

Rysunek 1. Podstawowe okno robocze oscyloskopu RTE1034

Analizowanie protokołów szeregowych oscyloskopami Rohde&Schwarz (1) – UART/RS232

Badanie protokołów komunikacyjnych jest obowiązkową funkcją oscyloskopów cyfrowych klasy co najmniej średniej. Jednak najczęściej nadal jest ona udostępniana 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.

Możliwość badania komunikacyjnych interfejsów szeregowych mają oscyloskopy wszystkich rodzin znajdujących się w ofercie firmy Rohde&Schwarz, a więc: RTO, RTE i RTM. Analiza protokołów jest też dostępna w oscyloskopach HMO znanych wcześniej jako wyroby Hamega, obecnie sprzedawanych z logo Rohde&Schwarz.

W cyklu artykułów o analizie protokołów jako model odniesienia przyjęto oscyloskop RTE1034, ale oprogramowanie firmowe

wszystkich rodzin jest na tyle do siebie podobne, że przedstawione tu informacje będą przydatne dla wszystkich oscyloskopów RTO, RTE i RTM. Analiza poszczególnych protokołów może być prowadzona pod warunkiem zainstalowania opcji dla innych rodzin – **tab. 1** udostępniających odpowiednie funkcje wyzwalania i dekodowania. Wyzwalanie protokołów I²C, SPI, UART/RS-232/RS-422/RS-485 jest objęte standardem w oscyloskopach RTO. Pomiar interfejsów

szeregowych mogą być wykonywane z zastosowaniem standardowych sond analogowych. Obowiązują wówczas ograniczenia wynikające z takich parametrów technicznych oscyloskopu jak: pasmo, szybkość próbkowania, długość rekordu akwizycji itp. Możliwości te są rozszerzane po zainstalowaniu opcji MSO obsługującej dodatkowo kanały cyfrowe. Mogą być one wykorzystywane również do analizy protokołów. Zestawienie możliwości poszczególnych rodzin przedstawiono w **tab. 1**. W **tab. 2** zestawiono natomiast parametry pomiaru sygnałów cyfrowych oscyloskopem z opcją MSO.

Jak widać z danych zamieszczonych w **tab. 1** możliwości analizy nieznacznie różnią się pomiędzy poszczególnymi rodzinami. Najszerze ma najbardziej zaawansowana technicznie rodzina RTO, mniejsze RTE i najmniejsze RTM. Należy zauważyć, że podstawowe interfejsy są obsługiwane przez wszystkie

Tabela 1. Zestawienie opcji wykorzystywanych do analizy protokołów w oscyloskopach Rohde&Schwarz z rodzin RTO, RTE i RTM

RTO		
Typ interfejsu	Opcje wymagane do wyzwania	Opcje wymagane do dekodowania
I ² C/ SPI	Standard	R&S RTO-K1
UART/RS-232/RS-422/RS-485	Standard	R&S RTO-K2
CAN/ LIN	R&S RTO-K3	R&S RTO-K3
FlexRay	R&S RTO-K4	R&S RTO-K4
I ² S/LJ/RJ/TDM	R&S RTO-K5	R&S RTO-K5
MIL-STD-1553	R&S RTO-K6	R&S RTO-K6
ARINC429	R&S RTO-K7	R&S RTO-K7
CAN FD	R&S RTO-K9	R&S RTO-K9
MIPI RFFE	R&S RTO-K40	R&S RTO-K40
MDIO	R&S RTO-K55	R&S RTO-K55
USB1.0/1.1/2.0/HSIC	R&S RTO-K60	R&S RTO-K60
RTE		
Typ interfejsu	Opcje dla wyzwania i dekodowania	
I ² C/SPI	R&S RTE-K1	
UART/RS-232/RS-422/RS-485	R&S RTE-K2	
CAN/LIN	R&S RTE-K3	
FlexRay	R&S RTE-K4	
I ² S/LJ/RJ/TDM	R&S RTE-K5	
MIL-STD-1553	R&S RTE-K6	
ARINC 429	R&S RTE-K7	
MDIO	R&S RTE-K55	
RTM		
Typ interfejsu	Opcje dla wyzwania i dekodowania	
I ² C/SPI	R&S RTM-K1	
UART/RS-232/RS-422/RS-485	R&S RTM-K2	
CAN/LIN	R&S RTM-K3	
Audio (I ² S/LJ/RJ/TDM)	R&S RTM-K5	
MIL-STD-1553	R&S RTM-K6	
ARINC 429	R&S RTM-K7	

oscyloskopy R&S, a dodatkowo w rodzinie RTO zawarto w standardzie funkcje wyzwania dla interfejsów SPI, I²C oraz UART/RS232/RS422/RS485.

Filozofia obsługi przyrządów

Rohde&Schwarz to firma z ponad 80-letnią tradycją, znana przede wszystkim z produkcji aparatury wykorzystywanej w pomiarach radiowych i telekomunikacyjnych (analogizatory widma, analizatory sygnałowe, wektorowe generatory sygnałów itp.). Do tej kolekcji stosunkowo niedawno dołączyły oscyloskopy cyfrowe. We wszystkich najnowszych wyrobach stosowane są najnowocześniejsze dostępne technologie i podzespoły. Wprowadzane jest też w miarę ujednolicone oprogramowanie firmowe, które

Tabela 2. Parametry pomiaru sygnałów cyfrowych

Liczba kanałów cyfrowych	16
Maksymalna częstotliwość sygnału cyfrowego	400 MHz
Minimalna amplituda sygnału cyfrowego	500 mV _{pp}
Szybkość próbkowania	5 GSa/s
Długość rekordu akwizycji	100 Mpróbek dla każdego kanału
Szybkość akwizycji	>200000 przebiegów/s
Bogaty zestaw opcji wyzwania	

pozwała łatwo obsługiwać przyrządy niezależnie od przeznaczenia i zasady działania. Jest to równocześnie dobra metoda przyciągnięcia i przywiązania klientów do firmy.

Coraz częściej w aparaturze pomiarowej wykorzystywane są duże wyświetlacze dotykowe, które znakomicie ułatwiają obsługę przyrządów (wyświetlacz taki zastosowano w oscyloskopach RTO i RTE). Kolejnym udogodnieniem jest myszka i klawiatura USB, bardzo dobrze znane chyba wszystkim użytkownikom przyrządów pomiarowych, bo chyba nie ma już takich osób, które na co dzień nie korzystałyby z komputerów. W wyższych modelach przyrządów zastosowano system operacyjny Windows, stwarzający dodatkowe możliwości na przykład w zakresie współpracy urządzeń, sterowania i wymiany danych. Dodatkową korzyścią jest względnie łatwe panowanie nad formą przedstawiania informacji wyświetlanych na ekranie. Funkcje oscyloskopu są uruchamiane kilkoma metodami: przyciskami znajdującymi się na pulpicie przyrządu, poleceniami zawartymi w menu wy-

świetlanym w dolnej części ekranu lub przez naciśnięcie odpowiedniej ikonki wyświetlanej w górnej części ekranu (**rysunek 1**).

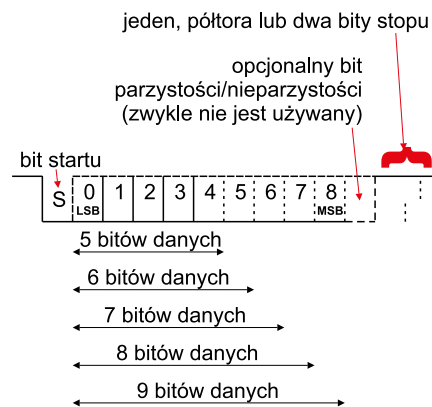
Analizowanie protokołu UART/RS232

Przesyłane interfejsem UART/RS232 informacje formowane są z ramek składających się z:

- bitu startu,
- 5, 6, 7 lub 8 bitów danych,
- bitu parzystości (opcjonalnego, zwykle pomijanego),
- jednego, półtora lub dwóch bitów stopu (**rysunek 2**).

Najczęściej stosowana jest ramka 8,n,1 (8 bitów danych, bez bitu parzystości, 1 bit stopu). Zwykle też jako pierwszy jest przesyłany zerowy, najmłodszy bit danej (LSB), ale dopuszczalna jest też kolejność odwrotna, tzn. od MSB do LSB. Interfejs UART/RS232 wykorzystuje transmisję asynchroniczną, co oznacza, że nie jest stosowana linia zegarowa. Można uznać, że elementami synchronizującymi są znaki startu, a dokładność rozpoznawania pozostałych elementów ramki zależy od dokładności lokalnych zegarów po stronie nadawczej i odbiorczej. Choć teoretycznie częstotliwość takiego zegara może być dowolna, to w praktyce są stosowane znormalizowane wartości.

Analizę wszystkich protokołów prowadzoną przy użyciu oscyloskopów R&S należy



Rysunek 2. Ramka protokołu UART/RS232

rozpocząć od wybrania odpowiedniego interfejsu i zdefiniowania wszystkich jego parametrów. Czynności te wykonuje się na przykład po naciśnięciu przycisku PROTOKOL znajdującego się na płycie czołowej oscyloskopu. Na ekranie zostaje wówczas wyświetlona zakładka zawierająca wszystkie parametry badanego interfejsu, która dla UART-a/RS232 będzie wyglądała, jak na rys. 1. Z łatwością rozpoznawane są na niej poszczególne pola i listy rozwijane definiujące parametry interfejsu, a więc: typ interfejsu, prędkość transmisji, liczbę bitów stopu, długość ramki, opcje parzystości, kolejność transmitowania bitów. Okno konfiguracji jest bardzo czytelne, w czym pomagają zdefiniowane kolorami bloki wyboru parametrów oraz graficzna interpretacja analizowanego protokołu.

W wielu analizatorach protokołów uwzględniane są tylko opcje parzystości: None, Odd i Even, czyli bez parzystości, nieparzystość i parzystość. W oscyloskopach R&S w analizie protokołów można ponadto wybierać opcje Mark, Space i Don't Care. Mark i Space to obecnie bardzo rzadko stosowane metody kontroli poprawności transmisji. Przypomnijmy, że jeśli wybrano kontrolę parzystości (Even), to bit parzystości w prawidłowo odebranym znaku powinien mieć wartość „0”, jeśli wystąpiła parzysta liczba jedynek w znaku (**rysunek 3**). Analogicznie jest po włączeniu kontroli nieparzystości (Odd). Bit kontrolny powinien być równy „0” tylko wtedy, gdy w przesyłanym znaku jest nieparzysta liczba jedynek. Jeśli więc włączono kontrolę nieparzystości (Odd) i w bajcie o nieparzystej liczbie jedynek bit kontrolny jest równy „1”, to mamy przypadek wykrycia błędu transmisji. Załóżmy, że nadajnik wysyła bajt z parzystą liczbą jedynek. Przy włączonej kontroli Even ustawia dla niego bit parzystości na „0”. Jeśli odczytany przez odbiornik bit kontrolny będzie miał wartość „1”, to będzie oznaczało, że któryś z bitów w tym bajcie został przekłamany (rys. 3). Niestety, w takiej metodzie kontroli nie ma możliwości identyfikacji tego bitu. Jak na ironię przekłamany może być sam bit kontrolny.

Bitu parzystości na oscylogramie przypisanym do danego interfejsu są wyodrębnione od całego znaku. Ich nieprawidłowa wartość jest sygnalizowana kolorem czerwonym, prawidłowa

natomiast kolorem pomarańczowym. Błędny znak jest wyświetlany kolorem fioletowym w odróżnieniu od żółtego znaku interpretowanego jako bezbłędny. Symbolami „<” i „>” opisywane są bity startu i stopu.

Opcja Mark stanowi dość specyficzną i niestety bardzo nieskuteczną kontrolę transmisji. Bit kontrolny powinien być w tym przypadku zawsze transmitowany i niezależnie od liczby jedynek w znaku zawsze ma wartość „1”. Analogicznie jest z opcją Space, tylko teraz bit kontrolny ma zawsze wartość „0”. Wybierając opcję Don't Care analizator pomija badanie bitu parzystości, który oczywiście powinien być nadawany zgodnie z którąś z powyższych metod.

Warto jeszcze zwrócić uwagę na kilka dodatkowych parametrów zawartych w opcjach analizy protokołu UART/RS232. Są to:

„Polarity”, określający poziom logiczny występujący na liniach TxD i RxD w stanie Idle, a więc wtedy, gdy nie są przesyłane żadne dane. W większości rozwiązań praktycznych jest to poziom wysoki.

„Packets”, określający czasami stosowaną metodę organizowania danych w większe bloki (pakiety). W rozwiązaniu takim przyjmowana jest zasada, że każdy blok danych zawierający powiązane ze sobą dane, np. wyniki pomiarów wykonanych w jednej sesji, jest kończony umownym znakiem. Z oczywistych powodów, jako znak końca bloku powinna być wybierana dana niewystępująca w samym bloku. Znak końca bloku jest ustalany w opcji „End word”, po wybraniu której wprowadza się stosowny wzorec. Znakiem końca pakietu może też być określona przerwa czasowa pomiędzy kolejnymi blokami. Zatem po wybraniu opcji „Timeout” należy wprowadzić długość tej przerwy w odpowiednich jednostkach czasu. Przykład transmisji, w której jest przesyłany blok danych kończony znakiem „*” (0x2A) przedstawiono na **rysunku 4**. Dodatkowo widoczne są tu przerwy między blokami, które mogą być również interpretowane jako koniec bloku.

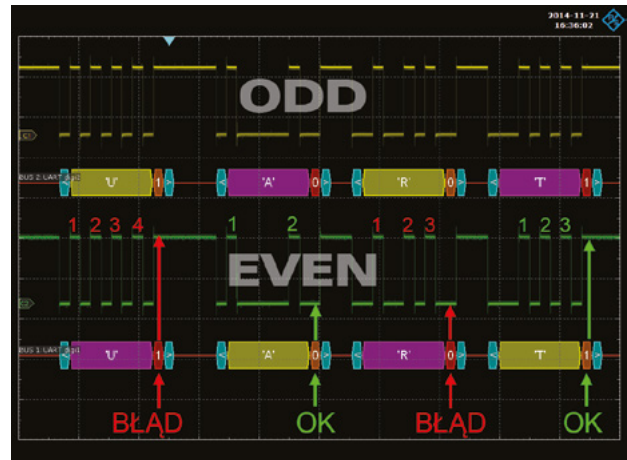
Analizator protokołów zamieszcza interpretację danych na wykresie BUS, który wygodnie jest umieścić bezpośrednio pod przebiegiem, np. jak na rys. 4. Jeżeli zdefiniowano transmisję blokową, to dodatkowo w zwartej postaci jest wyświetlany cały odczytany komunikat przesyłany w bloku. Możliwe są różne formaty prezentacji danych (np. hex, dec, octal, ASCII itp.). Przy dużej ilości danych, tak prezentowane wyniki stają się mało czytelne, dużo wygodniejsza jest forma tabelaryczna (**rysunek 5**). W tabeli z łatwością można odnaleźć interesujące miejsce transmisji. Po kliknięciu na odpowiadający mu numer widoczny w kolumnie „Word”, ponownie zostaje wyświetlony wykres, przy czym jego centralny punkt będzie odpowiadał wybranej w tabeli chwili czasowej.

Wyzwalanie zdarzeniami w protokole UART/RS232

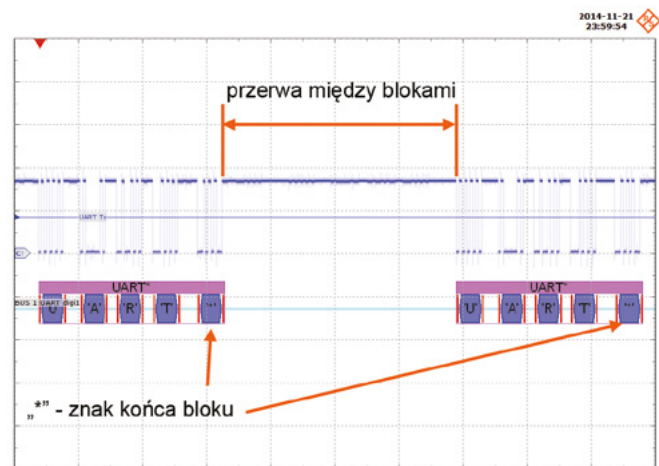
Podczas analizy danych transmitowanych badanym interfejsem bardzo przydatne są opcje wyzwalania zdarzeniami charakterystycznymi dla protokołu. Niezależnie od nich mogą być wykorzystywane i inne opcje dostępne w oscyloskopie, w tym standardowe.

Dobór zdarzenia wyzwalającego w wielu przypadkach decyduje o powodzeniu całego pomiaru. Po naciśnięciu przycisku TRIGGER na ekranie zostaje wyświetlona zakładka, na której dokonuje się konfiguracji układu wyzwalania (**rysunek 6**). Pierwszą czynnością jest wybranie źródła wyzwalania, do czego służy lista rozwijana „Source”. Ponieważ będzie badany protokół szeregowy, należy wybrać opcję „Serial bus”. Następnie ustalany jest typ badanego protokołu z listy „Protocol” – opcja „UART/RS232”. Pozostało jeszcze najważniejsze – wybranie samego zdarzenia. Dla protokołu UART/RS232 jest to: bit startu, początek bloku danych, dana o zdefiniowanej wartości, błąd parzystości, przerwa w nadawaniu (stan przeciwny do Idle), bit stopu. Niektóre z tych opcji wymagają określenia dodatkowych parametrów. Pojawiają się one po zatwierdzeniu wybranego zdarzenia.

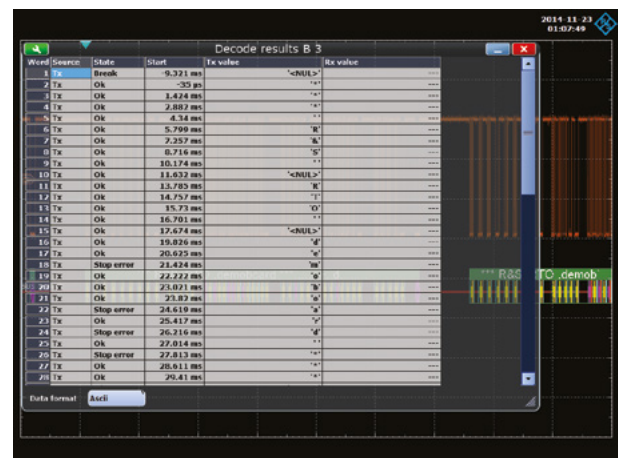
Wyzwalanie bitem startu, określoną daną, błędem parzystości i błędem znaku stopu jest dość oczywiste, podstawą czasu jest wyzwalana po stwierdzeniu wybranego zdarzenia. Opcja „Data” jest jednak szczególnie przydatna podczas badania transmisji blokowej. W oscyloskopach R&S zaimplementowano wyzwalanie po wykryciu określonej danej w bloku, nawet wtedy, gdy powtarza się ona w bloku wielokrotnie. Na **rysunku 7** przedstawiono zakładkę, na której widoczne są opcje wyświetlane po wybraniu



Rysunek 3. Rozpoznawanie błędów parzystości w protokole UART/RS232



Rysunek 4. Transmisja blokowa, w której końcem bloku jest znak „*” lub przerwa czasowa

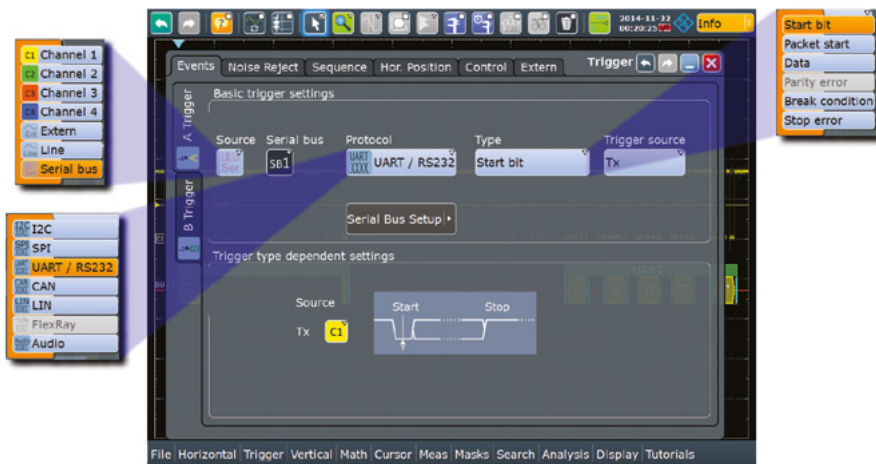


Rysunek 5. Tabelaryczna postać wyświetlania danych zdekodowanych przez analizator protokołów

wyzwalania typu „Data”. Wcześniej w konfiguracji protokołu w polu „Packets” zaznaczono opcję inną niż „None” („End word” lub „Timeout”). W opisywanym przykładzie wyzwolenie powinno nastąpić po rozpoznaniu w bloku znaku „%” występującego na pozycji o indeksie 7. Przyjęto, że pierwsza dana ma indeks 0, druga 1 itd. Indeks 7 oznacza więc ósmy znak w bloku. Jeśli będzie on równy „%”, nastąpi wyzwolenie. Sytuację taką przedstawiono na **rysunku 8**.

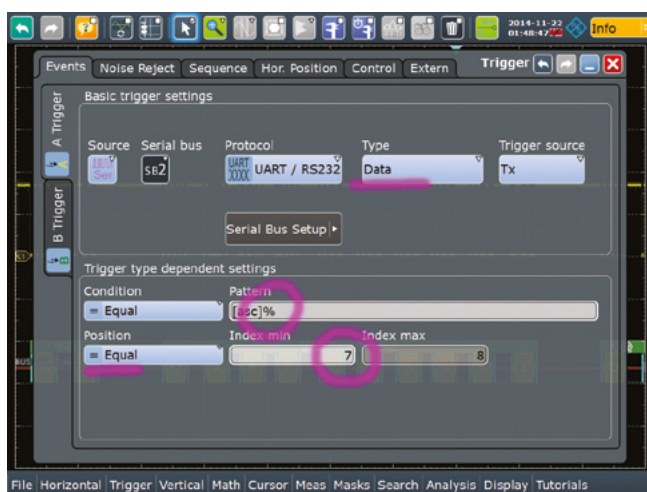
Błędy dekodowania

Analizator protokołów dekoduje dane na podstawie próbek zapisanych w rekordzie akwizycji. Dekodowanie rozpoczyna od poszukiwania

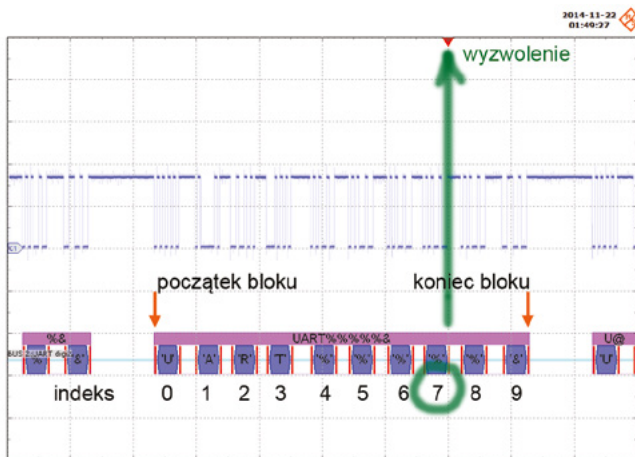


Rysunek 6. Zakładka konfiguracji układu wyzwalania zdarzeniami występującymi w protokole UART/RS232

pierwszego bitu „0”, który będzie traktowany jako bit startu (w typowej ramce). Problem polega na tym, że bity startu nie różnią się niczym od bitów danych. Co się więc zdarzy, gdy na początku rekordu akwizycji znajdzie się połowa danej z interfejsu? Analizator zinterpretuje pierwszy napotkany bit „0” jako bit startu, gdy tymczasem może to być jakiś wewnętrzny bit danej. Dalsza interpretacja będzie oczywiście błędna, przynajmniej do wyraźnego zakończenia ramki i odnalezienia następnej. Taki przykład przedstawiono na rysunku 9. Rozpatrywany jest tu blok danych składający się m.in. z kolejnych znaków ASCII: ABCD#s. Na rysunku 9a. nie ma żadnych wątpliwości z interpretacją. Teraz przesuwamy oscylogram pokrętkiem POSITION w lewo (w przykładzie o 330 μs), tak żeby początek znalazł się tuż przy lewej krawędzi ekranu (rysunek 9b). Nadal wszystko jest



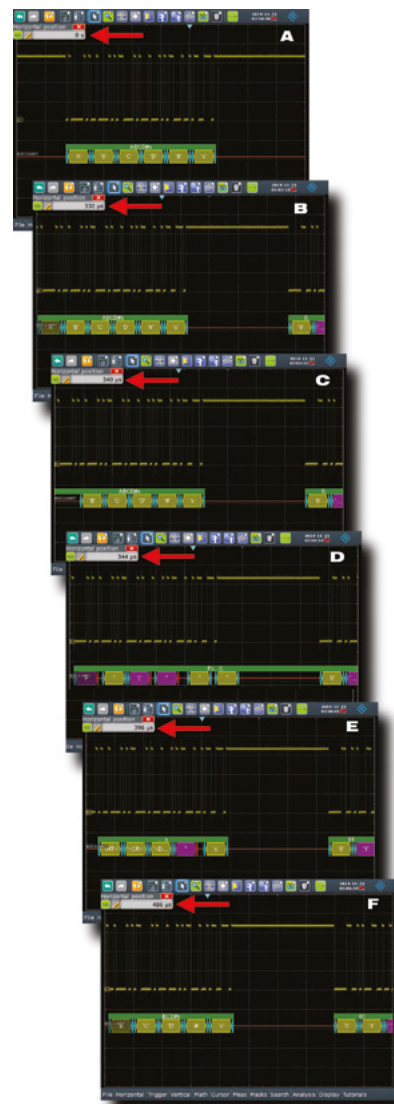
Rysunek 7. Zakładka konfiguracji układu wyzwalania od danej równej „%”, występującej w bloku na pozycji o indeksie 7



Rysunek 8. Oscylogram po wyzwoleniu zdarzeniem zdefiniowanym na zakładce z rys. 6

w porządku, ale dalsze przesunięcie o 10 μs (rysunek 9c) spowodowało, że pierwsza dana znalazła się poza ekranem, wypadła też z rekordu akwizycji. Tu złożyło się szczęśliwie i analizator tylko wyciął tę daną, wyświetlając zdekodowany blok jako: BCD#s. Przy przesunięciu oscylogramu o kolejne 4 μs (rysunek 9d)

szczęście jednak się skończyło. Tym razem analizator zupełnie się pogubił. Pojawiły się błędy ramek, niektóre dane są oznaczone jako błędne, a co gorsze dane wyświetlone jako bezbłędne zupełnie nie odpowiadają temu, co jest przesyłane interfejsem. Sytuacja powtarza się przy dalszym przesuwaniu, aż w pewnym momencie znowu trafiamy na prawidłową lokalizację



Rysunek 9. Błędy dekodowania danych przez analizator protokołów, na które trzeba zwracać uwagę

bitu startu i analizator poprawnie dekoduje dane (rysunek 9f).

Podobne kłopoty mogą występować na końcu rekordu, o czym użytkownik musi zawsze pamiętać. Jeśli jakaś ramka wychodzi poza rekord, najczęściej jest zaznaczana na oscylogramie jako błędna, choćby dlatego, że dla grupy bitów nie znaleziono bitu stopu. Sygnalizacja błędów jest w tym przypadku informacją dla użytkownika, że powinien wykonać jakiś zabieg zapewniający umieszczenie w rekordzie akwizycji również tego fragmentu transmisji, który został oznaczony jako błędny. Rozwiązaniem może być zmiana podstawy czasu, albo odpowiednie przesunięcie punktu wyzwoleń (opóźnienie).

Przykład ten powinien wyczulić użytkowników na pewne nietypowe sytuacje występujące podczas wykorzystywania oscyloskopów do analizy protokołów. Jak widać, nigdy nie można bezkrytycznie podchodzić do wyników, zawsze należy zachować czujność. Są to zresztą efekty występujące w każdym oscyloskopie, niezależnie od producenta.

Jarosław Doliński