

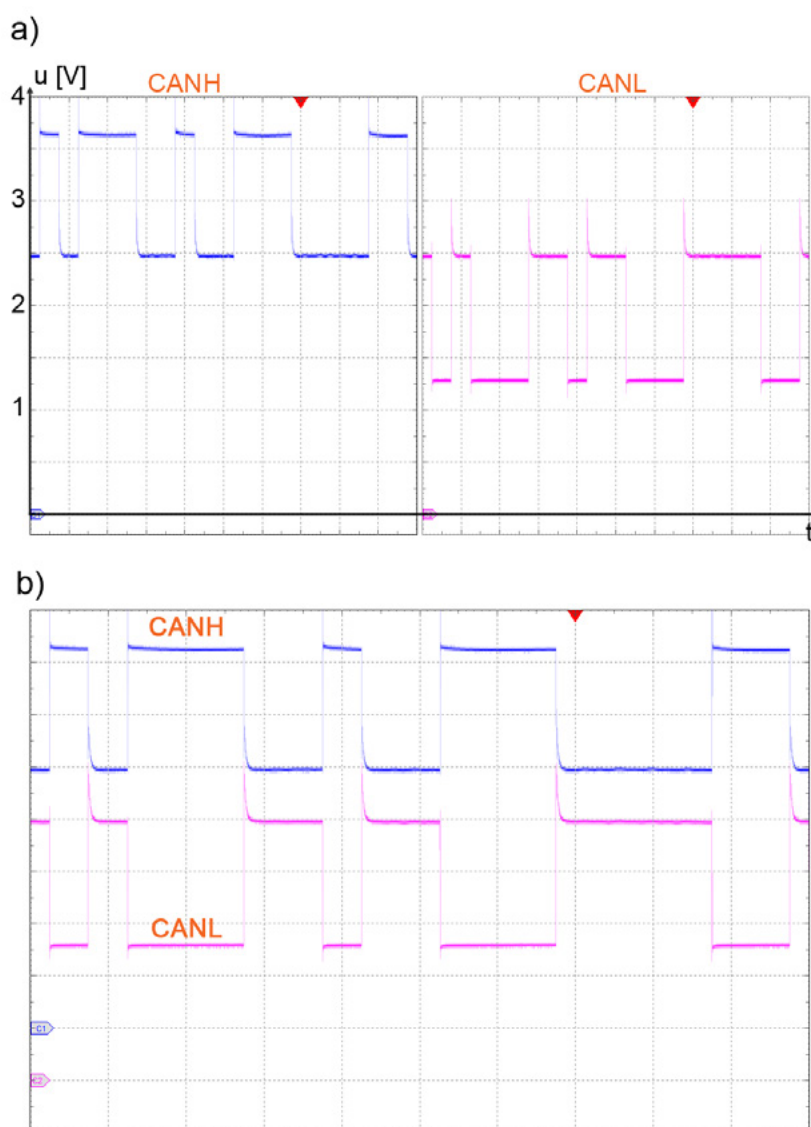
# Analizowanie protokołów szeregowych oscyloskopami Rohde&Schwarz (3)

## CAN, LIN

Badanie protokołów komunikacyjnych jest już obowiązkową funkcją oscyloskopów cyfrowych co najmniej średniej klasy. Nadal jednak jest ona udostępniana najczęściej jako opcjonalne rozszerzenie oprogramowania firmowego. Zwykle w cenie oscyloskopu mieści się tylko kilka najbardziej popularnych protokołów, np. SPI i UART. W cyklu artykułów przedstawiono rozwiązania dotyczące analizy protokołów zastosowane w oscyloskopach Rohde&Schwarz. W tej części omówiono badanie interfejsów CAN i LIN.

Omawiane w poprzedniej części artykułu interfejsy SPI i I<sup>2</sup>C pracują według modelu komunikacji Master – Slave. W tak zorganizowanym systemie jakkolwiek transmisję między dowolnymi urządzeniami zawsze inicjuje urządzenie nadrzędne – Master, natomiast urządzenia podrzędne (Slave) odpowiadają tylko na pytania i nie mogą porozumiewać się między sobą bez pośrednictwa urządzenia Master. W wielu aplikacjach przedstawione rozwiązania są jednak nie do przyjęcia. Przykładem jest elektronika motoryzacyjna, w której komunikacja musi być w wielu przypadkach natychmiastowa, bez metody pytań i odpowiedzi.

Między innymi, na potrzeby motoryzacji w latach osiemdziesiątych XX wieku opracowano specjalny interfejs komunikacyjny CAN, który (co prawda po kilku ulepszeniach i modyfikacjach) jest z powodzeniem stosowany do dziś. Obecnie zakres zastosowań wykraczał poza motoryzację – do automatyki przemysłowej. Na dokładny opis warstwy fizycznej i protokołu nie ma miejsca w tym artykule, odsyłamy do bogatej dokumentacji dostępnej z wielu źródeł. Poniżej przedstawiono jedynie najważniejsze zagadnienia niezbędne do zapoznania się z pomiarami wykonywanymi oscyloskopami Rohde&Schwarz.



Rysunek 1. Oscylogramy napięć na magistrali CAN, a) zależności napięciowe, b) zależności czasowe

### Protokół CAN

Zasadniczą różnicą między interfejsem CAN a omawianymi wcześniej SPI i I<sup>2</sup>C jest przyjęcie odmiennego modelu komunikacji, tzw. *Multi-master*. Wszystkie węzły mogą w nim komunikować się ze sobą wzajemnie bez stosowania dodatkowych linii sterujących, tylko korzystając z podstawowego medium, jakim jest dwuprzewodowa skrętka. Jest ona terminowana na obu końcach rezystorami ok. 120 Ω dla zminimalizowania odbić sygnału. Węzły są łączone w konfiguracji liniowej, gwiazdистой lub pierścieniowej. Sygnał logiczny jest tworzony różnicowo przez odjęcie napięcia CANL od CANH (CANL i CANH to nazwy linii interfejsu CAN). Przykładowe relacje między napięciami na rysunku 1a, relacje czasowe są natomiast

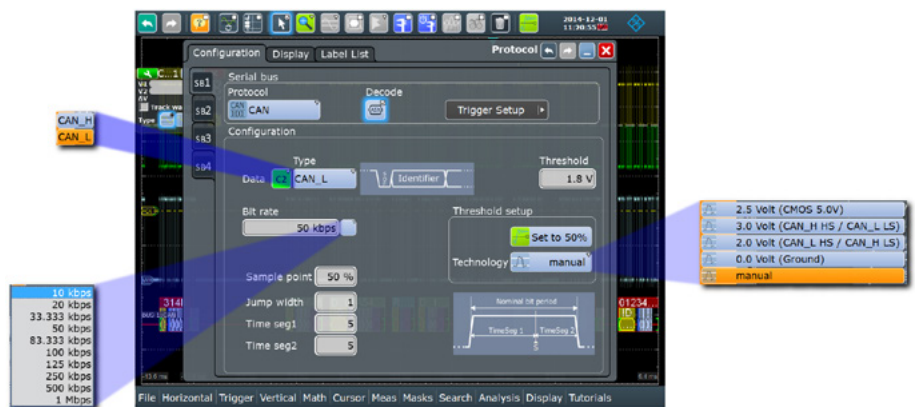
widoczne na **rysunku 1b**. Analizator protokołów odczytuje stan magistrali na podstawie tylko jednej linii. Sygnał z linii CANL jest interpretowany wprost, z linii CANH natomiast po zanegowaniu. Oba sygnały są jednak potrzebne do realizacji transmisji różnicowej umożliwiającej przesyłanie danych na duże odległości. Przy prędkości transmisji 5 kb/s możliwe jest uzyskanie zasięgu nawet do 10 km.

Założenia dotyczące sieci komunikacyjnej z zastosowaniem interfejsu CAN narzuciły budowę ramki transmisyjnej. Z uwagi na możliwość zgłaszania jednoczesnego dostępu do medium przez kilka węzłów warstwa fizyczna i protokół komunikacyjny musi uwzględniać rozstrzyganie konfliktów, sygnalizację błędów, przesyłanie potwierżeń itp. Dobry analizator protokołu powinien uwzględniać wszystkie zdarzenia występujące na magistrali, niezależnie od wersji badanego interfejsu – pamiętamy, że ulegał on modyfikacjom.

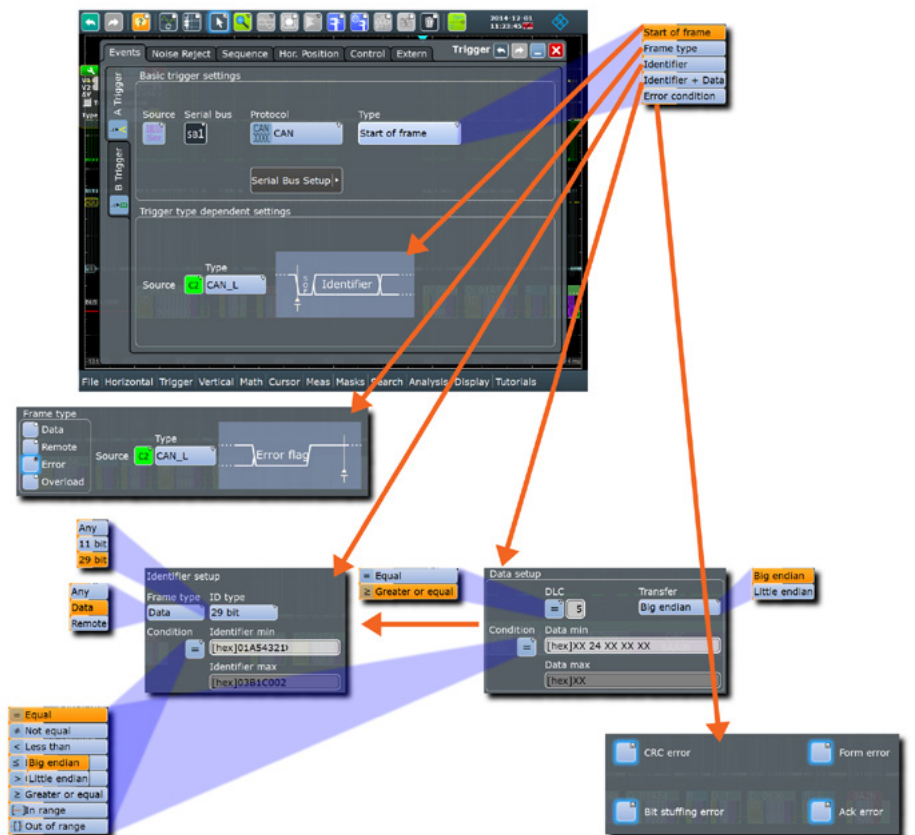
Ramkę protokołu CAN przedstawiono na **rysunku 2**. Składa się ona z kilku pól pełniących określone role w interpretacji zdarzeń występujących na magistrali. W wyniku prac nad interfejsem powstały dwie funkcjonujące dziś wersje protokołu: CAN 2.0A (tzw. standard – rys. 2b) i CAN 2.0B (extended – rys. 2a). Zasadniczą różnicą z punktu widzenia protokołu są inne długości bloku identyfikacyjnego, zwanego też polem arbitrażowym. Dla użytkownika bardziej istotna jest większa prędkość transmisji w interfejsie CAN 2.0B dochodząca do 1 Mb/s, podczas gdy w wersji CAN 2.0A można uzyskać tylko 125 kb/s. W protokole przewidziano transmisję czterech rodzajów ramek, są to: ramka danych, ramka zdalnego wywołania, ramka błędów i ramka przepełnienia. Transmisja jest w zasadzie asynchroniczna, w interfejsie nie ma linii zegarowej. Lokalne zegary po stronie odbiorczej są synchronizowane z nadawczymi zawsze na początku ramki. Transmisję inicjuje dowolny węzeł, a ewentualne konflikty są rozstrzygane poprzez przemyślany system arbitrażowy. Wykrycie ewentualnych błędów przez dowolny węzeł skutkuje natychmiastowym generowaniem ramki błędów, co jest informacją do nadawcy, że powinien powtórzyć ostatnią transmisję. Błędy mogą wynikać nie tylko z konfliktów na magistrali, ale też ze zwykłych przekłamań sygnału elektrycznego. Dlatego w każdej transmisji jest obliczana 15-bitowa suma kontrolna (CRC), którą teoretycznie można by było wykorzystać do korekcji pewnych błędów, jednak służy ona tylko do informowania o błędach. Co ciekawe, węzeł nadający może sam wykrywać swoje błędy dzięki równoczesnemu nasłuchowi magistrali podczas nadawania. Ramki nie mogą być nadawane bezpośrednio jedna



**Rysunek 2. Ramka protokołu CAN a) wersja z 29-bitowym identyfikatorem, b) wersja z 11-bitowym identyfikatorem**



**Rysunek 3. Zakładka konfiguracyjna dla protokołu CAN**



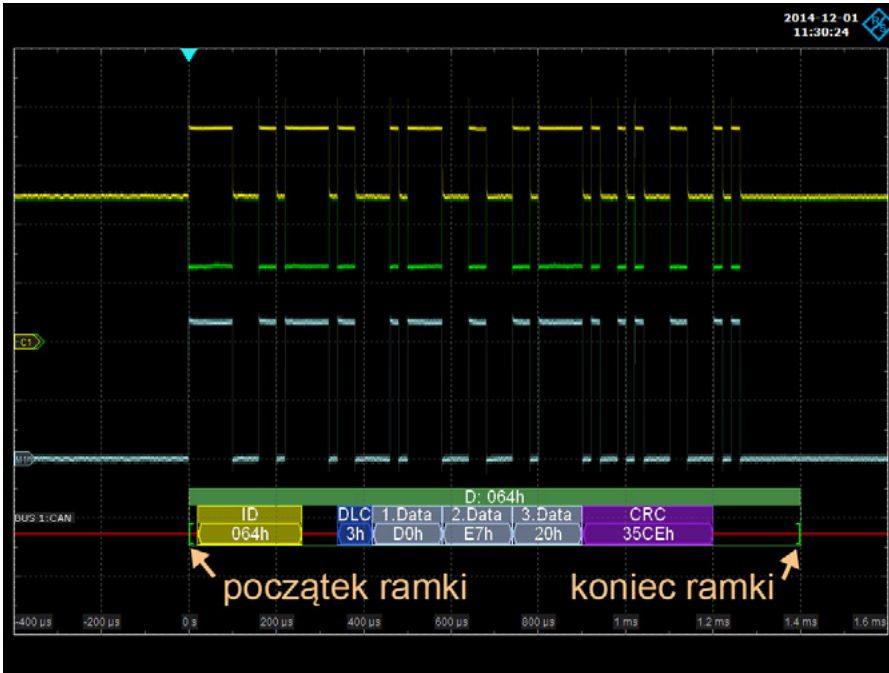
**Rysunek 4. Zakładka wyboru zdarzeń wyzwalających dla protokołu CAN**



po drugiej. Minimalny odstęp między kolejno nadawanymi ramkami powinien być nie mniejszy niż 3 bity. Koncepcja protokołu CAN zakłada, że o wszystkim co dzieje

się na magistrali wiedzą wszystkie węzły. Szczególnym rodzajem ramki protokołu CAN jest tzw. ramka zdalna, w której nie występuje w ogóle pole danych.

Konieczne jest jeszcze wyjaśnienie trochę odmiennego nazewnictwa bitów od ogólnie stosowanego. Ze względu na różnicową transmisję sygnału cyfrowego, w interfejsie CAN, zasadniczo nie stosuje się określeń „niski poziom” lub „wysoki poziom”, a zamiast nich wprowadzono pojęcie bitu dominującego i recesywnego. Sygnały cyfrowe są generowane przez wyjścia typu *Open Drain*, w jakie są wyposażone drivery linii. Stan niski jest utożsamiany z napięciem występującym na wyjściu, w którym został uaktywniony tranzystor wyjściowy – jest to stan dominujący, ponieważ wystarczy, aby tylko jedno wyjście zostało uaktywnione, by linia znalazła się w tym stanie. Przy wyłączonych wyjściach linia pozostaje w stanie recesywnym (odpowiednikiem stanu wysokiego). Zauważmy, że z elektrycznego punktu widzenia nie ma konfliktu, jeśli wyjścia kilku węzłów jednocześnie zostaną wysterowane, natomiast przyjęta metoda generacji stanów logicznych na liniach interfejsu CAN jest wykorzystywana do rozstrzygania przypadków próby jednoczesnego dostępu do linii przez kilka węzłów. Szczegółów należy szukać w specyfikacji interfejsu CAN.



Rysunek 5. Przykładowe wyzwolenie na początku ramki



Rysunek 6. Przykładowe wyzwolenie na różnych typach ramek, a) ramka danych, b) ramka zdalna, c) ramka błędu, d) ramka przepiętowania

**Pomiary interfejsu CAN**

Zakładkę konfiguracyjną protokołu CAN przedstawiono na **rysunku 3**. Do odczytania informacji przesyłanych interfejsem CAN wystarczy analiza jednej z jego linii: CAN\_L lub CAN\_H. Do interpretacji danych cyfrowych na podstawie linii CAN\_H konieczna będzie inwersja poziomów. W tym przypadku wyższe napięcie będzie odpowiadało stanowi logicznemu „0”, niższe natomiast będzie odpowiadało „1”. Oczywiście analizator wykonuje taką operację automatycznie podczas pracy, jednak bardziej oczywista dla użytkownika będzie częściej stosowana konwencja, w której wyższe napięcie odpowiada „1”, niższe „0”. Z tego względu do analizy wygodniej jest brać linię CAN\_L. Zawsze bardzo istotne jest prawidłowe ustawienie napięcia progowego (pole Threshold”). Należy pamiętać, że po każdej zmianie linii CAN\_L na CAN\_H lub odwrotnie, konieczna będzie korekcja napięcia progowego. Wyboru napięcia progowego można dokonać na podstawie określenia technologii wykonania interfejsu. Po naciśnięciu przycisku „Technology” zmieniana jest nastawa z ręcznej na jedną z kilku predefiniowanych (rys. 3).

Kolejnym parametrem decydującym o prawidłowym przebiegu analizy protokołu jest zastosowana w badanym układzie prędkość

transmisji. Parametr ten może być wpisywany ręcznie w polu „Bit rate” lub wybierany z listy.

Jak już było powiedziane, transmisja realizowana interfejsem CAN jest asynchroniczna, ale synchronizowana na początku każdej ramki. Bardzo groźne są długie sekwencje jednakowych bitów, gdyż nie występują wówczas momenty znamienne i może wówczas dochodzić do rozsynchronizowania się odbiornika, więc w protokole CAN uwzględniono zabezpieczenie mające wykluczyć podobne przypadki. Jeśli w transmisji pojawia się pięć jednakowych impulsów, to generowany jest dodatkowy impuls o przeciwnym znaku. Może to prowadzić do wydłużania poszczególnych pól ramki, np. danej do 9 bitów.

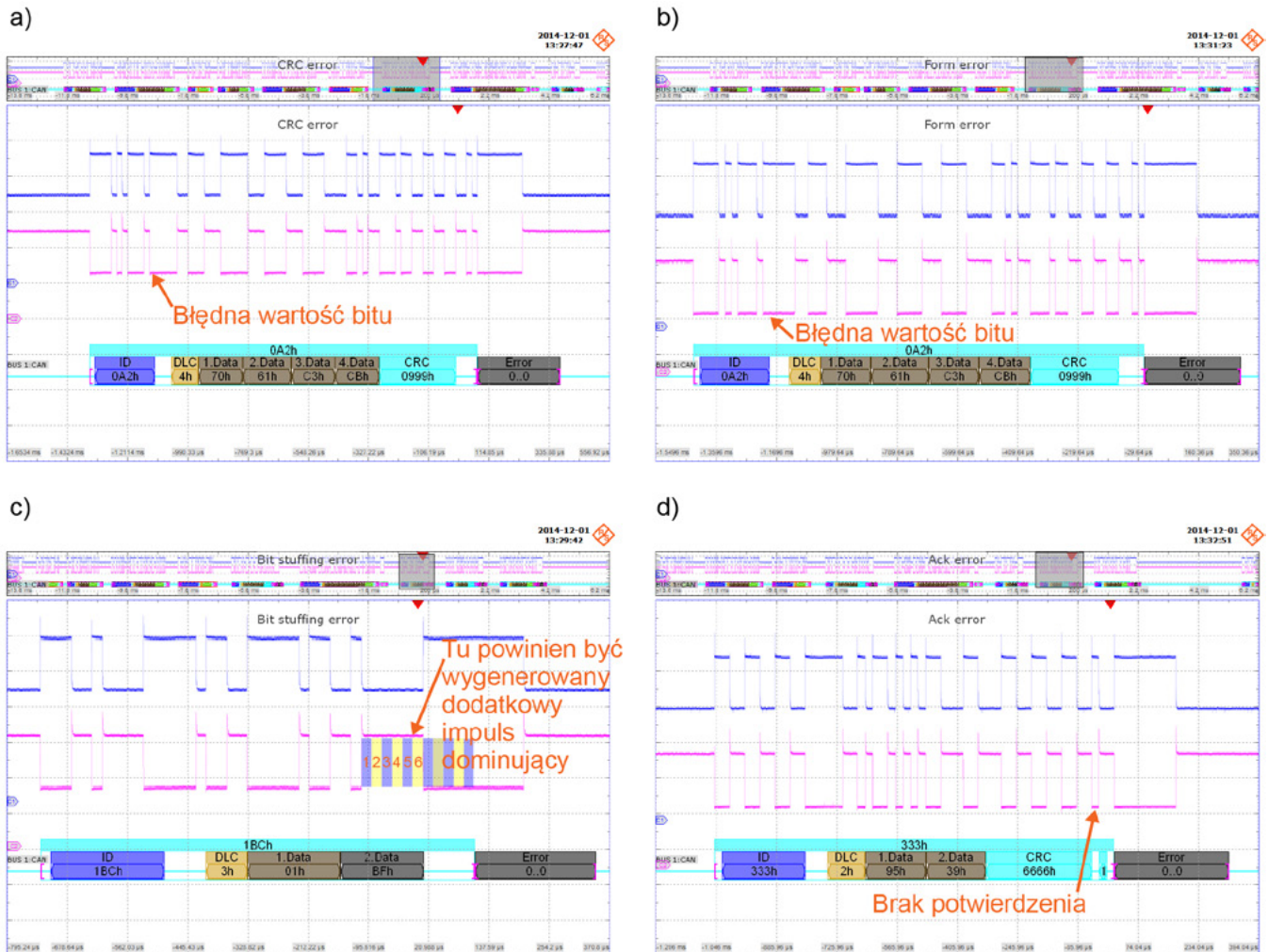
Teoretycznie, stany impulsów są odczytywane w połowie ich szerokości, ale czasami, np.

na skutek różnych zniekształceń sygnału w zastosowanym medium korzystne może okazać się przesunięcie punktu próbkowania. Należy wówczas zmienić parametr „Sample Point”, którego domyślną wartością jest 50%.

Po skonfigurowaniu protokołu można już przystąpić do pomiarów. Ich przebieg zależy w dużym stopniu od doboru trybu wyzwala-  
nia (**rysunek 4**). Podczas analizowania pracy



**Rysunek 7.** Oscylogram uzyskany po wyzwoleniu na ramce o identyfikatorze 01A54321h zawierającej daną 24h występującą na drugim bajcie pola danych

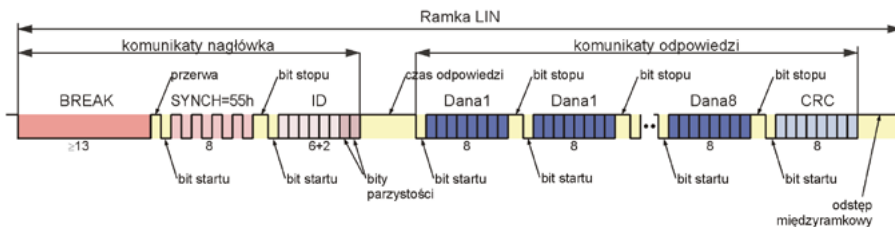


**Rysunek 8.** Wyzwalanie na błędach, a) CRC, b) ramki, c) kodowania, d) braku potwierdzenia

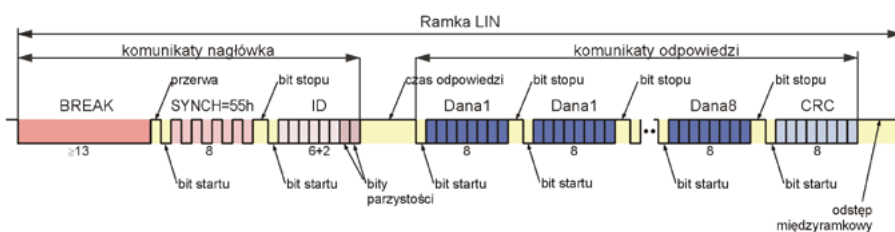




Rysunek 9. Magistrala LIN



Rysunek 9. Magistrala LIN



Rysunek 10. Ramka protokołu LIN

interfejsu CAN ma to szczególne znaczenie, gdyż rozróżnianych jest tu wiele różnych charakterystycznych zdarzeń. Pierwszym, najprostszym zdarzeniem wyzwalającym jest identyfikacja początku ramki. Na ekranie moment ten jest zaznaczany znakiem zielonego otwierającego nawiasu prostokątnego (rysunek 5). Analogicznie koniec ramki jest oznaczany nawiasem zamykającym. Przydatność tego rodzaju wyzwalania ogranicza się praktycznie do przypadków, w których transmisje są inicjowane sporadycznie, przy ciągłym ruchu na magistrali obraz nie będzie stabilny.

Nieco skuteczniejsze jest wyzwalanie na określonym typie ramki. Jeśli podejrzewamy, że w układzie są generowane błędy, najlepszą metodą analizy będzie zastosowanie wyzwalania po wykryciu ramki błędu. Przyczyn generowania błędów jest kilka, na przykład brak dodatkowego bitu o przeciwnym stanie po przesłaniu pięciu bitów jednakowych. O błędach będzie jeszcze mowa w dalszej części artykułu. Na rysunku 6 przedstawiono przykłady ramek różnego typu, z jakimi można się spotkać badając interfejs CAN.

Do poszukiwania konkretnych danych najlepiej nadają się opcje: „Identifier” i „Identifier + Data”. Pierwsza z nich powoduje wyzwolenie oscyloskopu po zlokalizowaniu ramki o określonym identyfikatorze. Próbuując ręcznie rozszyfrować dane w polu ID trzeba pamiętać, że identyfikator 29-bitowy jest rozdzielony w środku bitami SRR i IDE oraz, że w przypadku występowania co najmniej 5-bitowych ciągów jednakowych bitów pole to może być wydłużone o dodatkowo generowane impulsy synchronizujące. Na rys. 7 przedstawiono

wynik poszukiwania ramki o identyfikatorze 01A54321h zawierającej daną 24h występującą na drugim bajcie pola danych. Opcje wyzwalania zostały dobrane zgodnie z rys. 4.

Wracamy jeszcze do błędów. Pomiar analizatorem protokołów może być wyzwolony określonym rodzajem błędu, nie tylko przesłaniem samej ramki błędu. Stwarza to możliwość zawężania podejrzeń dotyczących źródła błędów w nieprawidłowo funkcjonującym urządzeniu. Po wybraniu opcji „Type=Error condition” należy zaznaczyć, które z czterech rodzajów błędów mają być rozpatrywane. Są to: błąd sumy kontrolnej (CRC), błąd formatu ramki (Form) występujący po stwierdzeniu niepoprawnego stanu znanych bitów, błąd kodowania (*Bit stuffing*) generowany po wystąpieniu w interfejsie więcej niż pięciu kolejnych jednakowych bitów i błąd potwierdzenia (Ack) generowany, gdy odbiornik nie wystawi bitu dominującego („0”) oznaczającego potwierdzenie odczytania ramki. Każdy z tych błędów przedstawiono na rysunku 8. Zdarza się, że jeden błąd może generować kolejne. Na rys. 8c został wykryty błąd kodowania. Po pięciu następujących po sobie bitach dominujących nie został wstawiony bit recesywny, co natychmiast zostało wykryte i zasygnalizowane jako błąd „Bit stuffing”.

### Protokół LIN

Jak mogliśmy się przekonać, interfejs CAN jest bardzo dobrym rozwiązaniem do realizacji względnie szybkiej transmisji danych pomiędzy równoprawnymi węzłami. Tak rozbudowany protokół nie jest jednak zawsze konieczny, na przykład odczytywanie

stanu czujników rozmieszczonych w samochodzie przez komputer pokładowy można zrealizować znacznie prościej. Do takich potrzeb opracowano interfejs LIN (Local Interconnect Network), stanowiący uzupełnienie CAN-a. Daje się go zrealizować znacznie prostszymi środkami, wystarczy do tego celu nawet tani mikrokontroler dysponujący interfejsem UART. W specyfikacji LIN można doszukać się podobieństw z poznanymi już protokołami UART i I<sup>2</sup>C, jednak istotną różnicę stanowi wykorzystanie tylko jednej linii transmisyjnej. Do wad należy zaliczyć dość niewielką prędkość transmisji, dochodzącą zaledwie do 20 kb/s. Przy tak małych prędkościach nie ma jednak problemów z zachowaniem synchronizacji, nawet przy zastosowaniu prostych oscylatorów. Do ich budowy nie są nawet konieczne rezonatory kwarcowe, wystarczą ceramiczne. Transmisja jest asynchroniczna synchronizowana na ramkach.

Protokół LIN, podobnie jak SPI czy I<sup>2</sup>C, wymaga podziału kompetencji wśród dołączonych urządzeń. W tzw. klastrze dopuszcza się występowanie co najwyżej 16 węzłów, tworzy go więc jeden Master (obowiązkowo) oraz do 15 Slave'ów. Po wyposażeniu Slave'a w zwielokrotniony interfejs fizyczny może on jednak pracować w kilku klastrach. Master realizuje zarówno swoje zadania (master task), jak i zadania Slave'a (slave task). Magistrala LIN ma budowę jak na rysunku 9.

Ramka protokołu LIN składa się z nagłówka będącego zadaniem Mastera i odpowiedzi generowanej przez Slave'a (rysunek 10). Z kolei w nagłówku wyróżnia się komunikaty (pola) „Synch Break”, „Synch Field” i „Ident Field”. Komunikat „Synch Field” zawiera sekwencję synchronizującą informującą urządzenia o prędkości transmisji. Poprzedza go przerwa synchronizująca trwająca co najmniej 13 bitów. Komunikat „Ident Field” zawiera identyfikator wiadomości, po którym dołączone do magistrali urządzenia rozpoznają czy przesyłane po nagłówku dane są adresowane do nich czy do innych urządzeń. W poprawnie zaprojektowanej magistrali LIN nie występują konflikty, ponieważ wszystkie węzły Slave tylko odpowiadają na zapytania Mastera. Nie przewidziano arbitrażu na przykład w przypadkach jednoczesnej odpowiedzi kilku Slave'ów. Jeśli tak się stanie, musi być sygnalizowany błąd. Do wykrywania błędów wykorzystywana jest m.in. suma kontrolna przesyłana w każdym polu danych. Ponadto każdy nadajnik na bieżąco sprawdza czy stan linii odpowiada wysyłanej informacji i reaguje natychmiast, jeśli wykryje niezgodność.

Wszystkie komunikaty oprócz synchronizujących mają budowę przypominającą ramkę protokołu UART. Składają się z bitu startu, 8 bitów informacyjnych i bitu stopu. Dane są przesyłane od najmłodszego

do najstarszego bitu. Jak widać, nie ma tu wyróżnionego bitu parzystości, ale w polu identyfikacyjnym przekazującą informację o nadawcy, odbiorcy (jednym lub wielu), celu ramki i długości pola danych, dwa ostatnie bity pełnią taką funkcję. Na samą treść pozostaje 6 bitów. Znając już cel transmisji określany przez Mastera w identyfikatorze, odpowiednie urządzenia Slave przesyłają odpowiedź składającą się z pola danych (1, 2, 4 lub 8 bajtów) i sumy kontrolnej.

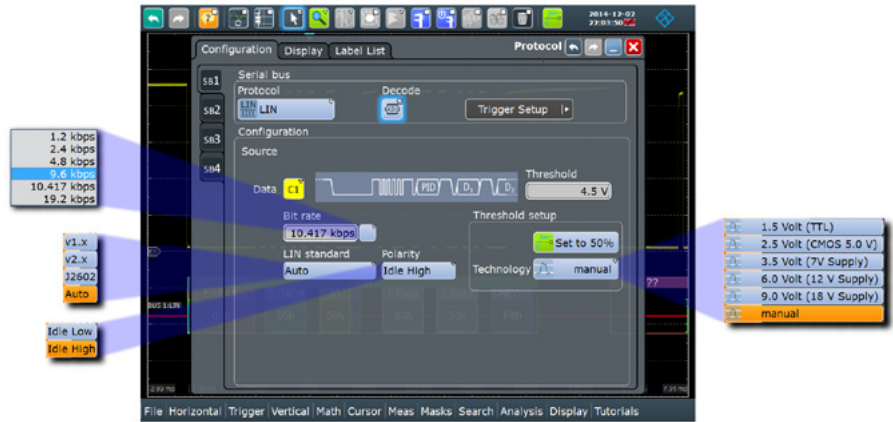
Specyfikacja LIN zakłada ponadto możliwość usypiania urządzeń dołączonych do magistrali. Jest to realizowane przez przesłanie ramki usypiającej. Wyłączenie wyjść wszystkich urządzeń dołączonych do magistrali powoduje, że na linii panuje stan recesywny. Urządzenia budzą się po wystawieniu stanu dominującego (o czasie trwania odpowiadającym 8 bitom) przez dowolny węzeł dołączony do magistrali. Sygnał budzenia musi wyprzedzać najbliższą ramkę o czas odpowiadający 4...64 bitów.

**Pomiary interfejsu LIN**

Jak zwykle pomiary rozpoczynamy od ustawienia parametrów badanego interfejsu, w tym przypadku LIN. Po naciśnięciu przycisku *PROTOCOL* na ekranie zostaje wyświetlona zakładka przedstawiona na **rysunku 11**. W polu „Bit rate” należy wprowadzić prędkość transmisji badanego interfejsu. Może to być dowolna wartość wpisana ręcznie lub jedna z opcji widocznych na liście rozwijanej. Oprogramowanie analizatora protokołów uwzględnia kilka wersji specyfikacji LIN, którą można wybrać z listy „LIN standard”. W większości pomiarów można skorzystać z automatycznego rozpoznawania standardu (opcja „Auto”). Pozostaje jeszcze wybranie wersji napięciowej części elektrycznej interfejsu. Tu również można skorzystać z predefiniowanych standardów dostępnych przez listę rozwijaną, ale próg logiki jest też ustawiany ręcznie, gdyby mierzone były nietypowe rozwiązania sprzętowe. Po tych czynnościach analizator jest już gotowy do pracy.

Pomiary najczęściej polegają na sprawdzeniu czy w badanym systemie występują określone zdarzenia. Tak jak w poprzednich przykładach, i teraz najlepiej jest korzystać z odpowiednio dobranych zdarzeń wyzwalających. Pozwolą one jednoznacznie i stabilnie uchwycić interesujący nas stan interfejsu. Może też okazać się, że po ustawieniu układu wyzwalania w tryb „Normal” oscyloskop nie będzie wyzwalany, a jest to też informacja, mówiąca o tym, że określone zdarzenie nie występuje w badanym układzie. Powinno nas to szczególnie zadowoić, gdy poszukiwane będą błędy.

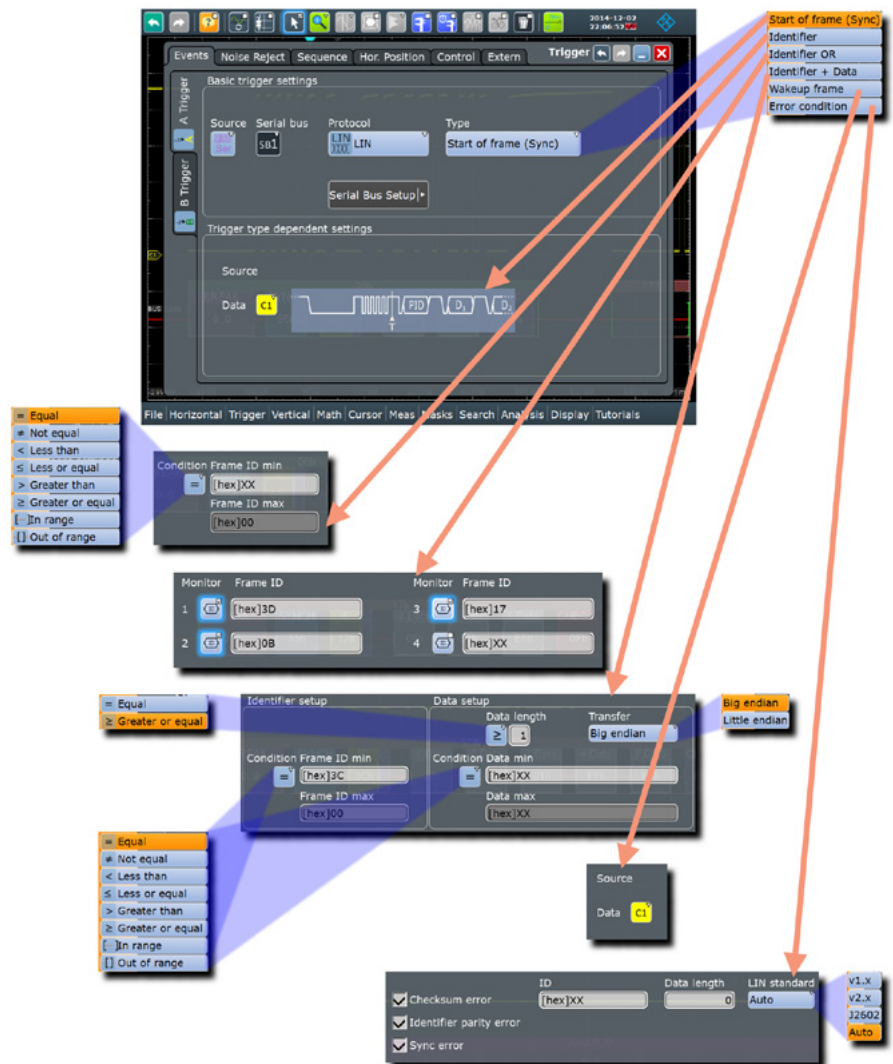
Opcje wyzwalania są dostępne po naciśnięciu przycisku *TRIGGER* (**rysunek 12**). Teraz trzeba się zdecydować, która z opcji



**Rysunek 11. Zakładka konfiguracyjna dla protokołu LIN**

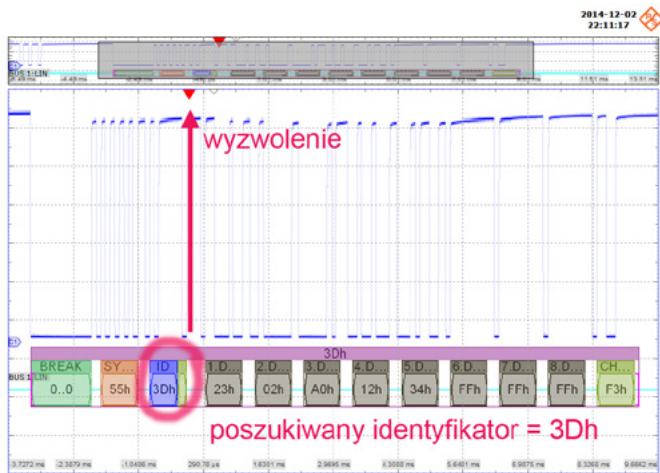
listy rozwijanej „Type” będzie najlepsza w konkretnym przypadku. Pamiętajmy o ustawieniu trybu wyzwalania „Normal”, który w większości przypadków będzie bardziej odpowiedni niż „Auto”. Po wybraniu zdarzenia wyzwalającego „Start of frame (Sync)” prawdopodobnie obraz nie będzie w pełni stabilny, gdyż będą na nim wyświetlane początki coraz to innych ramek. Opcja ta jest przydatna, gdy na magistrali LIN nie ma zbyt dużego ruchu.

Znacznie efektywniejsze będzie wyzwalanie na ramce o określonym identyfikatorze („Identifier”). Nie musi to być zresztą konkretna wartość, można zastosować również którąś z relacji: różny, mniejszy, większy, z zakresu, spoza zakresu. Po zdekodowaniu komunikatu identyfikacyjnego spełniającego wybrany warunek oscylogram zostaje stabilnie wyświetlony na ekranie (**rysunek 13**). Jeśli w rekordzie wyświetlania oscyloskopu znajdzie się kilka ramek o tym samym

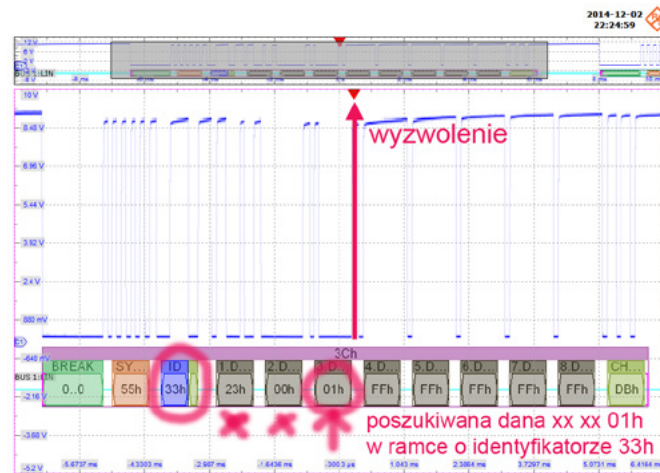


**Rysunek 12. Zakładka wyboru zdarzeń wyzwalających dla protokołu LIN**





Rysunek 13. Wyzwolenie po zlokalizowaniu ramki o identyfikatorze 3Dh



Rysunek 14. Wyzwolenie po zlokalizowaniu ramki o identyfikatorze 33h zawierającej daną 01h na 3 bajcie pola danych

identyfikatorze przyjętym do wyzwolenia, wyzwolenie nastąpi na pierwszej takiej ramce. Opcja „Identifier OR” pozwala natomiast monitorować do czterech ramek o zdefiniowanych przez użytkownika identyfikatorach. Wyzwolenie nastąpi na pierwszej ramce spełniającej jeden z tych warunków, więc cyfry wyświetlane przy opcjach na zakładce wyzwalań (rys. 12) nie mają znaczenia praktycznego. Każdą z czterech pozycji można za to zaznaczać lub odznaczać, przez co wybierane są tylko te identyfikatory, które mają być monitorowane.

Kolejne zdarzenie wyzwalające to „Identifier + Data”. Ten tryb wyzwalań poznaliśmy już podczas badania protokołu CAN. Wyzwolenie następuje po wykryciu ramki o podanym identyfikatorze (lub zakresie identyfikatorów) oraz zdefiniowanej danej występującej na konkretnej pozycji pola danych. Przy wprowadzaniu wzorców identyfikatorów i danych można korzystać ze znaku „X” zastępującego dowolną cyfrę heksadecymalną na danej pozycji, więc identyfikator „3X” może oznaczać np. „3D”, „3C” itp. Jeśli bajty z pola danych będą traktowane

jako liczba, to opcją transfer („Big endian” lub „Little endian”) można określać czy wzorzec jest wprowadzany od mniej znaczącej danej czy od bardziej znaczącej. Na przykład zaznaczenie opcji „Little endian” i wprowadzenie wzorca dla poszukiwanych danych „53 B3” oznacza *de facto* poszukiwania ciągu kolejno nadawanych danych: „B3 53”, czyli bajt B3 pojawi się na magistrali przed 53. Na **rysunku 14** przedstawiono efekt wyzwolenia po wprowadzeniu warunku: ID = 33h, dana = XX XX 01 (Big endian).

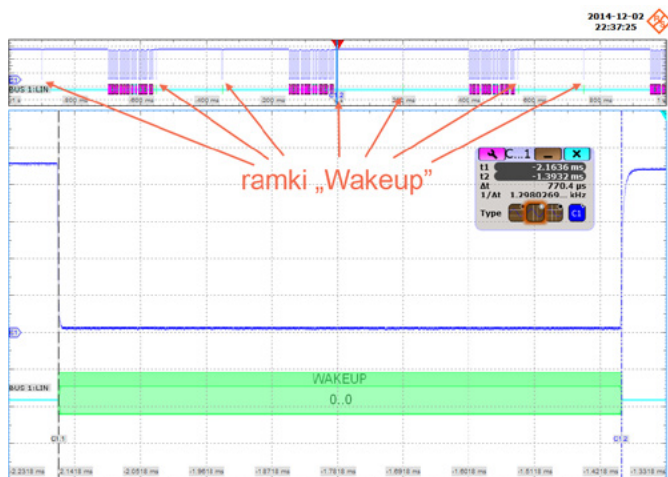
Pozostały do rozpatrzenia jeszcze dwa zdarzenia, które mogą być wykorzystane do wyzwolenia oscyloskopu. Pierwszym z nich jest ramka budzenia urządzeń dołączonych do magistrali LIN („Wakeup”). Jak wiemy, jest to impuls ze stanem dominującym o czasie trwania spełniającym podany wcześniej warunek. Na **rysunku 15** widoczne są takie ramki nadawane pomiędzy paczkami transmisji. W dolnej części ekranu ramkę tę powiększono funkcją *Zoom*.

Ostatnia opcja wyzwalań pozwala lokalizować błędy występujące na magistrali LIN. Są to: błąd sumy kontrolnej, błąd

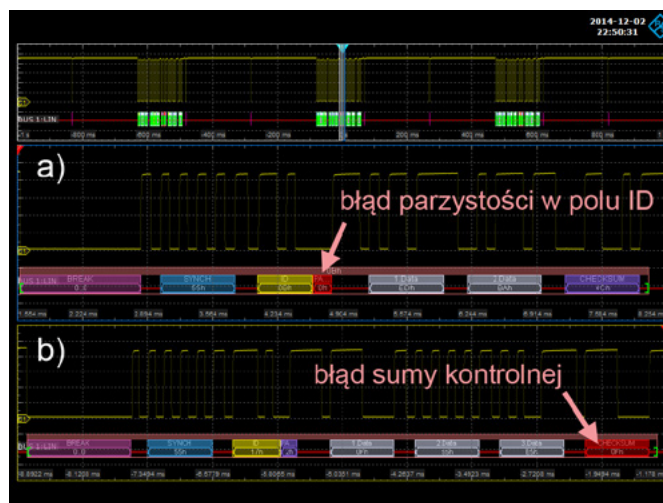
parzystości w polu identyfikatora i błąd synchronizacji. Przykład sygnalizacji takich błędów przedstawiono na **rysunku 16**. Na wykresie analizatora błędne pola są zaznaczone kolorem czerwonym. Należy zauważyć, że mimo braku arbitrażu w interfejsie LIN, zastosowane w nim mechanizmy kontroli poprawności transmisji są bardzo skuteczne.

Przedstawione w trzech częściach artykułu pomiary najbardziej popularnych szeregowych interfejsów komunikacyjnych – UART/RS232, SPI, I<sup>2</sup>C, CAN i LIN wykonywane za pomocą oscyloskopów firmy Rohde&Schwarz potwierdziły wysoką ich przydatność do podobnych zadań. Algorytmy pomiarowe działają bezbłędnie, na uwagę zasługuje intuicyjna obsługa analizatora protokołów umożliwiająca jego obsługę nawet użytkownikom o małym doświadczeniu. W artykule opisano kilka przykładowych protokołów analizowanych oscyloskopami R&S. Możliwości tych przyrządów są jednak znacznie szersze. Wykupując odpowiednie opcje można badać również takie protokoły jak: FlexRay, MIL-1553, ARINC 429.

Jarosław Doliński, EP



Rysunek 15. Ramka „Wakeup”



Rysunek 16. Wyzwolenie po zlokalizowaniu ramek, w których wykryto błędy a) parzystości w polu ID b) sumy kontrolnej