

Interfejsy CAN i LIN – zastosowania, część 2

Obsługa błędów

W protokole CAN pod pojęciem błędu rozumie się jedno z poniższych zdarzeń:

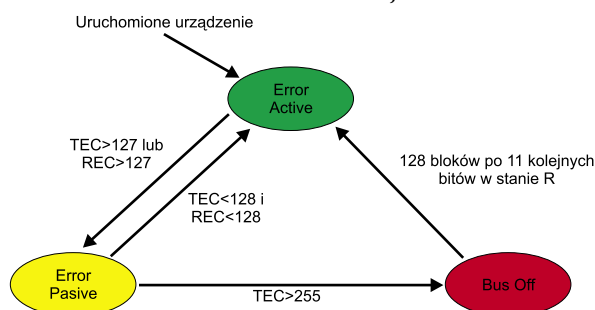
- niezgodność danych – nadawca wykrył różnicę pomiędzy danymi nadawanymi a obecnymi na magistrali (po fazie arbitrażu);
- błąd potwierdzenia – ramka nie została potwierdzona;
- błąd synchronizacji – więcej niż pięć jednakowych bitów zostało odebranych;
- błąd CRC – odebrane CRC nie zgadza się z wyliczonym;
- błąd formatu – jedno z ustalonych pól ramki (CRC, ACK, EOF) nie ma oczekiwanego formatu.

Błąd może zostać zgłoszony poprzez wysłanie ramki błędu przez dowolne urządzenie podłączone do magistrali. Po zgłoszeniu błędu ramka powinna zostać retransmitowana o ile wcześniej nie zostanie rozpoczęta transmisja innej ramki. Każde urządzenie podłączone do magistrali wyposażone jest w dwa liczniki:

- licznik błędów odbioru (TEC) – zwiększany przy wykryciu błędu przy nadawaniu ramki, zmniejszany przy poprawnym zakończeniu transmisji ramki;
- licznik błędów nadawania (REC) – zwiększany przy wykryciu błędu przy odbiorze ramki, zmniejszany przy poprawnym odbiorze ramki.

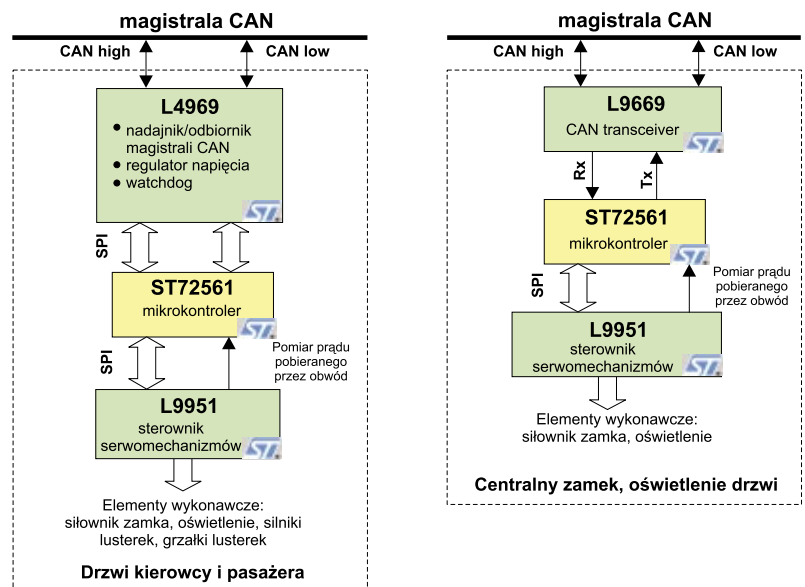
W związku z tymi licznikami urządzenie może znajdować się w jednym z trzech stanów detekcji błędów (rys. 3):

- kiedy oba liczniki zawierają wartości pomiędzy 0 a 127, urządzenie znajduje się w stanie Error Active. W stanie tym może nadawać i odbierać informacje z



Rys. 3.

W drugiej części artykułu omawiamy kolejne zagadnienia teoretyczne związane z działaniem urządzeń wykorzystujących magistrale komunikacyjne CANbus oraz LIN, przedstawiamy także proste przykłady ich aplikacji.



Rys. 4.

magistrali i sygnalizować wystąpienia błędu na magistrali.

- w przypadku osiągnięcia przez jeden lub oba liczniki wartości pomiędzy 128 a 255, urządzenie znajduje się w stanie Error Passive. W tym stanie urządzenie może nadawać i odbierać informacje z magistrali ale nie może sygnalizować wystąpienia błędu na magistrali.
- kiedy jeden z liczników osiągnie wartość powyżej 255 (nastąpi przepełnienie licznika) urządzenie przechodzi do stanu Bus off. W tym stanie urządzenie jest odłączone od magistrali. Może ono przejść do stanu Error Active jeśli poprawnie wykryje 128 bloków po 11 bitów R.

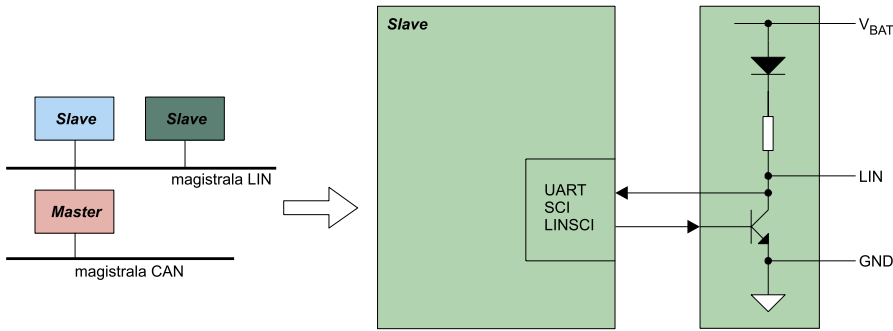
Wskazówki praktyczne

Typowy współczesny samochód średniej i wyższej klasy może zawierać dwie niezależne magistrale CAN przesy-

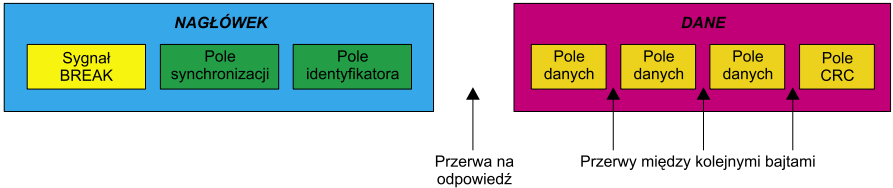
łające dane z różnymi prędkościami:

- CAN low-speed – pracuje z prędkością mniejszą niż 125 kb/s i wykorzystywany jest do obsługi elementów nie wymagających szybkiej komunikacji (takich jak sterowanie siedzeniami, oknami, lusterkami itp.). Urządzenia dołączone do tej magistrali obsługują tryb oszczędności energii (sleep mode), pozwalający oszczędnie wykorzystywać akumulatora przed uruchomieniem zapłonu.
- CAN high speed – pracujący z prędkością do 1 Mb/s jest wykorzystywany do aplikacji wymagających obsługi w czasie rzeczywistym, takie jak zarządzanie silnikiem, systemem hamulcowym, systemem kierowniczym.

Na rys. 4 przedstawiono przykłady wykorzystania magistrali CAN w instalacji drzwi samochodu średniej i wyższej klasy (wyposażonych w centralny zamek, dodatkowe oświetlenie, sterowanie lusterkami zewnętrznymi).



Rys. 5.



Rys. 6.

Przedstawiony w przykładach 8-bitowy mikrokontroler ST72561 firmy STMicroelectronics wyposażono w sprzętowe mechanizmy obsługi magistrali CAN w wersji 2.0B active (tzw. beCAN). Do obsługi magistrali w procesorze dostępne są:

- bufony sprzętowe używane do przechowywania danych nadawanych i odbieranych (rodzaj kolejki FIFO);
- filtry określające grupy akceptowalnych pakietów;
- układ nadawczy z programowalną szybkością transmisji (do 1 Mb/s);
- dedykowane przerwanie do potrzeb magistrali CAN;
- tryby pracy z oszczędnością energii.

Odbiór i nadawanie danych z użyciem wspomnianego procesora realizowane jest automatycznie po odpowiednim skonfigurowaniu procesora (zwłaszcza określeniu miejsca położenia buforów do przechowywania danych). W bardziej zaawansowanych systemach można posłużyć się procesorem z rdzeniem 16-bitowym z wbudowanymi mechanizmami obsługi magistrali CAN (np. ST92F150xx czy ST10F269x firmy STM). Do sprzężenia linii magistrali CAN z liniami kontrolera można zastosować:

- układ L9669 - nadajnik/odbiornik magistrali CAN;
- układ L4969 - regulator napięcia z układem nadajnika/odbiornika magistrali CAN (w wersji low-speed) z wbudowanym układem watchdog.

Do sterowania elementami wykonawczymi można wykorzystać:

- układ L9950 - sterownik serwo-mechanizmów lusterek zewnętrznych, siłowników centralnego zamka i oświetlenia drzwi.
- układ L9951 - sterownik siłowników centralnego zamka i oświetlenia drzwi.

Protokół LIN (Local Interconnect Network)

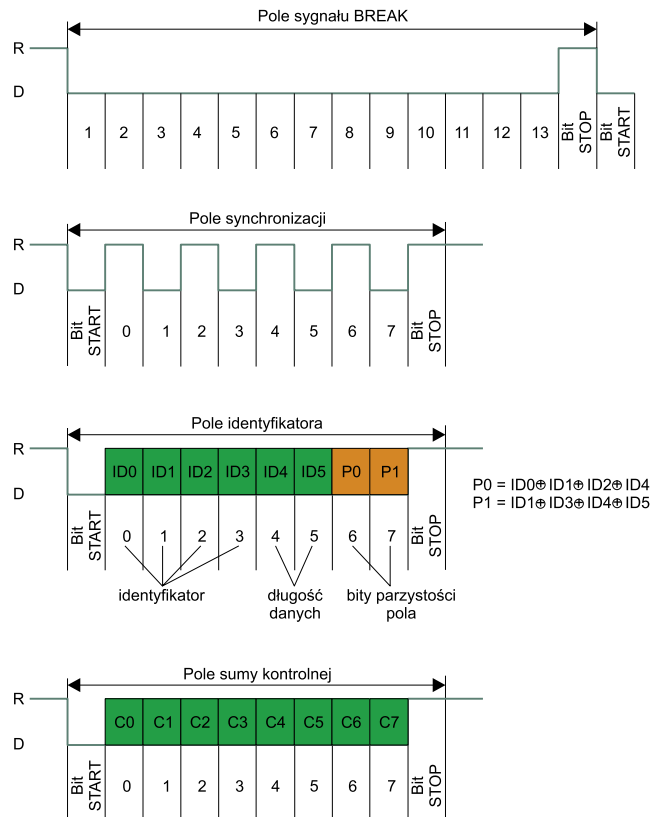
Protokół LIN stanowi tańszą alternatywę protokołu CAN. Wymaga użycia pojedynczego przewodu do transmisji danych za co płaci się niższymi prędkościami transmisji (do 20 kb/s). Magistrala LIN wykorzystywana jest jako podsieć w elektronicznych systemach sterowania.

Podstawowe parametry protokołu:

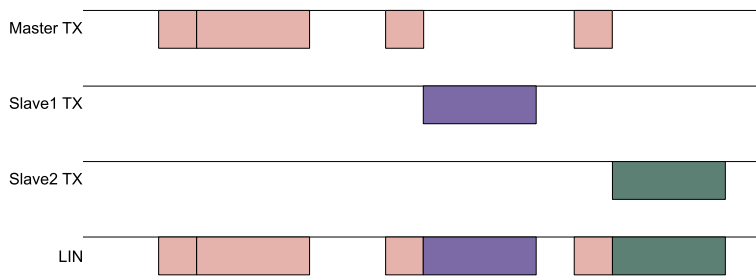
- magistrala szeregowo asynchroniczna jedynoprzewodowa;
- dane transmitowane w kodzie NRZ (Non Return to Zero). Poziomy logiczne podobnie jak w CAN nazywane są tu dominant (0) i recessive (1). W

dalszej części niniejszego opracowania stosowane będą oznaczenia skrótowe (odp. D i R);

- podłączenia do magistrali typu iloczyn na drucie (wired-AND) co powoduje, że spoczynkowym stanem magistrali jest stan R, natomiast w celu zmiany stanu magistrali na D wystarczy zmiana stanu na linii w jednym z podłączonych do niej urządzeń;
- każde urządzenie podłączone do magistrali dosynchronizowuje się do ciągu synchronizującego wysyłanego w nagłówku ramki wysyłanej przez urządzenie inicjujące transmisję;
- architektura typu Master-Slave. Do magistrali jest dołączone jedno urządzenie nadrzędne (Master), mogące inicjować transmisję oraz wiele urządzeń podrzędnych (Slave) mogących odbierać i wysyłać dane na żądanie urządzenia nadrzędnego.
- transmisje na magistrali są przesyłane w postaci ramek. Każda ramka składa się z nagłówka (zawierającego ciąg synchronizujący, identyfikator wiadomości) wysyłanego zawsze przez urządzenie nadrzędne oraz z właściwych danych zakończonych sumą kontrolną wysyła-



Rys. 7.



Rys. 8.

nych przez urządzenie nadrzędne lub przez urządzenie podrzędne na żądanie nadrzędnego.

- bezpośrednia komunikacja pomiędzy modułami Slave nie jest możliwa.
- komunikacja na magistrali podobnie jak w protokole CAN zorientowana jest obiektowo. Każda ramka zawiera identyfikator określający przesyłane dane. Każde urządzenie ma określoną grupę akceptowalnych identyfikatorów i pobiera tylko te, które są do niego przeznaczone.
- zaimplementowane mechanizmy sum kontrolnych i kontroli parzystości gwarantują bezpieczeństwo i detekcję błędów.

Na rys. 5 przedstawiono sposób dołączenia magistrali LIN do systemu z CANbusem.

W protokole LIN obsługiwany jest jeden format ramki. Składa się ona z dwóch części:

- nagłówka – wysyłanego tylko przez Mastera. Część ta zawiera dane synchronizujące, identyfikator przesyłanej wiadomości.
- danych – wysyłanego przez Mastera lub inny moduł na żądanie (podane w nagłówku bieżącej ramki).

Opis budowy ramki w protokole LIM przedstawiono na rys. 6. Na rys. 7 przedstawiono budowę i rolę poszczególnych pól ramki:

- Sygnał BREAK – wykorzystywany do sygnalizacji rozpoczęcia transmisji nowej ramki.
- Pole synchronizacji – wykorzystywane jest do zsynchronizowania wszystkich urządzeń podłączonych do magistrali LIN.
- Pole identyfikatora – zawiera długość danych, identyfikator oraz własną sumę kontrolną. Możliwe jest określenie do 64 identyfikatorów.
- Pole sumy kontrolnej służy kontroli poprawności przesyłanych danych. W pole to wpisywa-

ny jest najmłodszy bajt wyniku zanegowanej sumy wszystkich bajtów danych.

Przebieg transmisji

Każdą transmisję można podzielić na dwie fazy:

- Fazę inicjacji transmisji, podczas której urządzenie nadrzędne (Master) wysyła nagłówek wiadomości zawierający dane do synchronizacji wszystkich modułów podłączonych do magistrali oraz identyfikator określający grupę wiadomości. W tej fazie każdy z modułów Slave podłączonych do magistrali dosynchronizowuje się do przesyłanej ramki. Następnie każdy Slave analizuje przesyłany identyfikator i sprawdza czy należy on do grupy akceptowalnych przez dany moduł.
- Fazę transmisji danych, podczas której:
 - Master przesyła dane/polecenia wraz z sumą kontrolną do modułu/modułów Slave;
 - Slave przesyła do Mastera żądane dane wraz z sumą kontrolną.

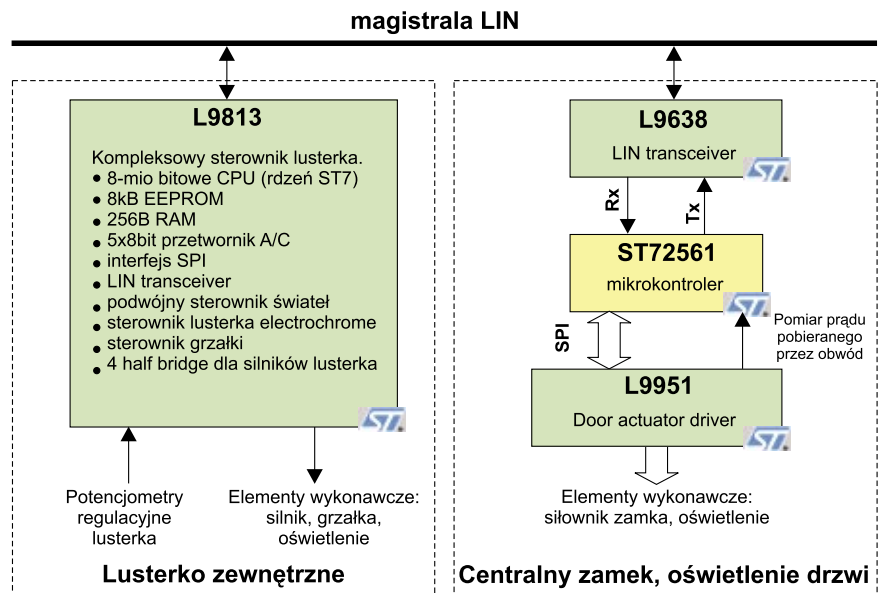
Na rys. 8 przedstawiono sposób przebiegu transmisji na magistrali. W protokole nie zdefiniowano mechanizmu potwierdzenia transmisji ani zgłaszania błędów na magistrali.

W celu osiągnięcia bardzo niskiego poboru mocy, jest możliwe „uśpienie” urządzeń dołączonych do magistrali. W tym celu moduł Master wysyła ramkę „usypiającą”. W stanie uśpienia na magistrali utrzymywany jest stan wysoki. W celu obudzenia magistrali dowolny moduł może przesłać sygnał budzący (chwilowe wymuszenie na magistrali LIN stanu D).

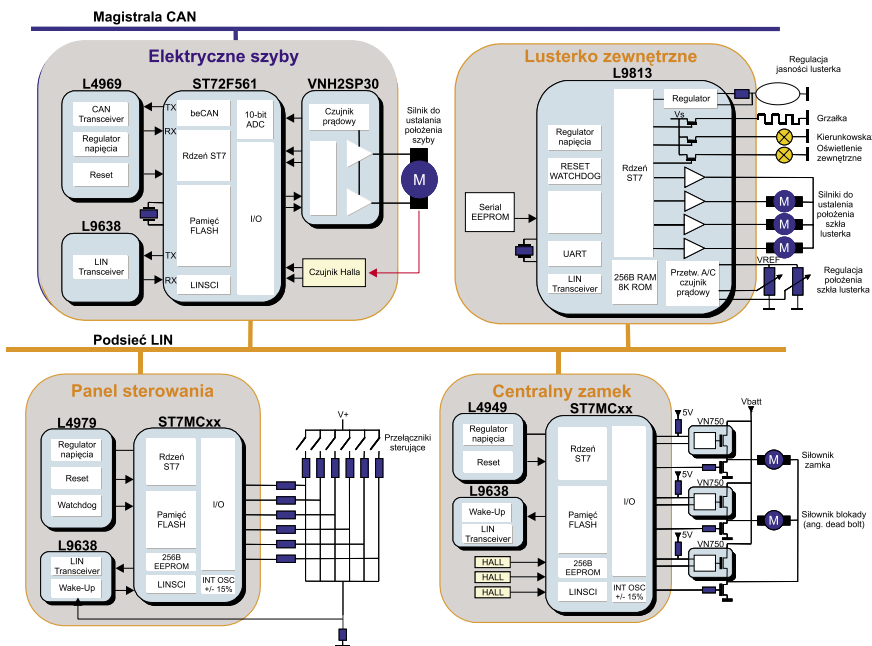
Wskazówki praktyczne

Prostota protokołu umożliwia wykorzystanie do jego realizacji ogólnie dostępnych układów. Najprostszym i najtańszym sposobem zrealizowania modułu jest użycie układu transceivera magistrali LIN (np. L9638 firmy ST) do sprzężenia poziomów magistrali LIN 0, +12 V z poziomami cyfrowymi w połączeniu z 8-bitowym mikrokontrolerem wyposażonym w sprzętową obsługę magistrali szeregową (UART, SCI, LINSICI) oraz elementami sterującymi w postaci układów półmostków i mniej lub bardziej specjalizowanych sterowników elementów wykonawczych.

Największy problem stanowi implementacja protokołu w mikroprocesorach ogólnego przeznaczenia z wykorzystaniem standardowych szeregowych interfejsów asynchronicznych. Większość elementów protokołu (związanych z synchronizacją, detekcją i analizą nagłówka obsłu-



Rys. 9.



Rys. 10.

gą trybu oszczędzania energii) musi zostać zrealizowana programowo. Alternatywą dla tego rozwiązania może być wykorzystanie mikrokontrolera z wbudowanym rozszerzonym interfejsem szeregowym. Przykładem takiego mikrokontrolera jest 8-mio bitowy ST72561 firmy STM. Układ ten posiada rozszerzony szeregowy interfejs asynchroniczny LINSICI posiadający sprzętowe mechanizmy wspomagające współpracę z magistralą LIN. Dodatkowo układ ten posiada wbudowane sprzętowe mechanizmy obsługi magistrali CAN co czyni go dobrym kandydatem na moduł Mastera magistrali LIN stanowiącego pomost pomiędzy magistralą CAN i LIN.

Na rys. 9 przedstawiono porównanie rozwiązania modułu składowego magistrali LIN bazującego na wykorzystaniu standardowego szeregowego interfejsu asynchronicznego i modułu rozwiązanego przy wykorzystaniu interfejsu rozszerzonego LINSICI.

Rozwiązanie klasyczne:

- Obsługa protokołu (analiza nagłówka wiadomości, synchronizacja, obsługa trybów oszczędności energii) w większości musi być realizowana programowo.
- Ograniczenie długości sygnału BREAK do 11 bitów – brak

możliwości realizacji modułu Master dla magistrali LIN (wymagającego możliwość generacji 13-to bitowego sygnału BREAK). Rozwiązanie bazujące na LINSICI (np. mikrokontroler ST72561, ST7MCxx)

- Możliwość generacji sygnału BREAK o długości 11 lub 13 bitów, co umożliwia wykorzystanie modułu w roli Mastera magistrali LIN.
- Automatyczny odbiór i analiza nagłówka.
- Automatyczna synchronizacja do pola synchronizacji w nagłówku wiadomości.
- Obsługa procedur wchodzenia i wychodzenia do/z trybów oszczędzania energii (możliwość wybudzania modułu po wykryciu opadającego zbocza na linii Rx).

CAN i LIN w praktyce

We współczesnych instalacjach samochodowych średniej i wyższej klasy najczęściej spotyka się rozwiązanie łączące interfejs CAN jako główną magistralę sterującą bezpośrednio elementy wymagające obsługi w czasie rzeczywistym i podsieci bazujące na standardzie LIN.

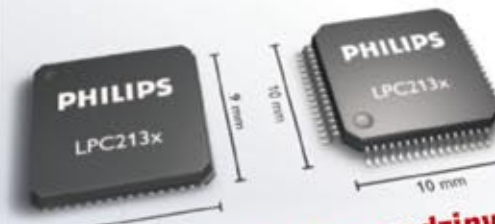
Na rys. 10 przedstawiono przykład kompleksowego rozwiązania sterowania elementami wykonawczymi drzwi samochodowych wykorzystującego magistralę CAN i LIN.

Na podstawie materiałów firmy STM opracował:

Artur Iwanicki, STM

LPC213x

Nowe ARM-y firmy



Nowe mikrokontrolery z rodziny LPC2000 już dostępne! Procesory 32-bitowe w cenie 8-bitowych!

Cechy charakterystyczne nowych mikrokontrolerów:

- rdzeń ARM7TDMI,
- częstotliwość taktowania rdzenia: 60 MHz,
- liczba uniwersalnych linii I/O: 47,
- pamięć programu Flash z możliwością programowania w systemie,
- pojemność pamięci Flash: 32...512 kB w zależności od modelu,
- pojemność pamięci SRAM: 8...32 kB w zależności od modelu,
- zasilanie: 3,3 V (pojedyncze),
- wejścia kompatybilne z logiką 5-woltową,
- wbudowany 10-bitowy przetwornik C/A z wyjściem napięciowym,
- zaawansowane moduły peryferyjne: 2*UART, 2*I²C, 2*SPI, WDT, RTC z wydzielonym zasilaniem, 10-bitowe przetworniki A/C, PWM,
- obudowy: LQFP64 i ultra miniaturowe HVQFN64.



50-071 Wrocław, Plac Wolności 7
tel. (71) 783 12 60
fax (71) 783 12 69

info.poland@eurodis.com

41-200 Sosnowiec, ul. Kiepury 45/1
tel. (32) 291 99 35
fax (32) 296 90 80

info.poland@eurodis.com

www.eurodis.com
Twój dostawca
elementów elektronicznych