

# Analiza protokołów (4)

## Analizowanie interfejsu I<sup>2</sup>C

**Termin „analiza protokołów” znany jest elektronikom nie od dziś. Specjaliści różnych dziedzin interpretują go jednak zgoła odmiennie. Inżynier telekomunikacji zajmujący się systemami łączności radiowej czy telefonii komórkowej analizę protokołów będzie rozumiał inaczej niż informatyk, a jeszcze inaczej konstruktor projektujący układy elektroniczne. W artykule zajmiemy się protokołami wykorzystywanymi w popularnych interfejsach komunikacyjnych.**

W kolejnej części cyklu omówiono protokół stosowany w interfejsie I<sup>2</sup>C. Do przesyłania danych wykorzystywane są tylko dwie linie, zatem dwukanałowy oscyloskop Rigol DS2202 pracujący jako analizator protokołów będzie odpowiednim przyrządem.

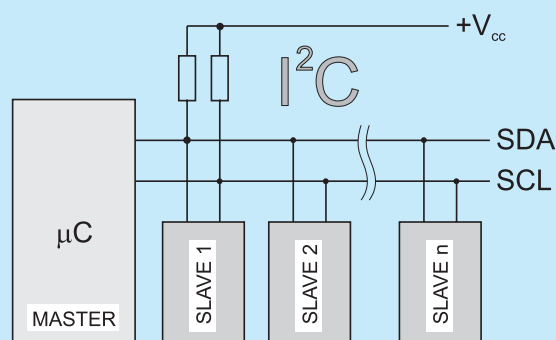
W wyniku burzliwego rozwoju systemów mikroprocesorowych, jaki miał miejsce na początku lat 80. pojawiła się konieczność opracowania wydajnego interfejsu, który byłby wykorzystywany do komunikacji między mikroprocesorem a urządzeniami peryferyjnymi rozumianymi głównie jako specjalizowane układy scalone lub moduły funkcjonalne montowane najczęściej na jednej płytce z procesorem. Stąd m.in. wywodzi się nazwa interfejsu. I<sup>2</sup>C to akronim od *Inter-Integrated-Circuit*. Zastosowane w interfejsie rozwiązania sprzętowe oraz przyjęty protokół przesyłania danych zapewniają dwukierunkową transmisję w systemie zawierającym wiele nadajników/odbiorników. Założenia techniczne powstały w firmie Philips, która widząc dużą przydatność praktyczną swojego opracowania zastrzegła do niego prawa autorskie. Nawiasem mówiąc jest to często stosowany zabieg pozwalający zwiększać zyski. W rezultacie pomysły zostały skopiowane w trudnych dziś do zliczenia odmianach tego interfejsu, pozwalających omijać zawile uwarunkowania prawne.

### Hardware I<sup>2</sup>C

Tematem tej części artykułu jest wprawdzie analiza protokołu I<sup>2</sup>C, ale trudno ją będzie przeprowadzić bez choćby podstawowej wiedzy o sprzętowych rozwiązaniach interfejsu.

Jak wiemy, w jednym systemie mikroprocesorowym można stosować wiele urządzeń komunikujących się *via* interfejs I<sup>2</sup>C. Do przesyłania danych wykorzystywana jest wyodrębniona magistrala składająca się z linii SDA, którą są przesyłane adresy i dane oraz linii SCL, którą przesyłany jest przebieg zegarowy taktujący transmisję. Procesor pełni rolę urządzenia nadrzędnego (Master), a pozostałe peryferia są urządzeniami podrzędnymi (Slave) – **rysunek 31**. Możliwa jest również topologia zawierająca kilka mikroprocesorów, ale ich funkcje w systemie muszą być rozdzielone – zawsze Masterem może być tylko jeden procesor.

Do linii interfejsu dołączane są urządzenia o bliżej nieokreślonym położeniu względem siebie. W ogólnym



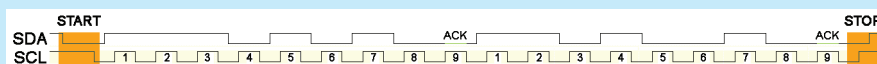
**Rysunek 31.** Typowy układ połączeń komunikujących się za pośrednictwem interfejsu I<sup>2</sup>C

przypadku nie jest też znana ich liczba. W takiej rozproszonej topologii może dochodzić do znacznego zniekształcania sygnałów cyfrowych, uniemożliwiającego prawidłową interpretację danych. Zawsze więc należy stosować odpowiednio dobrane rezystory terminujące. Wszystkie wyjścia dołączone do interfejsu I<sup>2</sup>C są typu *Open Drain*. Urządzenie nieaktywne wyłącza wyjście – nazywamy to zwolnieniem linii. Jednym z zadań rezystorów terminujących jest również podciąganie linii do stanu wysokiego.

Każda wymiana informacji między urządzeniami Master i Slave jest inicjowana przez Mastera. Master wysyła adres urządzenia Slave, do którego chce wpisać, lub z którego chce odczytać informację. Oczywiście każde urządzenie dołączone do danego interfejsu musi mieć unikatowy adres własny. Adres ten jest najczęściej na sztywno zapisywany w danym typie układu scalonego na etapie jego produkcji. Aby jednak nie dochodziło do konfliktów w przypadku stosowania kilku układów tego samego typu w jednym systemie, przewidziano możliwość modyfikacji kilku bitów adresowych przez użytkownika. Całkowity adres składa się więc z niezmienniej części identyfikującej typ układu i z modyfikowalnej części identyfikującej np. konkretny układ scalony wchodzący w skład systemu mikroprocesorowego. Specyfikacja I<sup>2</sup>C przewiduje stosowanie adresów 7- lub 10-bitowych

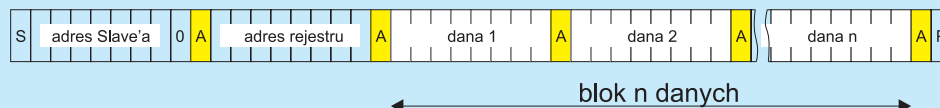
### Protokół I<sup>2</sup>C

W stanie bezczynności linie SCL i SDA są zwolnione (pozostają w stanie wysokim na skutek odłączenia wyjść).

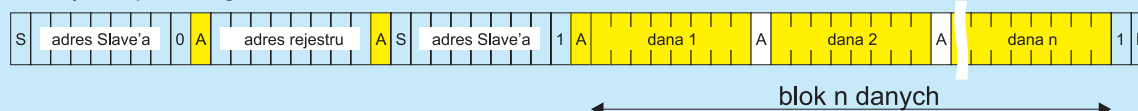


Rysunek 32. Sekwencja START i STOP oraz bity potwierżeń ACK

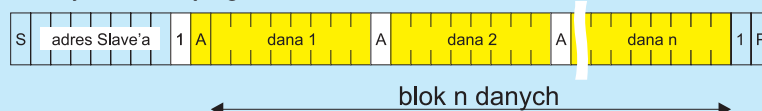
Zapis



Odczyt od podanego adresu



Odczyt od bieżącego adresu



- generuje Master
- generuje Slave
- S sekwencja Startu
- P sekwencja Stopu
- A bit potwierzenia (ACK)

Rysunek 33. Warianty protokołu I<sup>2</sup>C: a) bezpośredni zapis danych do urządzenia Slave, b) odczyt wskazanego rejestru urządzenia Slave, c) bezpośredni odczyt z urządzenia Slave

Transmisja jest inicjowana wygenerowaniem sekwencji startu „S” (rysunek 32). Odpowiada jej opadające zbocze sygnału SDA, podczas gdy linia SCL jest utrzymywana w stanie wysokim. Stany linii SDA mogą się zmieniać tylko wtedy, gdy linia SCL jest w stanie niskim. Po wygenerowaniu sekwencji startu linia SCL jest zerowana, a w chwilę później linia SDA przybiera stan zgodny z wartością pierwszego bitu transmitowanej danej. Następnie na linii SCL generowanych jest 8 impulsów taktujących kolejne bity przesyłane linią danych (SDA). W protokole I<sup>2</sup>C nie przewidziano modyfikacji długości danych. Zawsze są one 8-bitowe. W każdej ramce pojawia się jednak dziewiąty impuls zegarowy przeznaczony na przesłanie bitu potwierdzającego odebranie informacji. Jeśli operacja, np. nadawania, przebiegła prawidłowo, układ Slave wymusza zerowanie linii SDA (co oznacza pozytywne potwierdzenie — ACK). Stan ten może się przedłużać, np. z powodu dłuższej zajętości odbiornika, wynikającej choćby z obsługi przerwania. Układ Slave wprowadza Mastera w stan oczekiwania (*wait state*) przez przytrzymanie linii SCL w stanie niskim. Do momentu jej zwolnienia wszelkie operacje na magistrali I<sup>2</sup>C są wstrzymane. Brak potwierdzenia (bit ACK=1) powoduje zakończenie transmisji. Transmisję kończy wystawienie sekwencji stopu — P, czyli wygenerowanie narastającego impulsu na linii SDA, podczas gdy linia SCL pozostaje w stanie wysokim (rysunek 32).

W podstawowej specyfikacji I<sup>2</sup>C prędkość transmisji jest równa 100 kb/s. W trybie fast jest ona równa 400 kb/s.

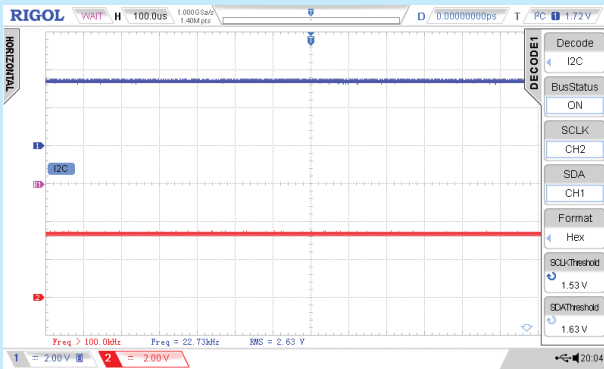
Od chwili opracowania interfejsu I<sup>2</sup>C był on kilkakrotnie modyfikowany. Aktualnie prędkość transmisji może dochodzić do 3,4 Mb/s (*High-Speed Mode*).

### Warianty pracy

W zależności od funkcji realizowanych przez dany układ peryferyjny stosowane są różne warianty komunikacji. Schematycznie przedstawiono je na rysunku 33.

W pierwszym przypadku (rysunek 33a) układ Master zapisuje informację do Slave. Po wygenerowaniu sekwencji startu (S) Master przesyła linią SDA adres układu Slave, do którego kieruje dane (transmisja przebiega od najstarszego do najmłodszego bitu). Po 7 bitach adresu, ósmy bit ma wartość „0” oznaczającą, że dane będą zapisywane do Slave’a. Bajt ten odbierają wszystkie układy dołączone do magistrali, ale tylko ten, którego adres jest zgodny z nadanym odpowiada wystawiając bit potwierdzenia ACK poprzez ściągnięcie linii SDA do stanu niskiego. Następnie nadawane są kolejne bajty, które zawsze muszą być potwierdzone bitem ACK (niski stan) przez układ Slave, do którego kierowane są dane.

Na rysunku 33b przedstawiono blokowy odczyt danych ze Slave’a, począwszy od rejestru o wskazanym adresie. Transmisja jest inicjowana wystawieniem sekwencji startu (S), a następnie zaadresowaniu odpowiedniego Slave’a. Bit ósmy występujący bezpośrednio po adresie, tak jak w poprzednim przypadku ma wartość „0”. Po zidentyfikowaniu bitu potwierdzenia wygenerowanym przez Slave’a Master wysyła adres rejestru, od



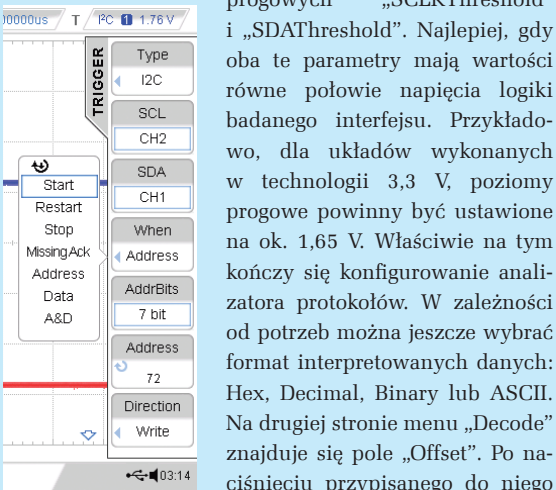
**Rysunek 34. Menu konfiguracyjne analizatora protokołu I<sup>2</sup>C**

którego będą odczytywane informacje. Ponownie jest on potwierdzany przez układ Slave bitem ACK. Master musi teraz ustawić logikę Slave'a obsługującą interfejs w tryb odczytu jego rejestrów. Jest to realizowane przez ponowne wysłanie adresu urządzenia Slave z bitem R/W (8 bit) równym 1, co oznacza odczyt. W takt kolejnych impulsów zegarowych przesyłane są bity danych z urządzenia Slave, a bit potwierdzenia ACK wystawia tym razem Master. Adresy rejestrów Slave'a są inkrementowane po przesłaniu każdej 8-bitowej danej. Transmisja kończy się brakiem potwierdzenia (9 bit równy „1”) i wygenerowaniem sekwencji stopu (P). Według podobnego schematu przebiega zapis danych do wskazanego rejestru, jednakże przy powtórzeniu adresu Slave'a bit R/W jest równy „0”, co oznacza zapis. Dalsza część komunikacji przebiega zgodnie ze schematem przedstawionym na rysunku 33a.

Na rysunku 33c przedstawiono przebieg transmisji podczas bezpośredniego odczytu urządzenia Slave.

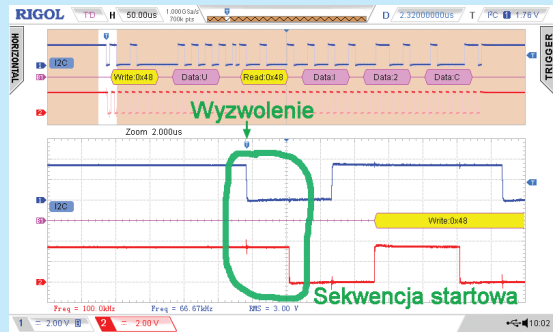
### Analiza protokołu I<sup>2</sup>C

Pracę rozpoczynamy od dołączenia kanałów pomiarowych oscyloskopu do linii interfejsu, np. kanał CH2 do linii SCL, kanał CH1 do linii SDA. Taką konfigurację należy zadeklarować w menu wyświetlanym po naciśnięciu przycisku *Decode1* (lub *Decode2*) – **rysunek 34**. Ramka protokołu I<sup>2</sup>C nie zawiera zmiennych elementów, w menu „Decode1” („Dekode2”) nie ma więc żadnych pól, od ustawienia których zależałaby interpretacja danych na liniach interfejsu. Do obowiązkowych czynności należy jedynie ustawienie prawidłowych poziomów progowych „SCLThreshold” i „SDAThreshold”.

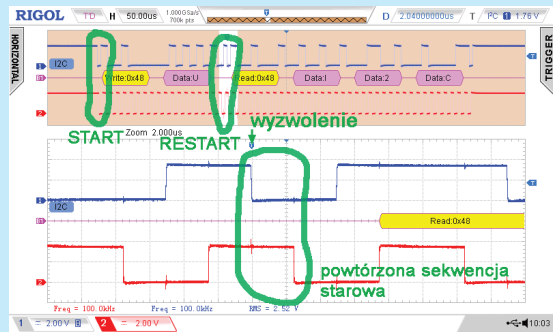


**Rysunek 35. Menu wyzwalania I<sup>2</sup>C**

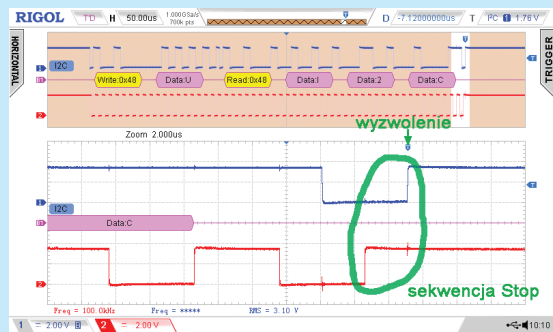
Najlepiej, gdy oba te parametry mają wartości równe połowie napięcia logiki badanego interfejsu. Przykładowo, dla układów wykonanych w technologii 3,3 V, poziomy progowe powinny być ustawione na ok. 1.65 V. Właściwie na tym kończy się konfigurowanie analizatora protokołów. W zależności od potrzeb można jeszcze wybrać format interpretowanych danych: Hex, Decimal, Binary lub ASCII. Na drugiej stronie menu „Decode” znajduje się pole „Offset”. Po naciśnięciu przypisanego do niego przycisku, za pomocą uniwersalnego pokrętki regulacyjnego usta-



**Rysunek 36. Wyzwolenie na sekwencji Start**



**Rysunek 37. Wyzwolenie na powtórzonej sekwencji startowej (Restart)**



**Rysunek 38. Wyzwolenie na sekwencji Stop**

wana jest najbardziej odpowiednia wysokość wyświetlania dekodowanych danych.

### Wyzwalanie protokołu I<sup>2</sup>C

Przebiegi występujące na liniach interfejsu I<sup>2</sup>C zostaną wyświetlone na ekranie nawet po zastosowaniu zwykłego wyzwalania zboczem. Prawdopodobnie obraz nie będzie jednak stabilny, konieczne będzie skorzystanie z jednej z opcji wyzwalania, które są dedykowane dla protokołu I<sup>2</sup>C.

Menu wyzwalania dla protokołu I<sup>2</sup>C przedstawiono na **rysunku 35**. Pierwsze trzy pola są powtórzeniem nastaw konfiguracyjnych analizatora, a więc np.: „Type”=I<sup>2</sup>C, „SCL”=CH2, „SDA”=CH1. Warto już na początku wybrać tryb wyzwalania Normal, który w większości przypadków będzie najbardziej odpowiedni. W trybie „Auto” oscylogram może być niestabilny.

W polu „When” należy wybrać jeden z kilku warunków wyzwalania. Wyboru dokonujemy w zależności od tego, co chcemy obserwować. Może to być:

- **Start.** Oscyloskop jest wyzwalany na sekwencji startowej. Jest to dobry wybór na przykład do obserwacji powtarzających się cyklicznie komunikatów. W sesjach z powtórzeniem sekwencji startowej, stoso-



Rysunek 39. Wyzwalenie na błędzie potwierdzenia **MissingAck**

wanej np. podczas odczytu rejestru o określonym adresie, przesłanym po adresie urządzenia Slave, sekwencja startowa występuje dwukrotnie – na początku i po przesłaniu adresu rejestru. Układ wyzwalania oscyloskopu DS2202 rozróżnia te sekwencje. Po wybraniu opcji *Start* wyzwalenie następuje zawsze na pierwszej sekwencji startowej (rysunek 36). Aby wyzwalac oscyloskop na drugiej sekwencji należy zastosować opcję *Restart*.

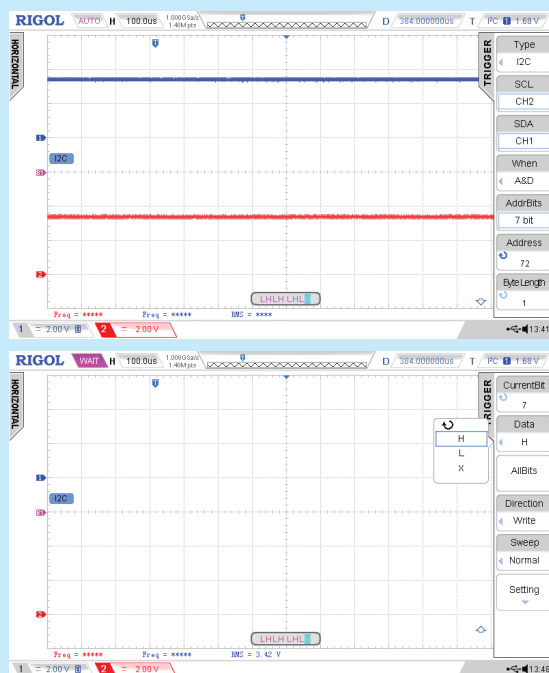
- **Restart.** Ta opcja jest wykorzystywana do wyzwalenia oscyloskopu w chwili powtórzenia sekwencji startowej. Jest przydatna w trybach zapisu lub odczytu rejestru urządzenia o podanym adresie. Oscyloskop jest wyzwalany po przesłaniu adresu rejestru i powtórzonej sekwencji startowej (restarcie), gdyż przed jej wystawieniem nie było sekwencji Stop.
- **Stop.** Wyzwalanie na sekwencji *Stop* kończącej transmisję (rysunek 38).
- **MissingAck.** Wyzwalanie po wykryciu błędnego potwierdzenia (bit ACK w ramce protokołu I<sup>2</sup>C). Na rysunku 39 przedstawiono przykładową sytuację, w której wykryto błędne potwierdzenie po przesłaniu adresu urządzenia Slave. Zgodnie z protokołem, na 9. bicie zegarowym powinien wystąpić niski stan

na linii SDA, co oznaczałoby poprawne potwierdzenie. Analizator wykrył stan wysoki, a wykryty błąd jest sygnalizowany czerwonym znacznikiem.

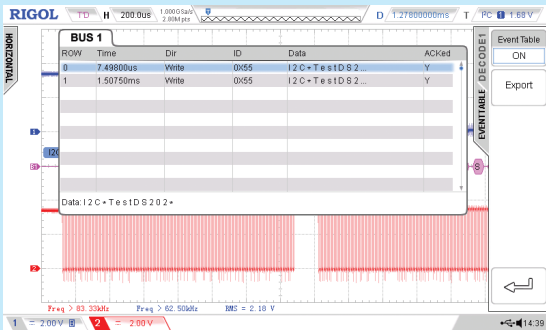
- **Address.** Wyzwalanie po wykryciu określonego adresu urządzenia Slave. Pamiętajmy, że w bajcie adresowym przesyłana jest też na jego najmłodszym bicie informacja o kierunku transmisji (R/W). Jeśli R/W=0, nastąpi zapis do układu Slave, jeśli R/W=1, nastąpi odczyt układu Slave. Sam 7-bitowy adres jest liczony od bitu nr 1 do bitu nr 7. Najlepiej zilustruje to przykład. Jeżeli cały bajt adresowy ma postać 10010000, to adres jest równy 1001000, czyli 48h i mamy do czynienia z zapisem. Jeżeli bajt adresowy ma postać 01001101, to adres jest równy 0100110, czyli 26h i mamy do czynienia z odczytem. W wyświetlonym na ekranie oscyloskopu adresie nie jest uwzględniany bit R/W (wg powyższej konwencji). Po wybraniu opcji *Address* należy określić długość adresu (7 lub 10 bitów). Adres jest niestety podawany w zapisie dziesiętnym (a więc zwykle trzeba go będzie przeliczać z postaci szesnastkowej). Kolejny parametr („Direction”), określający kierunek transmisji może przybierać wartości: *Write* (czemu odpowiada R/W=0), *Read* (czemu odpowiada R/W=1) lub *R/W* (wyzwalenie nastąpi na zadanym adresie w trybie zapisu, albo odczytu). Na rysunku 40a przedstawiono wyzwalenie po przesłaniu adresu 75h w trybie zapisu, a na rysunku 40b wyzwalenie nastąpiło na adresie 48h w trybie odczytu.
- **Data.** Wyzwalenie po wykryciu zdefiniowanej danej lub sekwencji danych. Analizowany ciąg danych może mieć długość od 1 do 5 bajtów. Definiowanie danych jest dość żmudne, gdyż wymaga indywidualnego ustawienia każdego bitu. Dla danej 5-bitowej trzeba więc wykonać aż 40 operacji. Bit może przyjmować wartości „L”, „H” lub „X”. „X” oznacza, że dany bit nie będzie sprawdzany w odebranym ciągu. Przy definiowaniu bitów przyjęto dość mylącą konwencję, w której najstarszy bit ma numer 0, najmłodszy



Rysunek 40. Wyzwalenie na adresie 75h w trybie: a) zapisu, b) odczytu



Rysunek 41. Tabelaryczna postać dekodowanych danych dla protokołu I<sup>2</sup>C



Rysunek 42.

szy zaś numer 7. Opcja ta byłaby bardzo przydatna w praktyce, niestety, w testowanym egzemplarzu oscyloskopu Rigol DS2202 nie działała.

- **A&D.** Działa za to poprawnie wyzwalanie sekwencją adres i dana. Ten tryb wyzwalania jest bardzo przydatny na przykład podczas lokalizacji początku transmisji do określonego rejestru urządzenia Slave o zadanym adresie. Na przykład, gdy chcemy wyzwolić oscyloskop na początku przesyłania danych do rejestru o numerze 55h należącego do urządzenia Slave o adresie 48h, należy wybrać opcję wyzwalania *A&D*, następnie zadeklarować długość adresu (np. 7 bitów), prowadzić ten adres (niestety tu znowu obowiązuje postać dziesiętna, adres 48h jest więc równy 72 i taką wartość należy wprowadzić w polu „Address”), teraz deklarujemy długość danej – w naszym przykładzie będzie równa 1. Pierwsza strona opcji wyzwalania powinna wyglądać tak, jak na **rysunku 41a**, ale to nie koniec. Przechodzimy na drugą stronę, na której znajdujemy znane już opcje ustawiania bitów danych, a dodatkowo należy tu również określić kierunek transmisji (*Read*, *Write*, *R/W*).

Druga strona opcji wyzwalania dla naszego przykładu powinna wyglądać jak na rysunku 41b. Przyjęto, że dana ma wartość 55h, czyli 01010101. Oscylogram dla tego przykładu przedstawiono na **rysunku 42**.

Niezależnie od trybów wyzwalania oscyloskop Rigol DS2202 pracujący jako analizator protokołu I<sup>2</sup>C, może wyświetlać zdekodowane dane jako oscylogramy lub w postaci tabelarycznej. Tabela jest włączana po naciśnięciu przycisku *Decode1* (*Decode2*) i przejściu na drugą stronę menu. Należy uaktywnić polecenie „Event Table”. Zdekodowane w ten sposób dane przedstawiono na **rysunku 43**.

W następnym odcinku omówimy ostatni, dostępny w oscyloskopie Rigol DS2202 protokół. Będzie to analiza magistrali równoległej.

Jarosław Doliński, EP

# Serwisy www

dla branży elektroniki i automatyki



**ElektronikaB2B**  
Portal branżowy dla elektroników



**AutomatykaB2B**  
Portal branżowy dla automatyków

**ELEKTRONIKA  
PRAKTYCZNA**



ponad  
**500 000**  
odsoń miesięcznie

ponad  
**140 000**  
użytkowników miesięcznie



ponad  
**11 000**  
subskrybentów codziennego newslettera

