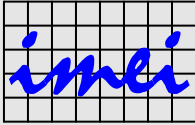


<p style="text-align: center;">CYFROWE PRZETWARZANIE SYGNAŁÓW Laboratorium Inżynieria Biomedyczna, studia stacjonarne pierwszego stopnia</p>	 <p style="text-align: center;">Instytut Metrologii, Elektroniki i Informatyki</p>
<p style="text-align: center;">Temat: Import próbek sygnałów z plików binarnych</p>	<p style="text-align: center;">Ćwiczenie 0</p>

1. Cel ćwiczenia

Celem ćwiczenia jest napisanie w środowisku LabWindows/CVI firmy National Instruments programu umożliwiającego odczyt wartości próbek zapisanych w pliku binarnym. Stworzony program będzie punktem wyjścia dla programów tworzonych na kolejnych ćwiczeniach laboratoryjnych z niniejszego przedmiotu.

2. Zagadnienia do przygotowania

- Funkcje języka ANSI C dotyczące: dostępu do plików (`fopen`, `fclose`), operacji na plikach (głównie odczyt danych; m.in. funkcja `fread`), dynamicznego przydzielania i zwalniania pamięci (`malloc`, `realloc`, `calloc`, `free`), podnoszenia do potęgi liczby 2.
- Składnia instrukcji warunkowej `if` oraz `switch-case` w języku ANSI C.
- Deklaracja zmiennych w języku ANSI C.
- Odnoszenie się do komórek tablicy w języku ANSI C.
- Przetwarzanie analogowo-cyfrowe (działania wykonywane w procesie próbkowanie, kwantowanie i kodowanie).
- Podstawowe parametry przetworników analogowo-cyfrowych (A/C): rozdzielczość, zakres i polaryzacja napięć wejściowych.
- Przeliczanie kodów uzyskanych na wyjściu przetwornika A/C na odpowiadające im wartości napięć przy znajomości: rozdzielczości B przetwornika, zakresu U_{FS} dopuszczalnych napięć wejściowych i typu przetwornika (unipolarny, bipolarny). Zagadnienie to zostało opisane w punkcie 4.3 niniejszej instrukcji.
- Format pliku CPS. Zagadnienie to zostało opisane w punkcie 4.2 niniejszej instrukcji.
- Prezentacja wartości liczbowych na elemencie Numeric w środowisku LabWindows/CVI. Zagadnienie to zostało opisane w punkcie 4.4 niniejszej instrukcji.
- Rysowanie wykresów na elemencie Graph i ich usuwanie w środowisku LabWindows/CVI. Zagadnienie to zostało opisane w punkcie 4.5 niniejszej instrukcji.

3. Program ćwiczenia

Zadanie 1

- Na podstawie informacji przygotowanych w oparciu o zalecenia zawarte w punkcie 2 oraz opis formatu pliku CPS (zamieszczonego w punkcie 4), napisz program, który umożliwi wczytanie z pliku binarnego i prezentację na elementach typu Numeric z wykorzystaniem funkcji `SetCtrlVal` następujących informacji:
 - zakresu U_{FS} napięć wejściowych przetwornika wyrażonego w woltach,
 - rozdzielczości B przetwornika A/C,
 - typ przetwornika,
 - zastosowanej częstotliwości próbkowania przy pobieraniu próbek wyrażonej w hercach,
 - liczby kodów próbek zapisanych w pliku.
- Przetestuj napisany program wczytując dane z plików wskazanych przez prowadzącego zajęcia. W tabeli 1 przedstawiono wyniki, jakie powinny zostać uzyskane dla każdego z plików testowych.

Tabela 1. Wyniki, jakie powinny zostać uzyskane w tworzonej aplikacji po wczytaniu plików testowych

Plik	test_uni.cps	test_bi.cps
Zakres napięć wejściowych przetwornika A/C [V]	5	5
Rozdzielczość przetwornika A/C [bity]	16	16
Typ przetwornika A/C	0 – unipolarny	1 – bipolarny
Zastosowana podczas pobierania próbek częstotliwość próbkowania [Hz]	5000	5000
Liczba próbek	100	100
Kształt sygnału	sinusoida wyprostowana dwupółkowo o amplitudzie 4,5 V	sinusoida o amplitudzie 4,5 V

- c) W przypadku niezgodności otrzymanych wyników z wynikami zaprezentowanymi w tabeli 1 dokonaj modyfikacji kodu programu. Poprawnie działający program zaprezentuj prowadzącemu zajęcia.

Zadanie 2

- Rozwiń tworzoną aplikację o wczytywanie kodów próbek, ich przeliczenie na jednostki napięcia oraz graficzną prezentację na elemencie *Graph*. W celu realizacji graficznej prezentacji próbek sygnału, na podstawie wartości częstotliwości próbkowania odczytanej z pliku wyznacz wartości chwil czasowych, w których zostały pobrane próbki i przedstaw wykres sygnału z osią odciętych skalowaną w jednostkach czasu. Do graficznej prezentacji wykresu zastosuj funkcję *PlotXY*.
- Przetestuj napisany program ponownie wczytując dane z plików wskazanych przez prowadzącego zajęcia. W ostatnim wierszu tabeli 1 przedstawiono informację o kształtach sygnałów, których próbki zawarte są w plikach testowych.
- W przypadku niezgodności otrzymanych wyników z wynikami zaprezentowanymi w tabeli 1 dokonaj modyfikacji kodu programu. Poprawnie działający program zaprezentuj prowadzącemu zajęcia.

4. Wskazówki do ćwiczenia

4.1. Wymagania ogólne

Podczas tworzenia programu należy:

- panel użytkownika oznaczyć imionami i nazwiskami autorów oraz numerem ćwiczenia,
- komentować na bieżąco kod całej tworzonej aplikacji w pliku źródłowym (komentarze powinny podawać przydatne informacje ułatwiające zrozumienie programu i wprowadzanie ewentualnych zmian),
- stosować poprawny styl kodowania charakteryzujący się:
 - wcięciami;
 - parowaniem nawiasów klamrowych { } grupujących określoną część programu w tej samej kolumnie;
 - nadawaniem zmiennym nazw, które jednoznacznie określają informację przechowywaną w tej zmiennej;
 - jednoznacznym nazewnictwem wszystkich elementów na panelu użytkownika;
- dokonać implementacji wszystkich założeń dla programu,
- zachować estetyczny i przejrzysty wygląd interfejsu użytkownika aplikacji.

4.2. Format pliku CPS

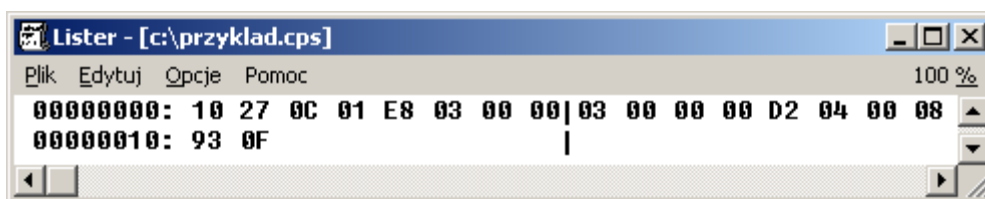
Podczas wszystkich ćwiczeń laboratoryjnych próbki analizowanych sygnałów udostępniane są w postaci plików binarnych. W każdym pliku oprócz kodów reprezentujących skwantowane wartości próbek sygnału pobrane w procesie przetwarzania analogowo-cyfrowego, znajdują się również podstawowe parametry zastosowanego przetwornika A/C. W tabeli 2 przedstawiono znaczenie poszczególnych pól zastosowanego formatu pliku.

Wartości ze znakiem kodowane są w kodzie uzupełnienia do 2. Kolejność bajtów dla informacji wielobajtowych odpowiada zapisowi w formacie LittleEndian, tzn. bajty są ustawione w kolejności od najmłodszego do najstarszego.

Tabela 2. Format pliku CPS

Rozmiar informacji w bajtach	Znaczenie	Typ danej	Dodatkowe informacje
2	Zakres U_{FS} napięć wejściowych przetwornika A/C	unsigned short int	Podane w [mV]
1	Rozdzielczość przetwornika	unsigned char	Podane w bitach
1	Typ przetwornika	unsigned char	0 – unipolarny, 1 – bipolarny
4	Zastosowana podczas pobierania próbek częstotliwość próbkowania	unsigned int	Podane w [Hz]
4	Liczba próbek	unsigned int	
2	Kod wartości próbki pierwszej	unipolarny – unsigned short int bipolarny – short int	
2	Kod wartości próbki drugiej	unipolarny – unsigned short int bipolarny – short int	
...	
2	Kod wartości próbki przedostatniej	unipolarny – unsigned short int bipolarny – short int	
2	Kod wartości próbki ostatniej	unipolarny – unsigned short int bipolarny – short int	

Na rys. 1 oraz w tabeli 3 przedstawiono przykład rozmieszczenia kolejnych bajtów informacji dla przypadku zapisu trzech kodów wartości próbek pobranych z częstotliwością próbkowania 1kHz przy wykorzystaniu bipolarnego przetwornika A/C o zakresie napięć wejściowych $U_{FS} = 10\text{ V}$ i 12-bitowej rozdzielczości.



Rys. 1. Podgląd przykładowego pliku CPS w formacie heksadecymalnym

Tabela 3. Znaczenie poszczególnych bajtów w przykładowym pliku z rysunku 1

Nr bajta	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
Wartość hex	10	27	0C	01	E8	03	00	00	03	00	00	00	D2	04	00	08	93	0F
Informacja	10000mV		12	bi	1000Hz			3 próbki			1234 _d	2048 _d	3987 _d					

W tworzonym programie odczyt każdego pola informacji można przeprowadzić z wykorzystaniem funkcji `fread`. Aby właściwie odczytać dane z pliku należy:

- stosować typy zmiennych, do których zapisywane są dane, zgodnie ze specyfikacją zawartą w tabeli 2;
- podawać właściwy rozmiar danych w parametrach funkcji `fread` (zgodnie z tabelą 2).

Próbki sygnałów testowych pozwalających na sprawdzenie poprawności implementacji odczytu informacji znajdują się w plikach `TEST_UNI.CPS` oraz `TEST_BI.CPS`.

4.3. Przeliczanie kodów próbek na wartości napięcia

Dysponując kodami próbek odczytanymi z pliku w formacie CPS można dokonać ich przeliczenia na odpowiadające im wartości napięć według następującej zależności

$$u(n) = k(n) \cdot q \quad (1)$$

gdzie n – numer próbki, $u(n)$ – obliczona wartość napięcia, $k(n)$ – kod próbki odczytany z pliku, q – krok kwantowania wyznaczony dla przetwornika unipolarnego według zależności

$$q = \frac{U_{FS}}{2^B} \quad (2)$$

natomiast dla przetwornika bipolarnego według zależności

$$q = \frac{U_{FS}}{2^{B-1}} \quad (3)$$

gdzie B – liczba bitów zastosowanego przetwornika A/C, U_{FS} – zakres napięciowy przetwornika A/C.

W celu uzyskania na wykresie sygnału osi odciętych skalowanej w jednostkach czasu, należy wyznaczyć tablicę wartości chwil czasowych przypadających w momentach próbkowania

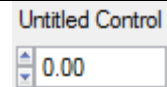
$$x(n) = n \cdot T_p = \frac{n}{f_p} \quad (4)$$

gdzie n – numer próbki, dla której przypada obliczana chwila czasowa, T_p – okres próbkowania, f_p – zastosowana częstotliwość próbkowania.

4.4. Prezentacja wartości liczbowych na elemencie Numeric

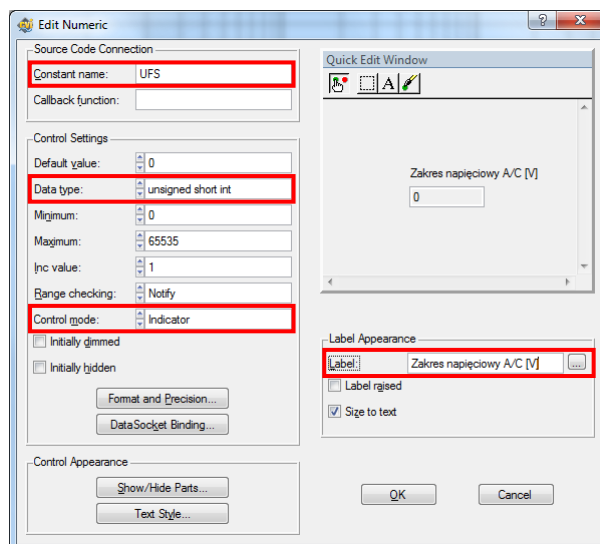
Tworzony program powinien umożliwiać prezentację na panelu użytkownika w postaci numerycznej wartości zapisanych w pierwszych pięciu polach informacyjnych, jakie można wyróżnić w formacie pliku CPS. W tym celu na panelu użytkownika należy umieścić elementy typu *Numeric*. Operację tą wykonujemy przechodząc do edytora interfejsu użytkownika i z menu *Create | Numeric* wybierając element *Numeric* oznaczony ikoną pokazaną w tabeli 3. Z pliku odczytywanych jest 5 różnych parametrów, dlatego na panelu użytkownika należy umieścić 5 elementów tego typu.

Tabela 3. Położenie elementów wymaganych na interfejsie użytkownika tworzonej aplikacji

Element	Ikona elementu	Położenie w menu
Numeric		Create Numeric Numeric

W celu dostosowania właściwości tych elementów do potrzeb tworzonej aplikacji, należy wejść po kolei do ustawień każdego z nich poprzez dwukrotne kliknięcie na wybrany element. Poniżej przedstawiono realizację zmian dla elementu *Numeric*, którego zadaniem będzie prezentacja informacji o zakresie napięcia wejściowego przetwornika analogowo-cyfrowego (pierwsze pole informacyjne w pliku). Zatem po wejściu do ustawień dokonujemy zmian w następujących polach (patrz rysunku 2):

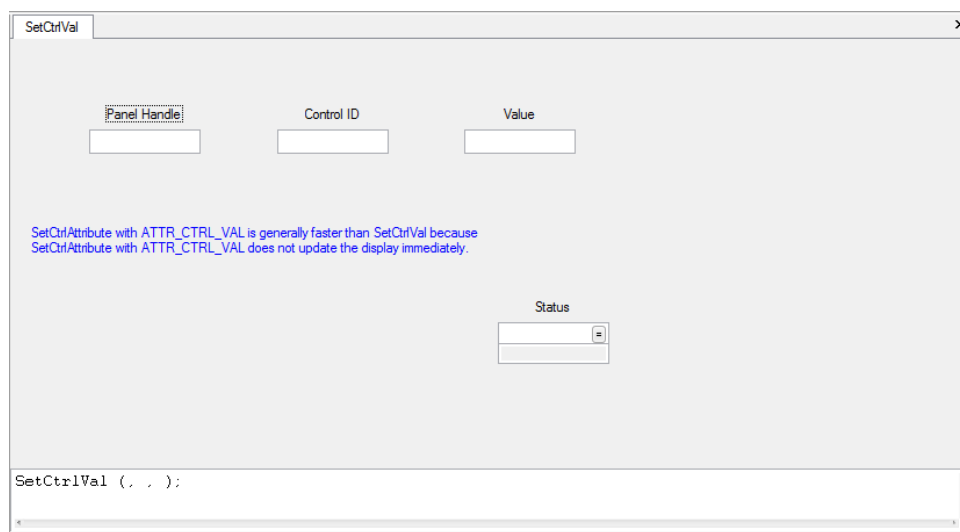
- w polu *Constant Name* wpisujemy `UFS`;
- pole *Data Type* ustawiamy na `unsigned short int` (zmiana w tym polu jest konieczna w celu dostosowania typu elementu do typu zmiennej, której wartość będzie na nim prezentowana; zakres napięciowy zastosowanego przetwornika jest zapisywany w pliku jako wartość typu `unsigned short int` – patrz tabela 1);
- pole *Control Mode* ustawiamy na `Indicator` (zmiana ta powoduje, że element ten będzie służył na panelu użytkownika jako wskaźnik, zatem użytkownik aplikacji nie będzie miał możliwości zmiany wartości na nim wyświetlanej; wszelkiego rodzaju zmiany wyświetlanej wartości będą mogły być dokonywane tylko na drodze programowej, niejako z wnętrza aplikacji);
- w polu *Label* wpisujemy: `Zakres napięciowy A/C [V]`.




Rys. 2. Okno zmiany parametrów elementu Numeric

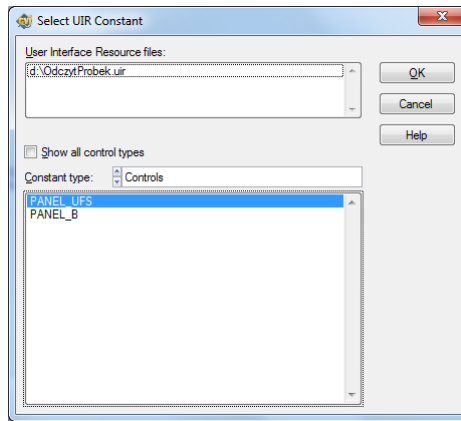
Chcąc programowo dokonać zmiany wartości wyświetlanej na elemencie *Numeric* należy użyć w tym celu funkcji `SetCtrlVal`. Okno pomocne przy uzupełnianiu parametrów wywołania tej funkcji pokazano na rysunku 3. Jego wywołanie uzyskuje się przez zaznaczenie nazwy funkcji w edytorze kodu programu środowiska LabWindows/CVI i wciśnięcie kombinacji klawiszy Ctrl+P.

W polu Panel Handle wprowadzamy nazwę zmiennej, która przechowuje uchwyt do panelu, na którym znajduje się element *Numeric*, do którego będziemy się odwoływali. W większości przypadków programy tworzone na niniejszym laboratorium będą składały się tylko z jednego panelu (okna aplikacji). Zatem jeśli nie dokonywano żadnych zmian w polu Panel Variable Name podczas generacji szkieletu kodu aplikacji, to zmienna ta nosi nazwę `panelHandle`.




Rys. 3. Okno uzupełniania parametrów wywołania funkcji SetCtrlVal

W polu Control ID należy wpisać identyfikator elementu, do którego ma odnosić się dana funkcja. W naszym przypadku jednemu z elementów *Numeric* został przypisany identyfikator o nazwie UFS, jednak w rzeczywistości pełna nazwa identyfikatora elementu tworzona jest z identyfikatora przypisanego panelowi (domyślnie PANEL) oraz identyfikatora przyporządkowanego samemu elementowi. Obydwie te nazwy są rozdzielane znakiem podkreślnika. Zatem w tym przypadku identyfikatorem elementu *Numeric* będzie ciąg znakowy `PANEL_UFS`. W przypadku, gdy nie jesteśmy pewni co do przypisanej elementowi nazwy identyfikatora, należy ustawić kursor w polu Control ID i wcisnąć klawisz  znajdujący się w pasku narzędzi okna (lub klawisz Enter), co powoduje otwarcie okna pokazanego na rysunku 4. Pozwala ono na wybór i wprowadzenie właściwej, a co najważniejsze poprawnej, nazwy identyfikującej element.

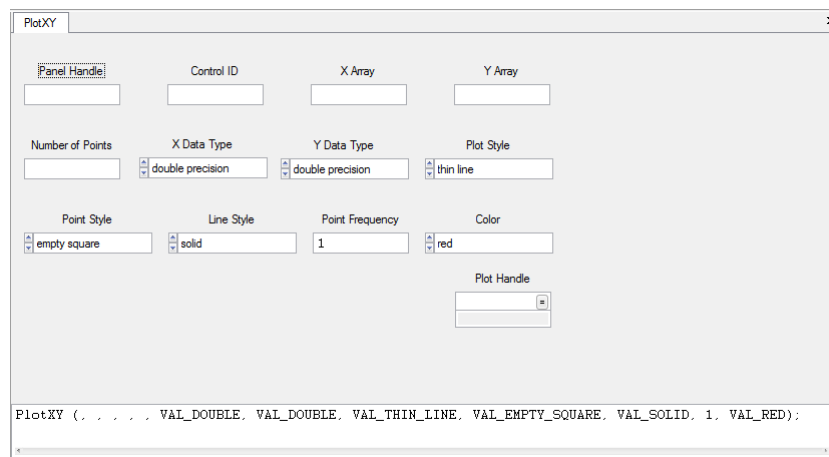


Rys. 4. Okno z nazwami identyfikatorów elementów użytych w tworzonej aplikacji

Na koniec w polu Value należy wprowadzić nazwę zmiennej, która przechowuje wartość, jaką chcemy wyświetlić w danym elemencie *Numeric*. Wprowadzenie uzupełnionego wywołania funkcji do kodu aplikacji dokonujemy klawiszem .

4.5. Rysowanie i usuwanie wykresów na elemencie Graph

Po dokonaniu odpowiednich przeliczeń odnośnie odczytanych kodów próbek na jednostki napięcia oraz określeniu chwil czasowych, w których próbki zostały pobrane, można wyświetlić przebieg sygnału na elemencie *Graph*. Do tego celu można wykorzystać funkcję **PlotXY**. W ogólności funkcja ta pozwala na rysowanie wykresów na elemencie *Graph*, przy czym wartości zapisane w komórkach tablic podanych w polach X Array i Y Array (patrz rysunek 5) tworzą pary współrzędnych kolejnych punktów wykresu. Elementem łączącym poszczególne wartości z tablic w pary jest ten sam numer komórki. Okno pomocne przy uzupełnianiu parametrów wywołania funkcji przedstawiono na rysunku 5.




Rys. 5. Okno uzupełniania parametrów funkcji PlotXY

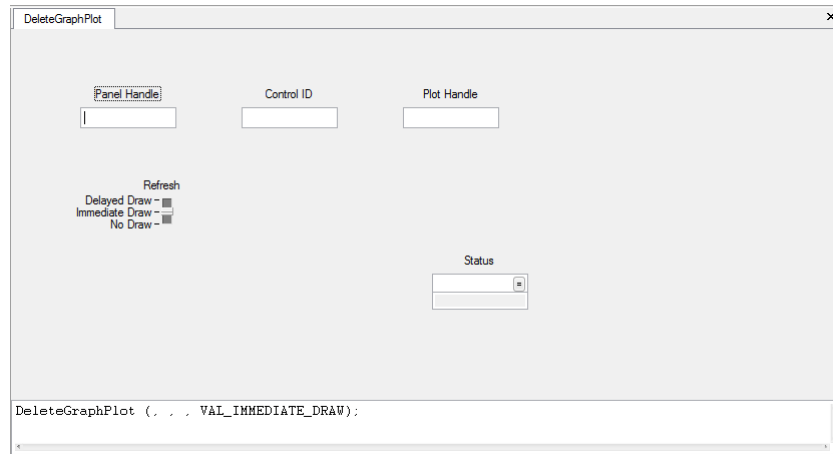
W otwartym oknie uzupełniamy lub dokonujemy zmiany tylko pewnych wybranych parametrów. Uzupełnienie pierwszych dwóch pól przebiega zgodnie z zasadami zaprezentowanymi przy omawianiu funkcji **SetCtrlVal**. Kolejne pola należy uzupełnić według poniższych wskazówek:

- pole X Array – wpisujemy nazwę tablicy, do której zostały zapisane wyznaczone wartości chwil czasowych, w których zostały pobrane próbki;
- pole Y Array – wpisujemy nazwę tablicy, do której zostały zapisane wartości próbek po przeliczeniu na jednostki napięcia;
- pole Number of Points – wpisujemy liczbę próbek, jaka ma być pobrana ze wspomnianych wcześniej tablic i wyświetlona na elemencie *Graph* (oczywiście nie może być ona większa niż rozmiar tablic, których nazwy podano we wcześniejszych polach);
- pole X Data Type – określamy typ zmiennej, jakie przechowuje każda komórka tablicy, której nazwa została wprowadzona w polu X Array;
- pole Y Data Type – określamy typ zmiennej, jakie przechowuje każda komórka tablicy, której nazwa została wprowadzona w polu Y Array;

- pole Plot Style – ustawiamy w nim base zero vertical bar, jeśli chcemy, aby każda próbka sygnału była prezentowana w postaci pionowego prążka.

W ten sposób uzupełnioną funkcję można wprowadzić do pliku z kodem tworzonej aplikacji przez wciśnięcie wspomnianego już wcześniej klawisza  znajdującego się w pasku narzędzi u góry okna.

Jeśli na elemencie *Graph* znajduje się jakiś wykres, to jego skasowanie może być zrealizowane z wykorzystaniem funkcji **DeleteGraphPlot**. Okno pomocne przy uzupełnianiu parametrów wywołania funkcji przedstawiono na rysunku 6.



Rys. 6. Okno uzupełniania parametrów funkcji `DeleteGraphPlot`

W polu Panel Handle wprowadzamy nazwę zmiennej, która przechowuje uchwyt do panelu, na którym znajduje się element *Graph*. Jak wspomniano przy omawianiu funkcji **SetCtrlVal**, jeśli nie dokonano żadnych zmian podczas tworzenia szkieletu kodu aplikacji, będzie to zmienna o nazwie `panelHandle`. W polu Control ID należy wpisać identyfikator element, do którego ma odnosić się dana funkcja. W naszym przypadku będzie to ciąg znaków `PANEL_SYGNAL`. Ostatni parametr opisany nazwą Plot Handle określa uchwyt do wykresu na elemencie *Graph*. Jeśli chcemy skasować wszystkie wykresy, to w polu tym należy wpisać wartość `-1`.

5. Literatura

- [1] Kernighan B.W., Ritchie D.M.: „*Język ANSI C*”. WNT, Warszawa 2002.
- [2] Prata S.: „*Język C. Szkoła programowania*”. Helion, Gliwice 2006.
- [3] Kulka Z.: „*Przetworniki analogowo-cyfrowe i cyfrowo-analogowe*”. WKŁ, Warszawa 1987.
- [4] van de Plassche R.: „*Scalone przetworniki analogowo-cyfrowe i cyfrowo-analogowe*”. WKŁ, Warszawa 1997.
- [5] Materiały firmy National Instruments opisujące środowisko LabWindows/CVI umieszczone na stronie internetowej związanej z niniejszym przedmiotem.