

myRIO – platforma edukacyjna od National Instruments (2)

Uniwersalny przyrząd laboratoryjny

Firma National Instruments zaproponowała kolejny system przeznaczony dla studentów. Tym razem jest on oparty o technologię wykorzystaną w popularnej platformie CompactRIO. Pozwala na przećwiczenie zagadnień związanych z automatyką, robotyką, systemami kontrolno-pomiarowymi, a szczególnie z wbudowanymi systemami operacyjnymi czasu rzeczywistego.

Zastanawiając się nad wykorzystaniem platformy postanowiłem przygotować przyrząd przydatny w pracowni elektroniki, zawierający możliwie dużo elementów systemu pomiarowego wykorzystywanych podczas uruchamiania i testowania układów. Moim celem było również maksymalnie

wykorzystanie dostępnych zasobów myRIO, czyli FPGA oraz wejść i wyjść układu. Powstałe narzędzie może być świetnym rozwiązaniem dla studentów stawiających pierwsze kroki w samodzielnym budowaniu i uruchamianiu urządzeń elektronicznych. Myślę, że również profesjonaliści zajmujący

się elektroniką również docenią jego zalety, chociażby tę, że zestaw przyrządów laboratoryjnych w jednej, niewielkiej obudowie można zawsze mieć „pod ręką”.

Założenia projektu możliwe do uzyskania w posiadanej przeze mnie konfiguracji:

1. Oscyloskop:
 - Liczba kanałów: 2.
 - Częstotliwość próbkowania: 250 kHz/kanał.
 - Podstawa czasu: 100 μ s/dz. – 200 s/dz.
 - Czułość podstawowa: 2 V/dz.
 - Możliwość sterowania modulem wzmacniaczy wejściowych.
 - Pomiar wielu parametrów sygnału.

2. Generator funkcyjny:

- Liczba kanałów: 2.
- Przebiegi generowane: sinusoidalny, prostokątny.
- Zakres częstotliwości: 0,01 Hz...100 kHz.

3. Analizator stanów logicznych:

- Liczba kanałów: 16.
- Częstotliwość próbkowania do 40 MHz.
- Wyzwalanie: dowolnym wzorcem.

4. Generator PWM:

- Liczba kanałów: 1.
- Zakres częstotliwości: 40 Hz...40 kHz.

5. Interfejsy komunikacyjne:

- 2×UART
- SPI.
- I²C.

6. Komunikacja z komputerem za pomocą USB lub Wi-Fi.

z przyrządów dysponuje oddzielnym panelem sterującym, pomiędzy którymi przechodzi się wybierając odpowiednią zakładkę. Uruchomienie aplikacji przebiega w dwóch etapach. Jako pierwszy należy uruchomić plik **RT Main**, odpowiedzialny za sterowanie przyrządem po stronie myRIO i właściwe skonfigurowanie struktury FPGA. Jako kolejny uruchamia się **Desktop Main** pamiętając o wcześniejszym wpisaniu właściwego adresu IP. Aplikacja pozwala zapamiętywanie w pliku wyników pomiarów oscyloskopowych oraz analizatorem stanów logicznych.

Oscyloskop

Oscyloskop mierzy sygnał analogowy z dwóch wejść różnicowych. Wykrywając spełnienie predefiniowanego warunku wyzwala pomiar w trybie ciągłym lub pojedynczym. Następnie, gromadzi określoną liczbę próbek i przesyła do komputera. Podczas pracy w trybie ciągłym kolejne wyzwolenie

następuje po ponownym spełnieniu przez mierzony sygnał warunków wyzwolenia. W takim trybie pracy na ekranie widzimy stabilny i niepłynący przebieg, ale są krótkie odcinki czasu, w których akwizycja nie jest prowadzona. Można rozbudować oscyloskop na przykład korzystając z kolejki **RT FIFO** i prowadzić akwizycję w czasie rzeczywistym od momentu wyzwolenia aż do zatrzymania. Gromadzone dane należy przesyłać do komputera partiami i zapisywać na dysku. Dzięki temu mogą być wykryte przypadkowo pojawiające się zakłócenia sygnału.

Na **rysunku 1** pokazano panel oscyloskopu pozwalający na sterowanie wszystkimi jego funkcjami. Elementy panelu czołowego:

- **Time** – podstawa czasu.
- **CH1, CH2** – wzmocnienie każdego z kanałów. Przy współpracy z płytką wzmacniaczy możemy wybrać jedno z siedmiu wzmocnień, od 100 mV/dz. do 20 V/dz. Jeśli nie dysponujemy tą płytką, wzmocnienie powinno wynosić 2 V/dz.
- **Trigger** – sposób wyzwolenia: wyzwolenie jednokrotne, praca w trybie ciągłym, źródło wyzwolenia – kanał 1 lub 2, poziom wyzwolenia, sposób wyzwolenia: zboczem narastającym, opadającym lub poziomem (ignorując zbocza).

Przebiegi można obserwować na dużym czytelnym obrazie.

Dzięki kursorom możemy mierzyć odstępy czasowe pomiędzy wybranymi próbkami (dx) i różnicę napięć (dy), oraz odwrotność (1/dx), co odpowiada częstotliwości wynikającej z odstępu kursorów na osi x. Dodatkowo, możemy włączyć jedną z wielu funkcji umożliwiających pomiary: częstotliwości, amplitudy, fazy, składowej stałej napięcia, wartości skutecznej napięcia RMS, maksymalny i minimalny poziom sygnału, okres, czas trwania impulsu, wypełnienie oraz obserwowanie transformaty FFT. Możemy również obserwować aktualnie mierzone wartości i postęp w zbieraniu próbek. Jest to informacja przydatna podczas pomiaru sygnałów wolnozmiennych.

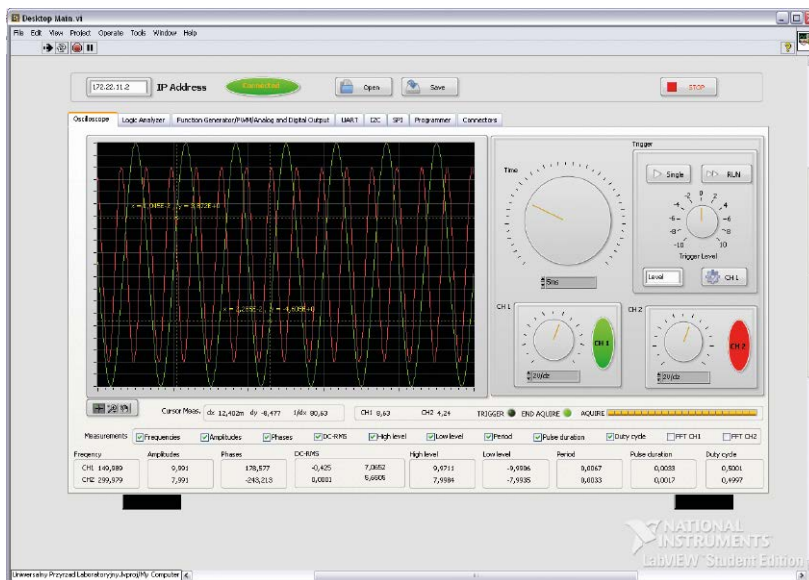
Analizator stanów logicznych

Okno analizatora stanów logicznych pokazano na **rysunku 2**. Przebiegi są wyświetlane na dużym, czytelnym wykresie. Kontrolka **Current Data** zawiera aktualny stan wejść. Jest ona aktualizowana co kilkanaście milisekund, niezależnie od stanu pracy analizatora. Do konfiguracji pomiaru służą następujące opcje menu:

- **Sampling Freq.** – częstotliwości próbkowania.
- **Data To Acquire** – liczba próbek. Wartość maksymalna wynosi 50000.
- **Mask** – maska, która pozwala na zamaskowanie stanu bitów nieistotnych podczas wyzwolenia akwizycji.

Aplikacja Pomiarowa

Wszystkie przyjęte powyżej założenia zrealizowano w pojedynczej aplikacji. Każdy



Rysunek 1. Panel oscyloskopu



Rysunek 2 Okno analizatora stanów logicznych z widocznym na liniach D0 i D1, przebiegiem transmisji I²C, na linii D6 przebieg PWM

- **Pattern** – wzorzec bitowy; konfiguracja bitów wyzwalająca pomiar.

Proponuję zastanowić się nad modyfikacją układu wyzwalania, aby wykrywał określone sekwencje bitów na wybranej linii na przykład, wykrywając przesłanie po magistrali I²C określonego adresu.

Generator funkcyjny

Na rysunku 3 pokazano panel umożliwiający sterowanie generatorem funkcyjnym. Do ustawienia mamy:

- Częstotliwość przebiegu.
- Fazę przebiegu.
- Typ przebiegu.
- Skalę.
- Składową stałą.

Wyjścia analogowe pozwalają na ustawienie napięcia wyjściowego z przedziału od 0...5 V z rozdzielczością 1,22 mV. Nie przewidziałem tutaj generowania przebiegu. Dlatego zrezygnowałem z wyjść audio. Ich budowa nie pozwala na przenoszenie napięć stałych – pracują tylko z sygnałami zmiennymi.

Wejścia wyjścia cyfrowe pozwalają na ręczne ustawienie trybu pracy i stanu każdego bitu w porcie A. Należy pamiętać aby nie wykorzystywać pinów zajętych przez interfejsy I²C, SPI, i generator PWM, podczas korzystania z nich.

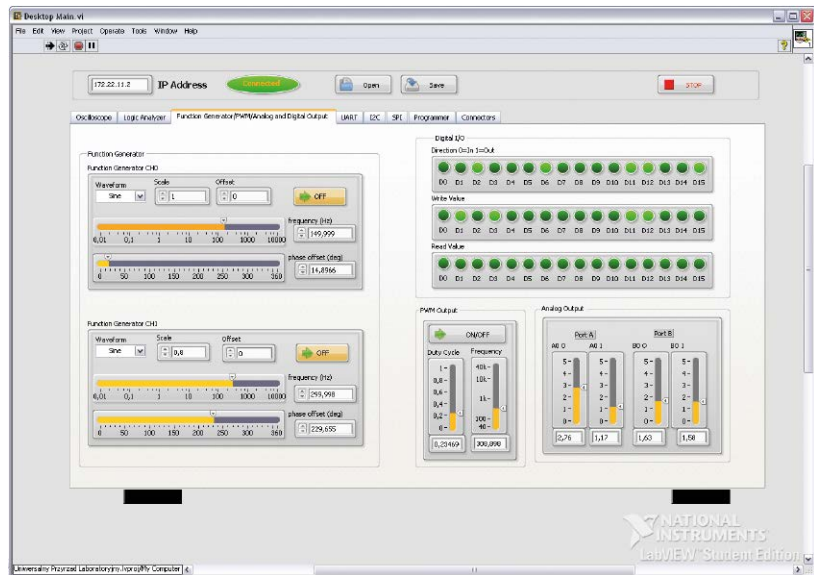
Generator PWM

Generatory PWM są wykorzystywane np. do sterowania serwo mechanizmami, przetwornicami napięcia itp. W porcie A udało się pozostawić jeden z generatorów. Możemy programować częstotliwość w przedziale 40 Hz...40 kHz i współczynnik wypełnienia sygnału przedziale 0...1. Należy pamiętać, że jego włączenie zajmuje linię DIO8 i nie można jej wykorzystywać jako wejście lub wyjście cyfrowe.

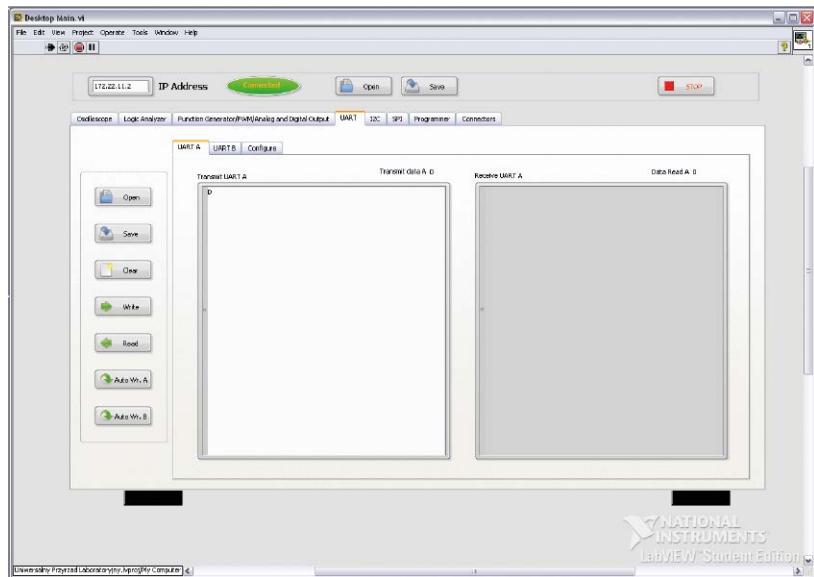
UART

Rysunek 4 przedstawia okno komunikacji przez dostępne interfejsy UART. Ich konfiguracja odbywa się w zakładce *Configure* i obejmuje takie parametry, jak: szybkość transmisji, liczba bitów danych, bit stopu, kontrolę parzystości, kontrolę przepływu, timeout, znak terminujący, odstęp czasowy pomiędzy kolejnymi transmisjami (wykorzystany w trybie automatycznego odpytywania). Do komunikacji przewidziano dwa bufor – nadawczy i odbiorczy.

Dane do bufora nadawczego można wpisywać z klawiatury lub wczytać z pliku tekstowego. Dane znajdujące się w buforze odbiorczym mogą być zapisane do pliku tekstowego. Zmieniając styl wyświetlania danych w buforze za pomocą *Display Style*, możemy wpisywać wartości znakowe lub liczby w kodzie szesnastkowym. Dane trafiające do portów



Rysunek 3. Przedstawia panel dostępu do generatorów funkcyjnych, generatora PWM, wyjść analogowych i wejść/wyjść cyfrowych



Rysunek 4. Panel interfejsów UART



Rysunek 5. Panel do transmisji przez interfejs I²C

automatycznie przesyłane są do buforów odbiorczych i widoczne w oknie terminala. Przewidziano również tryb automatycznego odpytywania z zadaniem odstępem czasowym. Umożliwiający, na przykład, rejestrowanie temperatury lub innych parametrów przez dłuższy czas.

Do bufora nadawczego można wpisać odpowiednie polecenie przyrządu, niektóre urządzenia potrafią przyjmować ciąg poleceń i wykonywać je kolejno. Zawartość bufora będzie wysyłana z do portu UART, a odpowiedź rejestrowana w buforze odbiorczym. Jako przykład mogę podać pomiar temperatury multimetrem METEX. Mierzona wartość jest wysyłana do portu szeregowego po odebraniu litery „D”, jednobajtowego polecenia. Wpisując literę D, do bufora nadawczego, ustalając odstęp czasowy możemy rejestrować wyniki pomiaru. Bardziej zaawansowane przyrządy wykorzystujące polecenia zgodne z standardem SCPI, mogą przyjmować ciąg poleceń, wówczas cały ciąg jest przesyłany jednocześnie a odpowiedzi urządzenia są rejestrowane. Należy pamiętać o dostosowaniu poziomów sygnału gdy chcemy korzystać z standardu RS232.

I²C

Komunikacja przez I²C jest często stosowana do wymiany danych pomiędzy układami scalonymi w obrębie jednego urządzenia. Moim celem było przygotowanie prostego narzędzia umożliwiającego wysyłanie poleceń i modyfikowanie zawartości rejestrów wewnętrznych układów. Korzystając z oscyloskopu i analizatora stanów logicznych możemy jednocześnie śledzić wymianę danych i odpowiedź układu na zmianę wybranych parametrów. Jako pierwszy powstał tryb terminalowy, w którym zawartość z pół *Data* przesyłana jest po magistrali pod wskazany adres. Można go wykorzystać do modyfikowania zawartości pamięci np. EEPROM przesyłając całą zawartość jednej strony pamięci, lub zapisać dane do kolejnych rejestrów.

Tryb rejestrowy jest przydatny podczas pracy z układami innego rodzaju np. przetwornikami, umożliwi modyfikację zawartości wybranych rejestrów. Dodatkowo przewidziana została możliwość wcześniejszej konfiguracji układu. Aby ułatwić pracę nie szukając za każdym razem danych katalogowych, po pierwszym ustawieniu wszystkich niezbędnych wartości, można je zapisać w pliku konfiguracyjnym. Z którego później mogą być wczytywane. Dane zapisywane do pliku konfiguracyjnego to zawartość kontrolek: *I2C Device Info*, *I2C Configure*, *I2C Register*, *I2C Start Sequence*:

- **I²C Configure** – konfiguruje sposób pracy interfejsu.

- **I²C Transfer Rate** – wybór szybkości pracy interfejsu.
- **WP line** – określa, która linia jest dołączona do sygnału WP. Sygnał występuje w pamięciach umożliwiając zabezpieczenie pamięci przed przypadkowym zapisem. Wybrana linia ustawiana jest w określony stan przed rozpoczęciem zapisu. Można ją wykorzystać na przykład do wyzwolenia analizatora stanów logicznych. Wówczas rejestracja rozpocznie się dokładnie w momencie aktywowania tej linii, pozwoli to precyzyjnie określić które przebiegi obserwujemy. Częstotliwość próbkowania analizatora najlepiej ustawić dwukrotnie większą niż szybkość pracy magistrali.
- **WP Line Polarity** – określa stan aktywny dla linii WP.

Kontrolka *I2C Selection of operations* określa tryb pracy.

Terminal – w tym trybie wszystkie operacje dotyczą kontrolki **I²C Write**, **I²C Read**.

- **Slave Address (7-bit)** – określa adres układu
- **Start Address** – określa adres komórki pamięci od której zacznie się zapis/ odczyt
- **Bytes to Read** – określa ile bajtów będzie odczytanych.
- **Data** – zawiera dane które będą wysłane po kolej na magistralę I²C, lub były z niej odczytane.

Wykorzystując tryb *Register* operujemy na rejestrach układów dołączonych do I²C. Operacje zapisu i odczytu dotyczą rejestrów, których adresy oraz wartości pobierane są z tabel *I2C Register* oraz *I2C Start Sequence*.

Kontrolka *I2C Device Info* zawiera podstawowe informacje o układzie scalonym. Nie wszystkie dane są istotne i niektóre mogą być pominięte lub mieć wartości domyślne. Na przykład, w przetworniku pomiarowym nie istotna jest wielkość pamięci czy rozmiar strony.

Znaczenie pół *I2C Device Info*:

- **Device Name** – symbol układu.
- **Description** – krótki opis tekstowy.
- **Interface** – typ interfejsu; obecnie I²C lub SPI.
- **Slave Address (7-bit)** – adres układu istotny dla interfejsu I²C.
- **Page Size** – rozmiar strony pamięci (aktywny dla pamięci).
- **Memory Size** – rozmiar pamięci w bajtach (aktywny dla pamięci).
- **Address Code** – przewidziany dla układów z interfejsem SPI, nieistotny dla interfejsu I²C.

Zmienna *I2C Register* zawiera opis rejestrów układu podłączonego do magistrali. Należy przygotować ją samodzielnie odczytując ze specyfikacji technicznej

adresy i domyślne wartości rejestrów oraz operacje, które można na nich przeprowadzić. Klikając na tabeli prawym przyciskiem myszki z menu kontekstowego można wybrać *Add Register*, po czym zostanie dodana nowa linia pozwalająca na wpisanie nazwy, zawartości i adresu rejestru. Wybierając *Delete Register* usuwamy wybraną linię. Klikając dwukrotnie na wybranym wierszu zmieniamy rodzaj operacji wykonywanej na rejestrze: W – zapis, R – odczyt, R/W – zapis i odczyt. Podczas operacji zapisu wybierane są tylko rejestry z atrybutami W i R/W, podobnie podczas operacji odczytu wybierane są rejestry z atrybutami R i R/W.

Opis kolumn:

- **Register** – nazwa rejestru.
- **Operation** – operacje, które chcemy lub możemy wykonać.
- **Address** – adres rejestru, liczba szesnastkowa w postaci x0000.
- **Value** – wartość zapisywane do rejestrów lub z nich odczytywane zależnie od pola *Operation*. Format zapisu x00; tylko liczby w formacie szesnastkowym.
- **Description** – może zawierać krótki opis tekstowy, np. znaczenia bitów.

Zaznaczenie pola *Use Start Sequence* spowoduje, że operacje zapisu rejestrów z tabeli *I2C Register*, są poprzedzone zapisem rejestrów zawartych w tabeli *I2C Start Sequence*.

Moim zamierzeniem było umożliwienie wstępnej konfiguracji układów, np. ustawienie wzmocnień, filtrów wejściowych i innych parametrów pracy przetworników pomiarowych. Wówczas tabela *I2C Register*, nie zawierałaby kluczowych dla pracy układu rejestrów, a odznaczenie tego pola zabezpieczy przed ich przypadkową modyfikacją.

W tabeli *I2C Start Sequence* są zawarte wartości mogące wstępnie konfigurować parametry pracy układu:

- **Register** – nazwa rejestru.
- **Address** – adres rejestru w kodzie szesnastkowym, format x0000.
- **Value** – wartość w kodzie szesnastkowym, format x00.
- **Description** – krótki opis tekstowy np. znaczenia bitów.

SPI

Na **rysunku 6** pokazano panel do komunikacji poprzez interfejs SPI. Cel i założenia są takie same, jak dla I²C, chociaż wymiana komunikatów jest nieco inna. Dlatego panel został rozdublowany o dodatkowe funkcje umożliwiające pracę z układami pamięci. Większości kontrolki jest taka sama, w interfejsie I²C, ale są one zdublowane, aby umożliwić jednoczesną pracę z układami na magistrali I²C i SPI.

Tabela *SPI Configuration* zawiera nastaw parametrów pracy interfejsu SPI:

- **Frequency** – częstotliwość zegarowa SPI.
- **Clock Phase** – faza zegara.
- **Clock Polarity** – polaryzacja zegara.
- **Data Direction** – kolejność transmisji bitów od najmniej znaczącego do najbardziej znaczącego lub odwrotnie.
- **Frame Length** – długość pola bitowego.
- **Chip Select Pin** – linia aktywowana przed rozpoczęciem transmisji danych.
- **Chip Select Polarity** – poziom aktywny linii *Chip Select Pin*.
- **WP** – sygnał aktywowany podczas zapisu do pamięci.
- **WP Polarity** – poziom aktywny linii *WP*.

Tabela *SPI Device Info* jest odpowiednikiem *I2C Device Info* i zawiera informacje o układzie. Wszystkie pola mają takie samo znaczenie, ale wyjaśnienia wymaga tutaj znaczenie *Address Code*, które ma istotne znaczenie podczas pracy z pamięciami w zakładce programatora. Analizując zapisy do pamięci można wyróżnić odmienne sposoby przesyłania adresów. W „małych” pamięciach np. 25AA040A (Microchip), starsza część adresu jest przesyłana wraz z właściwą instrukcją, zakodowaną na bicie czwartym (rysunek 7). Po nim jest przesyłana młodsza część adresu, a następnie dane. Algorytm zapisu do pamięci przewiduje kodowanie adresu na starszych bitach pola zawierającego instrukcje. Aby wybrać taki sposób kodowania adresów pole *Address Code* musi być równe *Add Inst*.

W drugiej sytuacji, pokazanej na rysunku 8, jako pierwsza jest przesyłana instrukcja, a następnie 16-bitowy adres i dane. Taki sposób kodowania wybieramy nadając polu *Address Code* wartość *Normal*. Możliwe, że należałoby przewidzieć jeszcze inne sposoby przesyłania adresów uwzględniając inne układy pamięci. Można to zrobić dodając tutaj nowe wartości i rozbudowując algorytmy we właściwych funkcjach.

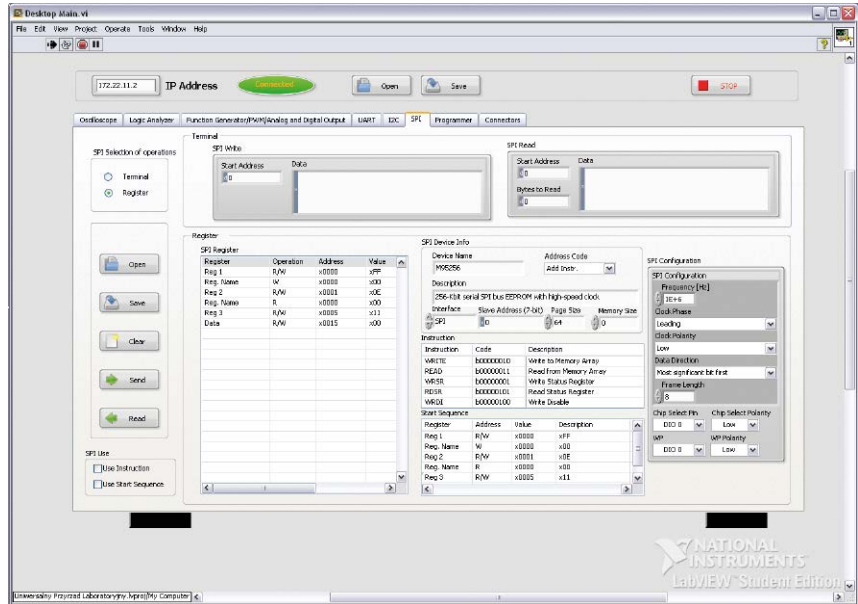
Zmienna *SPI Selection of operations* – określa tryb pracy.

Tryb *Terminal* pracuje jak poprzednio, ale nie ma tu adresów urządzenia podrzędnego. Operacje zapisu i odczytu dotyczą kontrolki *SPI Write* i *SPI Read*:

- **Start Address** – określa adres komórki pamięci, od której zacznie się zapis/odczyt.
- **Bytes to Read** – określa liczbę bajtów odczytanych.
- **Data** – pole danych.

Tryb *Register* operuje na rejestrach układów dołączonych do SPI. Operacje zapisu i odczytu dotyczą rejestrów określonych w tabelach *SPI Register* oraz *SPI Start Sequence*.

Tabela *SPI Register* zawiera opis rejestrów układu dołączonego do magistrali. Jest odpowiednikiem tabeli *I2C Register*. Tabela *SPI Start Sequence* zawiera dane do wstępnej konfiguracji układu. Jest odpowiednikiem *I2C Start Sequence*.



Rysunek 6. Panel do transmisji przez interfejs SPI

Operacje zapisu i odczytu danych z pamięci dołączonych do magistrali SPI wymagają użycia odpowiednich instrukcji, co pociągnęło za sobą konieczność uwzględnienia tego w algorytmach. Mimo iż kody instrukcji często się pokrywają, to dla niektórych układów mogą się różnić. Dlatego przewidziałem możliwość ich indywidualnego ustawienia. Są one również zapisywane w pliku konfiguracyjnym. Pozwala to na obsłużenie większej liczby układów pamięci.

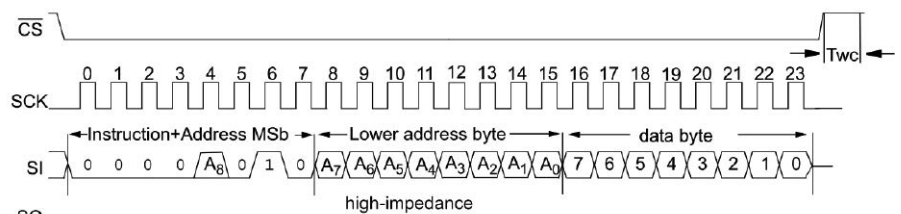
Zmienna *SPI Instruction* zawiera instrukcje niezbędne do wykonania operacji na pamięciach. Ich kolejność jest ściśle określona i nie można jej zmieniać. Należy tylko wpisać właściwe kody w odpowiednie komórki. Akceptowany jest format binarny b00000000 lub szesnastkowy x00. Znaczenie pól:

- **Instruction** – skrócony literowy kod instrukcji.
- **Code** – właściwy kod instrukcji.
- **Description** – krótki tekstowy opis.

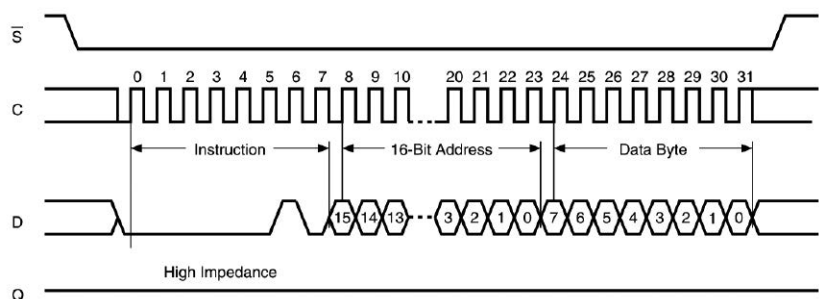
W chwili obecnej przewidziałem tylko 6 podstawowych instrukcji. Ich kolejność musi być taka, jak podana:

1. **WRITE** – zapis do pamięci.
2. **READ** – odczyt z pamięci.
3. **WRST** – zapis rejestru statusu.
4. **RDSR** – odczyt rejestru statusu.
5. **WRDI** – blokada zapisu.
6. **WREN** – odblokowanie zapisu.

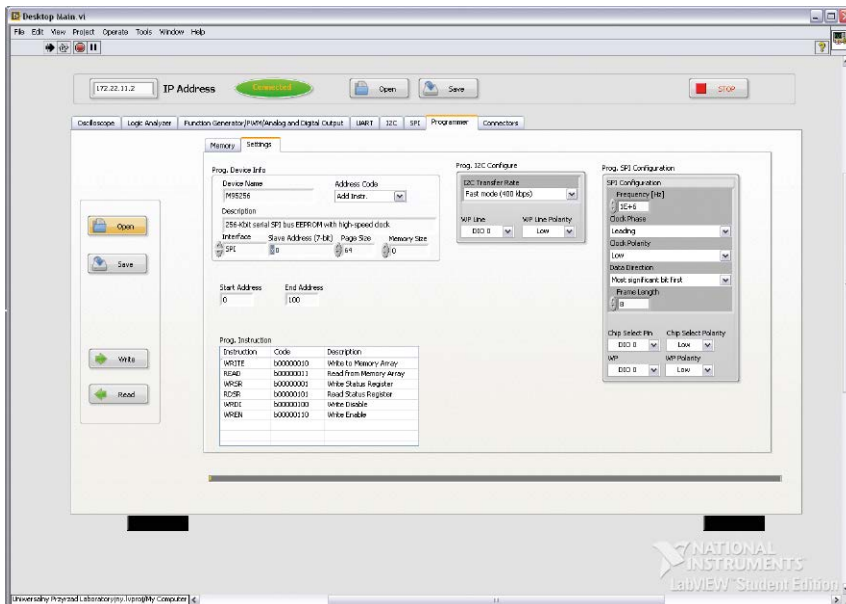
Zaznaczenie pola *Use Instruction* powoduje wybór algorytmu przystosowanego do pracy z pamięciami i używającego kodów instrukcji z tabeli *SPI Instruction*. Zaznaczenie pola *Use Start Sequence* spowoduje, że zapis rozpocznie się od rejestrów wyspecyfikowanych w tabeli *SPI Start*



Rysunek 7. Przebiegi podczas zapisu do pamięci Microchip Technology 25AA040A



Rysunek 8. Przebiegi podczas zapisu do pamięci ST M95256



Rysunek 9. Zakładka z ustawieniami programatora

Sequence, a następnie będą zapisywane rejestry z tabeli *SPI Register*.

Programmer

Umożliwia programowanie pamięci dołączonych do magistrali I²C i SPI. Obsługę różnych układów umożliwiają właściwie przygotowane pliki konfiguracyjne.

Okno programatora pokazano na **rysunku 9**. Gdy jest aktywna zakładka *Memory*, przycisk *Open* wczytuje zawartość pamięci z plików w formacie Intel HEX. Wybranie *Save* zapisuje zawartość odczytanej pamięci w formacie Intel HEX. Po przejściu do zakładki *Settings* te same przyciski pozwalają na wczytanie plików konfiguracyjnych. Zawierające informacje o ustawieniach interfejsu, kodach instrukcji, pojemności i organizacji pamięci. Oczywiście, wszystkie można ustawić bez wczytywania, ale lepiej po pierwszej konfiguracji zapisać ustawienia, nie tracąc na nie czasu następnym razem. Znaczenie wszystkich pól zostało opisane podczas opisu zakładki I²C i SPI, a dodatkowo:

- **Prog. Device Info** – jest odpowiednikiem kontrolki z ...*Device Info* zakładki I²C i SPI.
- **Prog. I2C Configure** – zawiera konfigurację interfejsu I²C.
- **Prog. SPI Configuration** – zawiera konfigurację interfejsu SPI.
- **Prog. Instruction** – zawiera kody instrukcji dla pamięci SPI.
- **Start Address** – adres początku danych, domyślnie jest to 0. Zmieniając tę wartość należy zadbać, aby adres wskazywał początek wybranej strony, a nie dowolną komórkę.
- **End Address** – powinien wskazywać adres ostatniej strony pamięci. Domyślnie jest ustawiany zgodnie z wartością pola *Memory Size* w *Prog. Device Info*.

Connectors

W tej zakładce umieściłem opis wyprowadzeń, ich funkcji i rozmieszczenie w złączu. Dzięki temu nie trzeba ich szukać przed każdą zmianą połączeń.

Przygotowanie plików konfiguracyjnych

Podjąłem próbę obsługi wielu układów za pomocą pojedynczego programu. Niestety, zróżnicowanie formatów ramek szczególnie przy zastosowaniu interfejsu SPI, znacznie utrudnia to zadanie. Dlatego do wyboru właściwego algorytmu i konfiguracji interfejsów wykorzystałem pliki, w których zapisuję niezbędne informacje. Pliki te są wykorzystywane podczas pracy z magistralami I²C i SPI. Zawierają informacje o każdym interfejsie. Jeśli dane nie dotyczą danego układu, mogą mieć przypadkowe wartości lub być puste.

Plik konfiguracyjny jest plikiem tekstowym zapisanym z rozszerzeniem „.con”. W projekcie do ich przechowywania przewidziałem katalog *Config File*. Strukturę pliku pokazano na **listingu 1**. Można go przygotować np. w notatniku czy dowolnym edytorze tekstowym. Ale wygodniej zrobić to w programie w jednej z zakładki *I2C*, *SPI* lub *Programmer* → *Settings*, wypełniając istotne dla danego układu pola i zapisując plik.

Obsługa plików konfiguracyjnych została zrealizowana z wykorzystaniem funkcji z palety *File I/O* → *Configuration File*

Listing 1. Struktura pliku konfiguracyjnego

```
[Device Info] - nazwa sekcji nie można jej zmieniać
Device Name = „ nazwa układu „
Description = „Krótki opis”
Interface = 0 = I2C, 1 =SPI
Slave Address = adres układu na magistrali I2C
Page Size = rozmiar strony pamięci dotyczy pamięci
Memory Size = całkowity rozmiar pamięci
Address Code = 1 - sposób kodowania adresu

[I2C Info] - konfiguracja I2C
I2C Frequency = 1 - 0 = 100kHz, 1 = 400kHz
WP Line = 9 - numer linii WP
WP Polarity = 0 - polaryzacja WP

SPI Info] - konfiguracja SPI
SPI Frequency = 1000000,000000
Clock Phase = 0
Clock Polarity = 0
Data Direction = 0
Frame Length = 8
CS Pin = 0 - numer linii CS
CS Polarity = 0 - polaryzacja CS
WP Pin = 0 - numer linii WP
WP Polarity = 0 - polaryzacja WP

[Register Section] - nazwa sekcji, sekcja zawiera zawartość tabeli Register
Register Section Size = 6 - rozmiar, informuje ile rekordów zostało zapisanych w tej sekcji

[Register 0] - pierwszy rekord
Reg. Name = „nazwa rejestru”
Operation = „W” - rodzaj operacji R=odczyt, W= zapis R/W odczyt i zapis
Address = „x0000” - adres rejestru
Value = „xFF” - wartość rejestru
Description = „krótki opis rejestru”
...

[Instruction Section] - nazwa sekcji, sekcja zawiera kody instrukcje dla pamięci SPI
Instruction Section Size = 6 -rozmiar informuje ile rekordów zostało zapisanych w tej sekcji
...

[Instruction Nr. 0] - pierwszy rekord

Instruction = „WRITE” - typ instrukcji
Instr. Code = „b0000010” - kod instrukcji
Description = „Write to Memory Array” - opis
...

[Init Sequence Section] - nazwa sekcji, sekcja zawiera zawartość tabeli Init Sequence.
Init Sequence Size = 6 - rozmiar informuje ile rekordów zostało zapisanych w tej sekcji.

[Init Sequence Nr. 0] - pierwszy rekord
Reg_Name = „Reg 1”
Address_Reg = „R/W”
Value_Reg = „x0000”
Description seq = „xFF”
```

Vis. Są to funkcje łatwe do zastosowania. Dodanie do plików nowych parametrów nie powinno sprawić problemów.

Roźmieszczenie elementów systemu.

Projekt wykorzystuje niemal każdy dostępny pory, część wyprowadzeń zachowało swoje pierwotne funkcje, natomiast pozostałe związane są z funkcjami przyrządu.

Port MPX A

Port A pozostaje prawie w nienaruszonym stanie (tabela 1). Ze względu na ograniczone zasoby FPGA usunąłem z niego dwa generatory PWM oraz wejścia analogowe. Pozostałe funkcje pozostały bez zmian. Można z nich korzystać wykorzystując funkcje zawarte w palcie myRIO. Najlepiej korzystać z funkcji niskiego poziomu.

Port MPX B

Funkcje wyprowadzeń portu B zostały przypisane do analizatora stanów logicznych (tabela 2). Każda z linii DIO0...DIO15 pracuje w roli wejścia analizatora i może być źródłem wyzolenia pomiaru. W porcie pozostał jeszcze interfejs UART i wyjścia dwóch przetworników C/A. Dostęp do nich zapewniają funkcje palety myRIO.

Port MSP C

Wyprowadzenia tego portu są najlepiej dostosowane do sygnałów analogowych, dlatego przypisana im rola związana jest z funkcjami oscyloskopu i generatora funkcyjnego (tabela 3). Dostępne sygnały cyfrowe są tylko wyjściami i z założenia mogą sterować dodatkowymi układami kondycjonowania np. wzmacniaczami wejściowymi. Wejścia AIN0+, AIN0-, AIN1+, AIN1- pełnią rolę różnicowych wejść oscyloskopu. Wyjścia AO0, i AO1 bipolarnych wyjść generatorów.

W tabeli 4 przedstawiono przyporządkowanie wyprowadzeń do sterowania układem dodatkowych wzmacniaczy wejściowych oscyloskopu.

Podsumowanie

Wykonana aplikacja pokazuje uniwersalność i możliwości, które daje myRIO, Platforma jest przeznaczona głównie do celów edukacyjnych i dlatego zwykle miejscem jej użycia są pracownie uczelniane lub laboratoria studenckie. Korzystając z pokazanej aplikacji studenci mogą zapewnić sobie niezbędne narzędzia potrzebne przy pierwszych próbach konstruowania własnych urządzeń.

Tabela 1. Roźmieszczenie sygnałów w porcie MPX A

+3,3 V	DIO10	DIO9	DIO8 / PWM0	DIO7 / SPI.MOSI	DIO6 / SPI.MISO	DIO5 / SPI.CLK	DIO4	DIO3	DIO2	DIO1	DIO0	A13 – Nie wyk.	A12 – Nie wyk.	A11 – Nie wyk.	A10 – Nie wyk.	+5 V
33	31	29	27	25	23	21	19	17	15	13	11	9	7	5	3	1
34	32	30	28	26	24	22	20	18	16	14	12	10	8	6	4	2
DIO15 / I2C.SDA	DIO14 / I2C.SCL	DGND	DGND	DIO13	DGND	DIO12 / ENC.B	DGND	DIO11 / ENCA	DGND	UART.TX	DGND	UART.RX	DGND	AGND	AO1	AO0

Tabela 2. Roźmieszczenie sygnałów w porcie MPX B

+3,3 V	DIO10	DIO9	DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0	A13 – Nie wyk.	A12 – Nie wyk.	A11 – Nie wyk.	A10 – Nie wyk.	+5 V
33	31	29	27	25	23	21	19	17	15	13	11	9	7	5	3	1
34	32	30	28	26	24	22	20	18	16	14	12	10	8	6	4	2
DIO15	DIO14	DGND	DGND	DIO13	DGND	DIO12	DGND	DIO11	DGND	UART.TX	DGND	UART.RX	DGND	AGND	AO1	AO0

Tabela 3. Roźmieszczenie sygnałów w porcie MSP C

+15V	-15V	AGND	AO0 (F.G.CH0)	AO1 (F.G.CH1)	AGND	AIO+ (+OSC.CH1)	AIO- (-OSC.CH1)	A11+ (+OSC.CH2)	A11- (-OSC.CH2)	DIO0 (K0.CH1)	DIO1 (K1.CH1)	DIO2 (K2.CH1)	DIO3 (ON/OFF.CH1)	DIO4 (K0.CH2)	DIO5 (K1.CH2)	DIO6 (K2.CH2)	DIO7 (ON/OFF.CH2)	DGND	5V
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Tabela 4. Sterowanie dodatkową płytka wzmacniaczy

Wyprowadzenie	Funkcja	Wartości
DIO0...2	Sterowanie wzmocnieniem kanału 1	(K2,K1,K0) 000 – 100 mV/dz 001 – 200 mV/dz 010 – 500 mV/dz 011 – 1 V/dz 100 – 2 V/dz 101 – 5 V/dz 110 – 10 V/dz 111 – 20 V/dz
DIO 3	Włączenie/wyłączenie kanału 1	0 – wyłączony 1 – włączony
DIO4...6	Sterowanie wzmocnieniem kanału 2	Jak dla kanału #1
DIO7	Włączenie/wyłączenie kanału 2	0 – wyłączony 1 – włączony

Sterowanie zawartościami rejestrów układów scalonych wyposażonych w SPI lub I²C umożliwia zapoznanie się z ich funkcjonalnością. W następnej części opiszę sposób skonfigurowania struktury FPGA oraz

ważniejsze fragmenty kodu, aby ułatwić modyfikowanie i rozbudowywanie oprogramowania ma własne potrzeby.

Wiesław Szaj
wszaj@prz.edu.pl

<http://ep.com.pl>

IDEALNY PREZENT

D L A F O T O A M A T O R A

Chcesz zrobić efektowny i niebanalny prezent – sobie lub komuś z Twoich Bliskich?

Interesujesz się fotografią lub chcesz kogoś zarazić tą pasją?

Idealny prezent to roczna prenumerata największego polskiego magazynu fotograficznego – miesięcznika **Digital Camera Polska** – rozpoczęta gratisowym egzemplarzem unikatowego albumu najświetniejszych zdjęć amerykańskiego magazynu **LIFE** – **The Classic Collection**.

Ten bogaty prezent mamy dla naszych Czytelników szokująco tanio: wystarczy, że opłacisz roczną prenumeratę Digital Camera Polska od lipca 2015 do czerwca 2016 (ze zniżką 16% kosztuje ona 199 zł), a natychmiast wyślemy Ci album The Classic Collection. **Gratis!**

~~358,80 zł~~

199,00 zł

To naprawdę rzadka okazja, bo dwanaście numerów Digital Camera Polska plus album kosztowałyby łącznie 12 x 19,90 zł + 120,00 zł = 358,80 zł!
Zakładając, że znalazłbyś ten album w jakimś antykwariacie...

Do wyczerpania zapasów ten wyjątkowy prezent dostępny jest na stronie www.digitalcamerapolska.pl/prezent. Można też zamówić go mailem (prenumerata@avt.pl) lub telefonicznie (22 257 84 22), lub po prostu wpłacić 199 zł na konto AVT-Korporacja sp. z o.o., 03-197 Warszawa, ul. Leszczyńska 11, BNP Paribas Bank Polska SA, 97 1600 1068 0003 0103 0305 5153, koniecznie z dopiskiem „Prenumerata + LIFE” w tytule wpłaty.

