

# Analog Discovery 2 w praktyce (4)

## Analizator stanów logicznych

W artykule opisano sposoby użycia analizatora i generatora stanów logicznych zawartych w Analog Discovery 2.

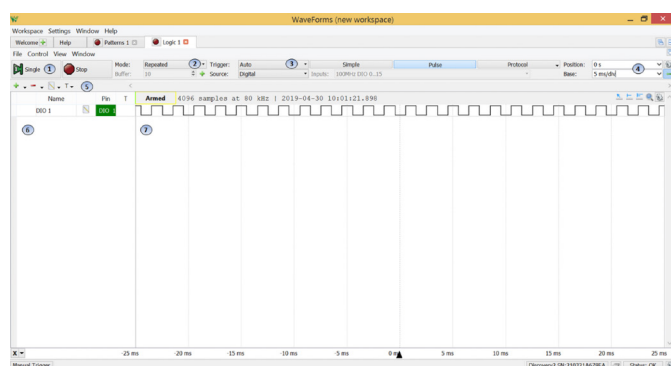
Analog Discovery umożliwia analizę sygnałów logicznych lub ich generowanie. Do dyspozycji zostało oddane 16 niezależnych kanałów o częstotliwości próbkowania 100 MS/s (możliwa jest analiza lub generowanie sygnału o częstotliwości do 50 MHz), w standardzie CMOS 3,3 V.

Przykłady przedstawione w artykule wykorzystują moduły *Logic* i *Patterns* (rysunek 1). *Patterns* służy za generator sygnałów logicznych, *Logic* odpowiada za ich analizę. Aby umożliwić wykonanie przykładów bez dodatkowego sprzętu, będziemy wykonywać analizę sygnałów logicznych wygenerowanych przez AD2 (np. sygnał zegarowy, UART, SPI).

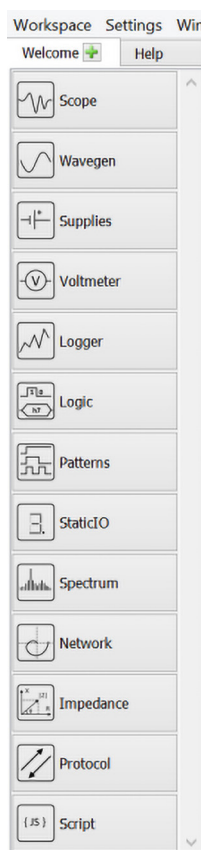
### Analizator stanów logicznych

Na rysunku 2 przedstawiono interfejs analizatora stanów logicznych. Pojedynczy pomiar (uzyskanie ustalonej liczby próbek danych po wyzwoleniu podstawy czasu – *trigger*) uruchamia się za pomocą przycisku *Single*. Pomiar ciągły można uruchomić przyciskiem *Run/Stop* (1). W polu (2) ustawia się tryb akwizycji danych:

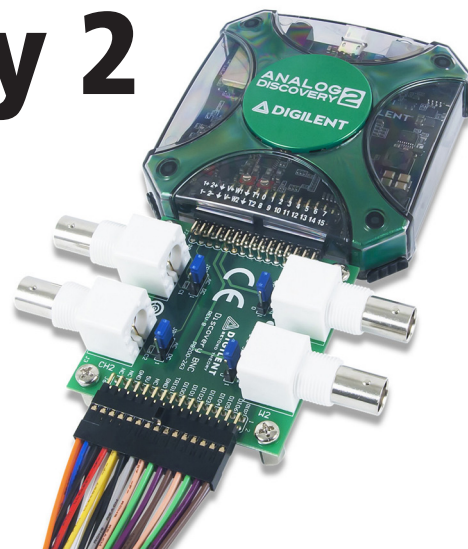
- *Repeated* (podstawa czasu jest wyzwalana wraz z sygnałem synchronizacji *trigger* i ustawiona w punkcie 0 ms),
- *Shift* (okno pomiarowe jest przesuwane wraz z podstawą czasu),
- *Screen* (gdy podstawa czasu osiągnie koniec okna pomiarowego, przeskakuje na początek okna),



Rysunek 2. Interfejs graficzny analizatora stanów logicznych



Rysunek 1. Wybór generatora i analizatora stanów logicznych



- *Record* („nagrywanie” sygnału przy dużej skali podstawy czasu, np. pełne okno zawiera pomiar trwający 1 minutę).

Wyzwalanie podstawy czasu jest konfigurowane za pomocą pól *Trigger*, *Source*, *Simple*, *Pulse*, *Protocol* (3).

*Trigger* zawiera globalne ustawienia podstawy czasu: *None* (brak synchronizacji), *Auto* (automatyczna), *Normal* (konfigurowana przez użytkownika).

*Source* umożliwia ustawienie źródła synchronizacji: sygnały cyfrowe, oscyloskop, generator funkcyjny itd.

*Simple*, *Pulse* i *Protocol* to tryby synchronizacji sygnałami cyfrowymi:

- *Simple* – prosta synchronizacja poziomem logicznym (stan wysoki, niski) lub zboczem (narastające, opadające) (rysunek 3),
- *Pulse* – synchronizacja w momencie wystąpienia impulsu o podanej długości trwania (rysunek 4),
- *Protocol* – synchronizacja w momencie wystąpienia transmisji zgodnej z danym protokołem (np. UART).

Podstawę czasu konfiguruje się za pomocą pól *Position* (pozycja punktu synchronizacji względem punktu „0”) oraz *Base* (rozdzielczość – czas trwania jednej podziałki, np. 5 ms/div) (4).

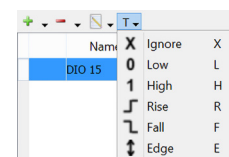
Dodawanie mierzonych sygnałów odbywa się za pomocą przycisku +, usuwanie za pomocą –, kliknięcie w ikonę notatnika umożliwia zmianę nazwy sygnału na zdefiniowaną przez użytkownika, *T* ustawia globalny rodzaj synchronizacji w trybie single (np. zbocze narastające) (5).

W polu (6) umieszczone są mierzone sygnały. Kolumna *Name* zawiera nazwę sygnału, *ikona notatnika* pozwala na jej zmianę. Kolumna *Pin* zawiera nazwę mierzonego sygnału, kolumna *T* umożliwia indywidualne ustawienie trybu synchronizacji dla każdego sygnału w trybie *Simple*.

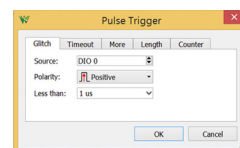
Obszar (7) zawiera pomierzone dane. W lewym górnym rogu opisany jest stan synchronizacji – *Config* (konfigurowanie AD2), *Armed* (oczekiwanie na sygnał synchronizacji), *Busy* (wykonywanie operacji pomiaru), *Done* (pomiar wykonany, wystąpiła synchronizacja podstawy czasu).

Przycisk *View* (powyżej (1)) umożliwia uruchomienie narzędzi pomocniczych (rysunek 5):

- *Data* – wyświetla pomierzone próbki danych,
- *Event* – wyświetla zdarzenia podczas analizy protokołów (np. rozpoczęcie transmisji, zapis konkretnej wartości itd.).

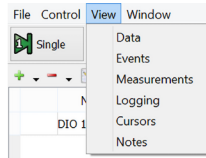


Rysunek 3. Konfiguracja trybu Simple synchronizacji czasu



Rysunek 4. Konfiguracja trybu Pulse synchronizacji czasu (np. ródło sygnału DIO0, stan wysoki przez co najwyżej 1 μs)

- *Measurements* – umożliwia pomiar parametrów sygnałów cyfrowych (częstotliwość, okres, szerokość impulsu itd.),
- *Logging* – zapis pomierzonych danych do pliku,
- *Cursors* – pomiar odległości między dwoma punktami pomiarowymi (różnica w czasie),
- *Notes* – notatki użytkownika.



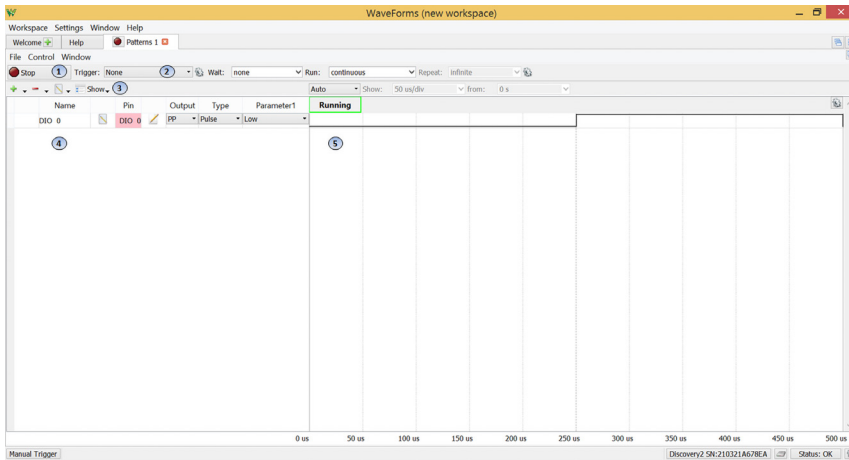
Rysunek 5. Dodatkowe narzędzia ułatwiające analizę pomierzonych danych

– ciągle generowanie sygnału). Alternatywnie można ustawić liczbę powtórzeń sekwencji sygnału logicznego (pole *repeat*).

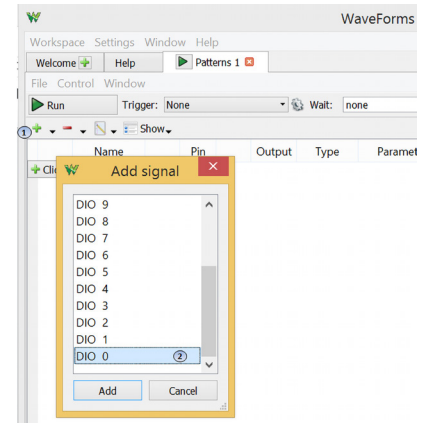
Pole (3) zawiera interfejs dodawania sygnałów logicznych:

- + dodaje sygnał logiczny,
- - usuwa sygnał logiczny,
- ikona notatnika – umożliwia edycję nazwy sygnału oraz edycję parametrów sygnału (rysunek 7),
- Show – ustawienie kolumn, które są widoczne w tabeli aktywnych sygnałów (np. parametry) (4).

Generowanie sygnałów logicznych odbywa się w następujących etapach:



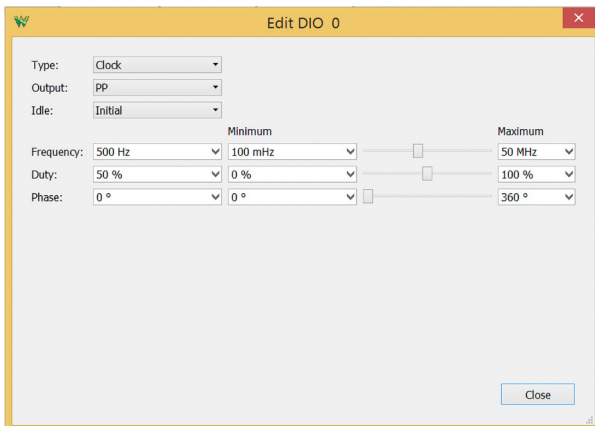
Rysunek 6. Interfejs graficzny generatora sygnałów logicznych



Rysunek 9. Dodanie sygnału do generatora sygnałów logicznych

### Generator stanów logicznych

Na rysunku 6 przedstawiono interfejs generatora stanów logicznych. Przycisk Run/Stop (1) uruchamia generowanie sygnału. Generowany sygnał może być zsynchronizowany z innym sygnałem (2). Pole Trigger umożliwia wybranie źródła synchronizacji – np. *sygnał cyfrowy*, *oscylloskop*, *generator funkcyjny*. W polu Wait można ustawić opóźnienie generowania sygnału po wystąpieniu synchronizacji. Run umożliwia konfigurację czasu trwania generowanego sygnału (*continuous*

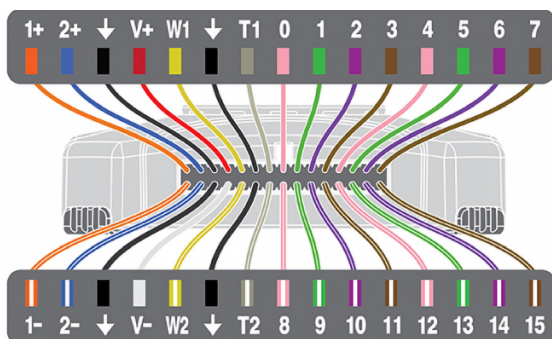


Rysunek 7. Edycja parametrów sekwencji sygnału logicznego

- Wybranie sygnału, dla którego projektujemy sekwencję sygnałów logicznych (kolumna *Name*), ewentualnie edycja nazwy użytkownika;
- Kolumna *Pin* wyświetla pierwotną nazwę sygnału,
- Kolumna *Output* umożliwia konfigurację typu wyjścia sygnału:
  - PP (push-pull – standardowe wyjście),
  - OD (open-drain – otwarty dren [równoznaczne z otwartym kolektorem]),
  - OS (open-source – otwarte źródło [równoznaczne z otwartym emiterem]).

Wyjścia OD i OS przydają się w implementacji protokołów z tzw. iloczynem na drucie, gdzie nie ma jednostki sterującej arbitrażem na magistrali, np. I<sup>2</sup>C.

- Kolumna *Type* – ustawienie rodzaju sekwencji sygnału logicznego:
  - *constant* – stała wartość 0 lub 1,
  - *clock* – zegar – parametry: częstotliwość, wypełnienie (*duty cycle*), przesunięcie fazy,
  - *pulse* – impuls – parametry: stan logiczny na początku sekwencji, czas trwania stanu wysokiego i niskiego,
  - *random* – losowe próbki,
  - *custom* – sekwencja użytkownika.
- *Parameters* – liczba parametrów zależna od rodzaju sekwencji.



Rysunek 8. Połączenie sygnałów TX i RX

### Przykład: symulacja protokołu UART

W tym przykładzie moduł Patterns odpowiada za generowanie transmisji UART (TX), a moduł Logic za analizę protokołu UART (RX):

- Sygnał DIO 0: TX, generowanie transmisji UART,
- Sygnał DIO 1: RX, analiza transmisji UART.

Aby móc analizować protokół UART, należy połączyć sygnał 0 i 1 (rysunek 8).

Generowanie protokołu UART – Patterns

1. Aby dodać sygnał, klikamy ikonę + (1), następnie wybieramy DIO 0 z listy dostępnych sygnałów (2) (rysunek 9).
2. Następnie konfigurujemy sygnał (rysunek 10):
  - a. Edycja nazwy sygnału – klikając w notatnik (1), zmieniamy wyświetlaną nazwę sygnału na TX (2). Fizyczny sygnał jest wyświetlany w polu *Pin* (3).

Tabela 1. Sekwencja bitów transmisji UART

Bit startu	D0	D1	D2	D3	D4	D5	D6	D7	Bit stopu
0	0	1	0	0	0	0	0	1	1

- b. Ustawiamy typ wyjścia na PP (*push-pull*) (4).
  - c. Zmieniamy typ sekwencji na *custom* (użytkownika) (5).
  - d. Przechodzimy do trybu edycji sekwencji, klikając w notatnik (6).
3. Edycja generowanej sekwencji.

UART jest podstawową transmisją szeregową – składa się z jednej linii danych, transmitującej 8-bitowe dane. Dane są przesyłane asynchronicznie, tzn. odstęp między danymi jest ustalany z góry, nie występuje linia zegarowa. Transmisja danych odbywa się w następujący sposób:

- Bit startu (stan niski – 0),
- 8 bitów danych (rozpoczynających się od bitu najmłodszego – LSB),
- Opcjonalny bit parzystości (określa, czy występuje parzysta liczba „jedynek” – używany jest do kontroli poprawności transmisji, nie jest używany w przykładzie),
- Bit stopu (stan wysoki – 1).

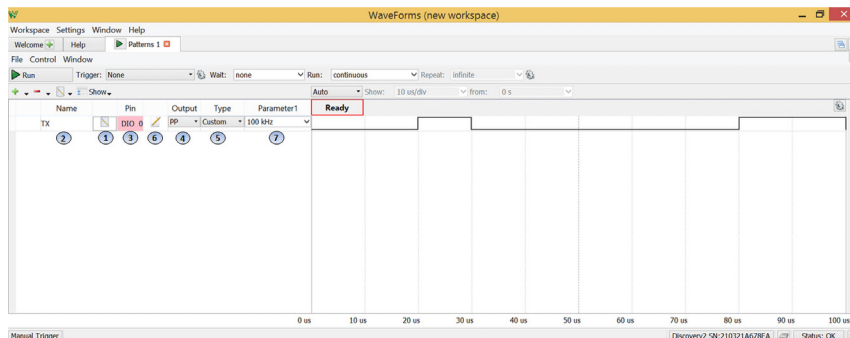
W przykładzie wygenerujemy transmisję UART bajtu danych o wartości 0x82 (10000010), co daje sekwencję bitów przedstawioną w tabeli 1.

Interfejs graficzny edycji sekwencji jest przedstawiony na rysunku 11.

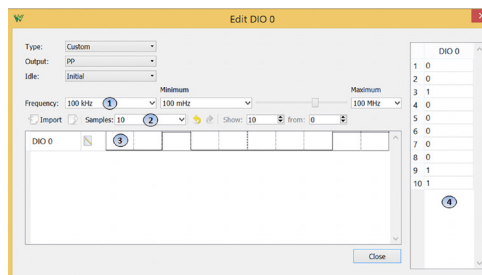
Edycja sekwencji polega na ustaleniu wartości poszczególnych próbek:

1. Na początku ustawiamy częstotliwość próbkowania (1) i liczbę próbek w sekwencji (2) – w przykładzie użyjemy prędkości 100 kHz i 10 próbek (tabela 1).
2. Klikając na pola poszczególnych próbek (3), zmieniamy stan logiczny próbki. Alternatywnie można wpisać wartość poszczególnych próbek w edytorze tekstowym (4).

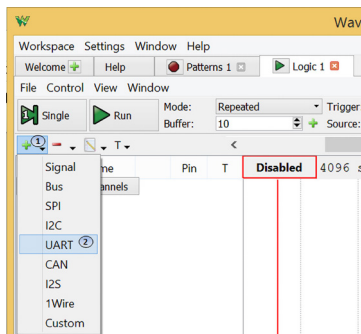
Po konfiguracji sekwencji należy uruchomić generowanie sygnału (Run). Ustawienia synchronizacji sygnału: *Trigger: None, Wait: none, Run: continuous*.



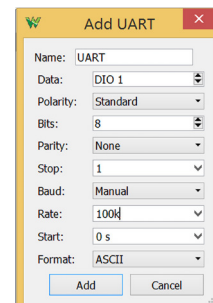
Rysunek 10. Interfejs graficzny konfiguracji generowanego sygnału



Rysunek 11. Interfejs graficzny edycji sekwencji



Rysunek 12. Dodawanie sygnału, który ma być mierzony

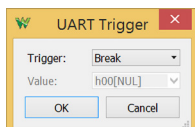


Rysunek 13. Konfiguracja protokołu

**Analiza protokołu UART – Logic**

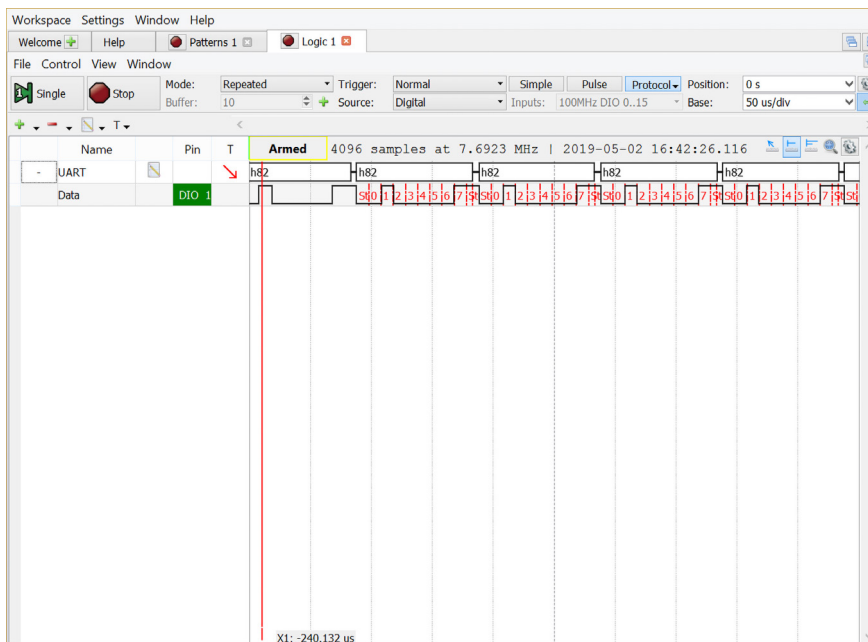
1. Aby dodać sygnał protokołu UART, należy kliknąć ikonę + (1) i z rozwijanej listy wybrać UART (2) (rysunek 12).
2. Po dodaniu sygnału otworzy się okno dialogowe z konfiguracją protokołu UART (rysunek 13). Należy ustawić wszystkie opcje tak jak na rysunku.
3. Następnie należy skonfigurować wyzwalanie podstawy czasu (*trigger*):
  - Mode: repeated,
  - Source: digital,
  - Tryb synchronizacji: Protocol → UART → Break (rysunek 14).
  - Podstawa czasu: *position 0, base 50 μs/div*.

Po konfiguracji klikamy przycisk *Run*, uruchamiając akwizycję danych. Jeżeli wszystkie ustawienia zostały wykonane poprawnie, na ekranie powinniśmy ujrzeć transmisję bajtu 0x82 (rysunek 15).



Rysunek 14. Konfiguracja synchronizacji UART

Jakub Wiczyński



Rysunek 15. Pomiar protokołu UART – transmisja bajtu 0x82