. 3 pokazano najbardziej typowy przykład użycia mechanizmu PPS. Ponieważ rejestry RPnR i RPOR są umieszczone w obszarze SFR, to są narażone na przypadkowe przeprogramowanie. Żeby temu zapobiec wprowadzono blokowanie zapisu do tych rejestrów przez ustawienie bitu IOLOCK. Ustawienie IOLOCK jest możliwe po wpisaniu sekwencji 46h i 57h do 8 młodszych bitów rejestru OSCCON (IOLOCK jest bitem b6 rejestru OSCCON). Inna sekwencja zapisu do OSCON nie spowoduje jego ustawienia. Jeżeli dodatkowo bit IO1LWAY jest ustawiony, to nie będzie możliwe ponowne wyzerowanie bitu IOLOCK (i odblokowanie PPS) aż do ponownego zerowania mikrokontrolera. Sekwencja pokazana na rys. 3 całkowicie zabezpiecza układ PPS przed przeprogramowaniem jeżeli IO1LWAY jest ustawiony. Na rys. 4 pokazano sekwencję odblokowania PPS po zerowaniu mikrokontrolera.

Kiedy IOLOCK jest wyzerowany, to można modyfikować rejestry RPnR i RPOR. Dobrym zwyczajem jest po modyfikacji tych rejestrów ponowne ustawienie bitu IOLOCK, żeby je zabezpieczyć przed przypadkowymi wpisami.

Układ PPS ma wbudowany niezależny układ kontroli zwartości swoich rejestrów. Wszystkie one mają swoje kopie (niewidzialne przez użytkownika). Jeżeli z powodów innych niż działanie rozkazu zapisu (na przykład z powodu zakłóceń ESD) zwartość rejestru zostanie zmieniona, to układ kontroli spowoduje zerowanie mikrokontrolera.

Dodatkowym zabezpieczeniem jest konieczność zapisu sekwencji zapisu rejestrów OSCCON w jednym rozkazie mikrokontrolera. Jeżeli program jest pisany w języku C, to sekwencje modyfikacji OSCCON należy napisać w assemblerze. Kompilator MPALB-C30 ma specjalnie do tego celu wbudowaną funkcję `__builtin_write_OSCCON (value)`.



Rys. 5. Okno Visual Initializer

Microchip jest znany z tego, że potrafi umiejętnie wspierać swoje rozwiązania. Wraz z najnowszą wersją środowiska IDE MPLAB dostarczana jest moduł *Visual Initializer*.

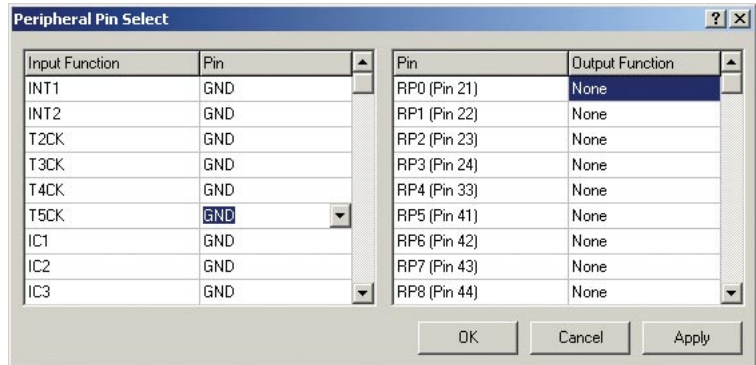
Visual Initializer używany jest do konfiguracji układów peryferyjnych mikrokontrolera. Jeżeli zostanie wybrany mikrokontroler z układem PPS (na przykład PIC24FJ16-GA004), to pojawi się ikona Periperal Pin Select (rys. 5). Po kliknięciu na tę ikonę rozwija się okno PPS pokazane na (rys. 6), w którym można przypisać każdemu z wejść i wyjść RP linie układów peryferyjnych podlegających procesowi PPS. Po graficznym zdefiniowaniu wszystkich peryferii generowany jest kod inicjalizujący, który można umieścić w aplikacji.

Kod tworzony jest w asemblerze MPASM C-30, ale funkcję inicjalizującą można wywołać z poziomu języka C:

```
main()
{
    VisualInitialization();
}
```

Fragment inicjalizacji PPS pokazano na list. 1.

Tomasz Jabłoński, EP
Tomasz.jablonski@ep.com.pl



Rys. 6. Okno Peripheral Pin Select

List. 1. Incjalizacja PPS wygenerowana przez Visual Initializer

```
.text
VisualInitialization:
; Feature=PeripheralPinMap - Associate mappable pin functions with pins.
; Unlock Registers
MOV #OSCCON, W1
MOV #0x46, W2
MOV #0x57, W3
MOV.B W2, [W1]
MOV.B W3, [W1]
BCLR OSCCON, #6
MOV #0x0100, W0
MOV W0, RPINR0
; Lock Registers
; Lock Registers
MOV #OSCCON, W1
MOV #0x46, W2
MOV #0x57, W3
MOV.B W2, [W1]
MOV.B W3, [W1]
BSET OSCCON, #6
```

R E K L A M A



moduł I/O sterowany przez sieć Ethernet



- nadzór nad urządzeniem poprzez stronę internetową
- dostęp do urządzenia zabezpieczony hasłem
- tryb dynamicznego pobierania adresu sieciowego (DHCP)
- możliwość pracy ze stałym adresem IP
- możliwość zmiany adresu MAC
- łatwa konfiguracja przez port USB
- cztery wejścia cyfrowe (odczyt stanów przez www)
- cztery wyjścia wykonawcze o obciążalności 8 A/230 VAC
- możliwość zmiany strony urządzenia (zapis strony internetowej przez USB lub sieć LAN (FTP))
- sygnalizacja stanu pracy: diody LED
- zasilanie: 12 VDC/200 mA (dołączony zasilacz)

Kod handlowy: AVTMOD05
Cena: 350zł

AVT-Korporacja Sp. z o.o.,
03-197 Warszawa, ul. Leszczynowa 11
tel. 022 257 84 50, fax 022 257 84 55,
e-mail: handlowy@avt.pl



www.sklep.avt.pl