

Ćw. 2. Wprowadzenie do graficznego programowania przyrządów pomiarowych

Problemy teoretyczne:

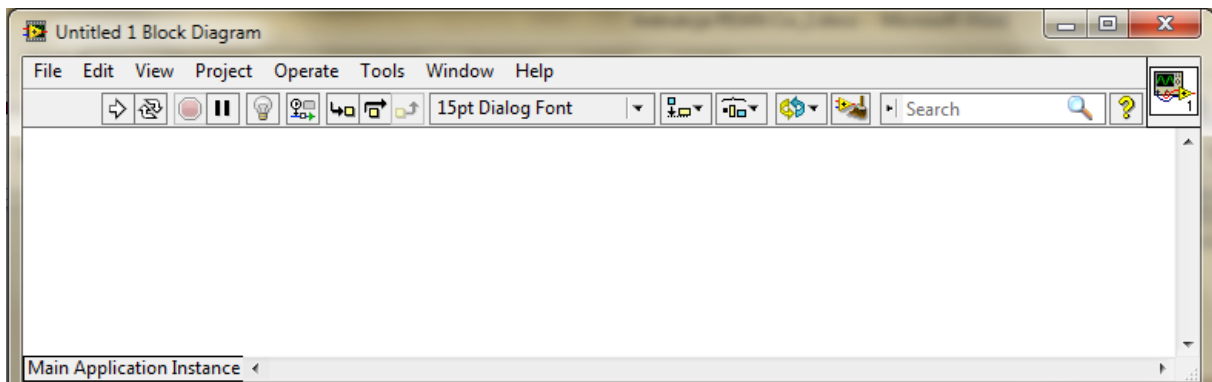
- Prezentacja podstaw budowy komputerowych systemów pomiarowych (dopasowanie wymogów sprzętowych). Prezentacja środowiska programistycznego LabVIEW i jego aplikacji. Budowa prostych aplikacji umożliwiających poznanie możliwości interfejsu programistycznego środowiska.
- Obsługa interfejsów komunikacyjnych w środowisku LabVIEW. Budowa prostych przyrządów wirtualnych, jako składnika komputerowych systemów pomiarowych.

Program ćwiczenia:

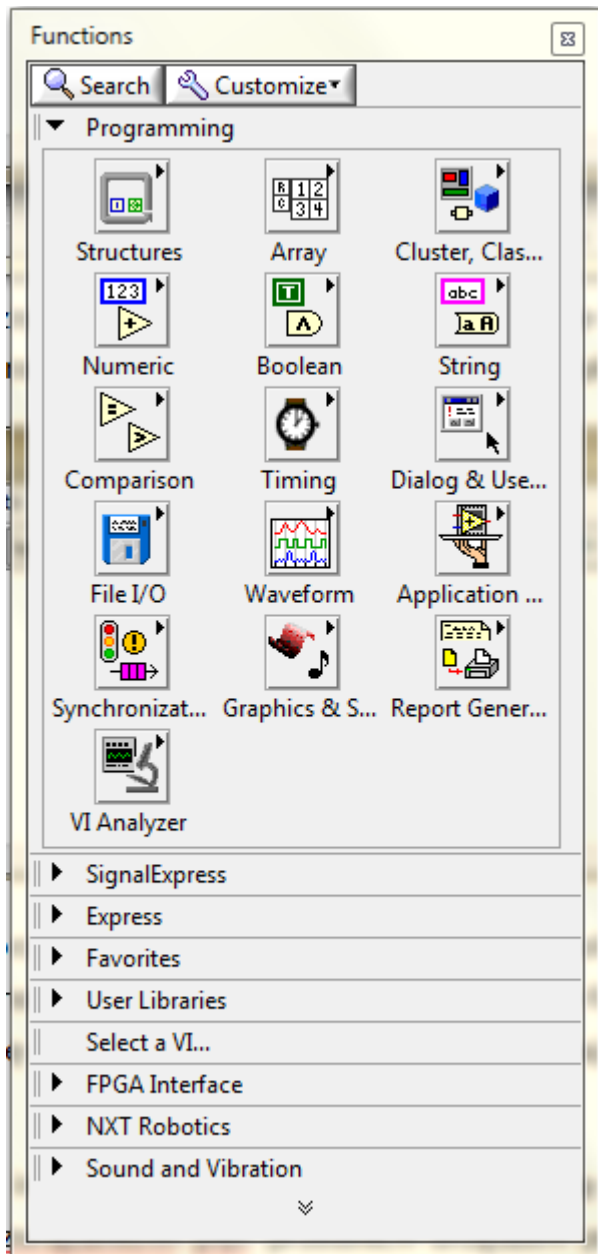
1. Uruchomić komputer PC – system operacyjny WINDOWS XP, WINDOWS 7 – oraz zalogować się w systemie jako użytkownik *student/zmisp* (bez hasła).
2. W folderze **C://PEWN/** założyć unikalny folder dla studenckiej grupy laboratoryjnej (tylko w tym folderze można dokonywać zapisów i modyfikacji własnych plików).
3. Po wysłuchaniu wprowadzenia teoretycznego z godnie z poleceniami prowadzącego wykonać zadane ćwiczenie w środowisku LabVIEW.

Okna w programie LabVIEW

Block diagram- jest graficznym zapisem kodu programu. Definiuje funkcjonalność aplikacji w języku graficznym G. Diagram jest zapisem kodu programu w języku graficznym G.

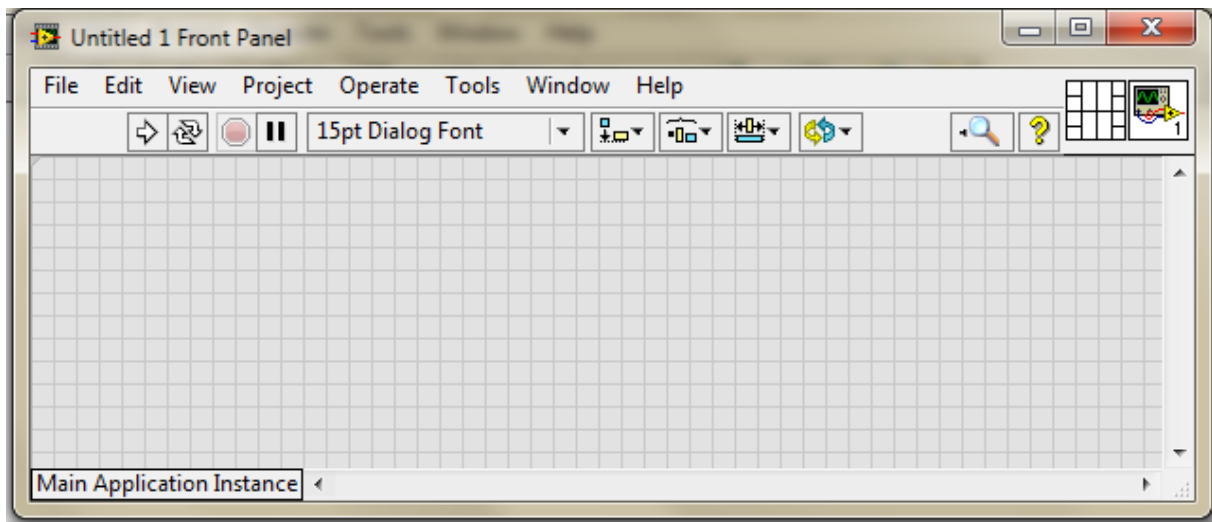


Do zapisu programu wykorzystuje się końcówki (terminals controls indicators), węzły (nodes), przewody (wires) oraz konstrukcje sterujące (structures) a także funkcje. Po naciśnięciu prawego klawisza myszy pojawia się **okno Functions Palette**, w którym znajdują się wszystkie dostępne w programie funkcje.



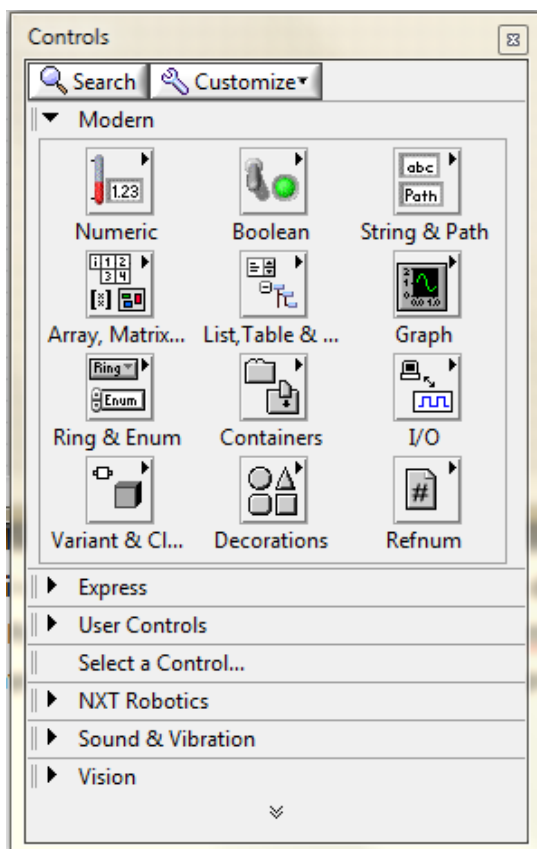
Function Palette

Front panel - pełni rolę interfejsu użytkownika. Jest graficzną formą interfejsu użytkownika. Wizualnie odpowiada rozwiązaniom płyt przednich urządzeń pomiarowych. Buduje się go za pomocą elementów kontrolnych i prezentacyjnych, które stanowią odpowiednio końcówki wejściowe i wyjściowe programu VI. Elementami kontrolnymi są pokrętła nastawcze, przyciski, przełączniki itp.



Front Panel

Symulują one elementy nastawcze fizycznych urządzeń i dostarczają danych wejściowych do diagramu VI. Do elementów prezentacyjnych zalicza się wyświetlacze numeryczne, alfanumeryczne, LED-y, wyświetlacze graficzne (wykresy), tabele itp. Symulują one elementy prezentacyjne fizycznych urządzeń i wyświetlają dane dostarczane przez diagram. Po naciśnięciu prawego klawisza myszy na Front Panelu pojawia się **okno Controls Palette**, w której znajdują się Controlki (do wprowadzania danych) oraz Indicators i Graphy (do wyprowadzania danych).







Controls Palette

Pomiary Elektryczne Wielkości Nielektrycznych – Ćw. 2












Przełączanie pomiędzy oknami Front Panel I Block Diagram klawiszami CTRL – E.

Elementy paska narzędzi (Toolbar) do uruchamiania programów

-  **Run** - uruchamia program (pojedyncze wykonanie)
-  **Run Continuously** - uruchamia program w trybie ciągłym
-  **Abort Execution** - przerywa wykonywanie programu
-  **Pause** - wstrzymuje (pauzuje) wykonywanie programu

Warto zapoznać się z funkcją automatycznego wyboru narzędzi (**Automatic Tool Selection**). Jest ona niezwykle przydatna i pozwala na szybkie i sprawne budowanie elementów programu. Widok okna dostępny w zakładce **VIEW Tools Palette**.

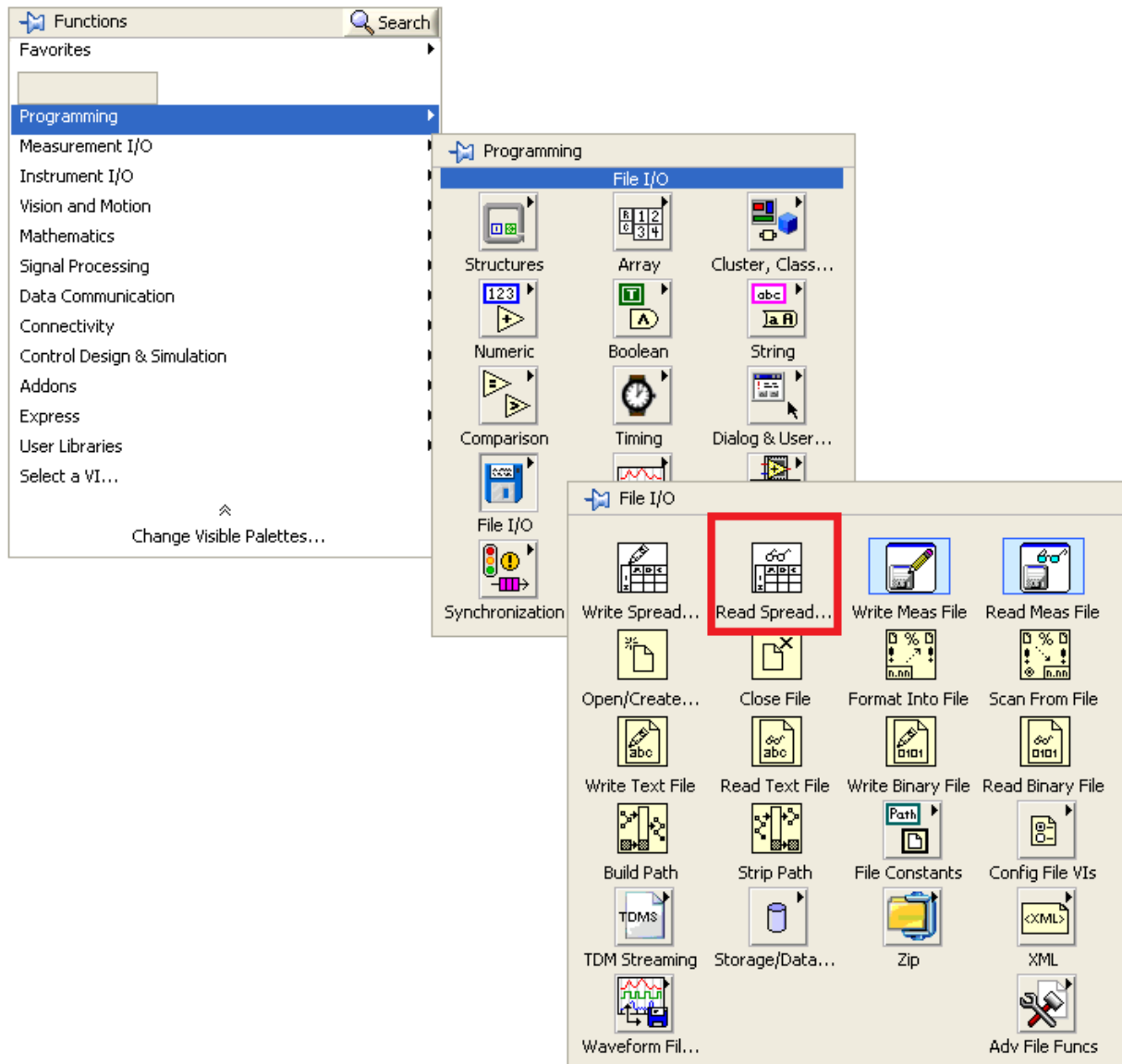
Elementy palety narzędzi (Tools Palette) okno Tools Palette

-  **Automatic Tool Selection** - automatycznie wybiera narzędzie w zależności od wskazania kursora, jeśli dioda jest załączona wybór narzędzi jest automatyczny.
-  **Operate Value** - pozwala na zmianę wartości wybranego elementu
-  **Position/Size/Select** - Wybiera element, zmienia jego rozmiar lub pozycję
-  **Edit Text** - edytuje tekst
-  **Connect wire** - łączy elementy
-  **Object shortcut menu** - tworzy skrót do obiektu
-  **Scroll Window** - przewija okno
-  **Set/Clear Breakpoint** - ustawia / kasuje pułapkę (breakpoint) - służy do wykrywania błędów
-  **Probe data** - pobiera dane
-  **Get Color** - pobiera kolor - zgodnie ze wskazaniem
-  **Set Color** - ustawia kolor

Realizacja programu w środowisku LabVIEW.

1. Tworzenie pierwszego bloku

Pierwszym elementem, jaki pojawi się u nas na schemacie jest blok o nazwie: **Read From Spreadsheet File**. Bloczek znajdziemy klikając prawy klawiszem myszki na puste pole w naszym projekcie na Block Diagramie następnie zakładka **Programming** -> **File I/O** -> : **Read From Spreadsheet File**. Innym sposobem odnalezienia bloczku jest skorzystanie z opcji **Search**, wystarczy kliknąć prawym klawiszem myszki na nasz projekt, wybieramy opcje search i wpisujemy wybrany przez nas blok czyli **Read From Spreadsheet File**.



Wybrany przez nas blok umożliwia nam wczytanie danych do programu z pliku txt. Możemy również wczytać w taki sam sposób dane z programu Microsoft Excel, które jednak wcześniej są wyeksportowane do pliku txt. Ażeby uzyskać więcej informacji na temat wybranego przez nas bloku wystarczy kliknąć opcję help.

Read From Spreadsheet File VI

Owning Palette: [File I/O VIs and Functions](#)

Requires: Base Package

Reads a specified number of lines or rows from a numeric text file beginning at a specified character offset and converts the data to a 2D, double-precision array of numbers, strings, or integers. You must [manually select the polymorphic instance](#) you want to use.

You optionally can transpose the array. The VI opens the file before reading from it and closes it afterwards. You can use this VI to read a spreadsheet file saved in text format. This VI calls the [Spreadsheet String to Array](#) function to convert the data.

Use the pull-down menu to select an instance of this VI.

Select an instance ▾

+ Add to the block diagram Find on the palette

Double

format (%.3f) new file path (Not A Path i...
file path (dialog if empty) all rows
number of rows (all:-1) first row
start of read offset (chars... mark after read (chars.)
max characters/row (no lim... EOF?
transpose (no:F)
delimiter (\t)

format specifies how to convert the numbers to characters. If the **format** is `%.3f` (default), the VI creates a string long enough to contain the number, with three digits to the right of the decimal point. If **format** is `%d`, the VI converts the data to integer form using as many characters as necessary to contain the entire number. If **format** is `%s`, the VI copies the input string. Use the [format string syntax](#).

file path is the path name of the file. If **file path** is empty (default) or is `<Not A Path>`, the VI displays a dialog box from which you can select a file. Error 43 occurs if you cancel the dialog box.

number of rows is the maximum number of rows or lines the VI reads. For this VI, a row is a character string ending with a carriage return, linefeed, or a carriage return followed by a linefeed; a string ending at the end of file; or a string that has the maximum line length specified by the max characters per row input. If **number of rows** is `<0`, the VI reads the entire file. The default is `-1`.

start of read offset is the position in the file, measured in characters (or bytes), at which the VI begins reading. The offset unit is bytes rather than numbers because byte stream files can contain segments of different types of data. Therefore, to read an array of 100 numbers that follows a header of 57 characters, set **start of read offset** to 57.

max characters/row is the maximum number of characters the VI reads before ending the search for the end of a row or line. The default is 0, which means that there is no limit to the number of characters the VI reads.

If **transpose** is TRUE, the VI transposes the data after converting it from a string. The default is FALSE.

delimiter is the character or string of characters to use to separate fields in the spreadsheet file. For example, a value of `,` (comma) specifies a single comma as the delimiter. The default is `\t`, which specifies a single tab character as the delimiter.

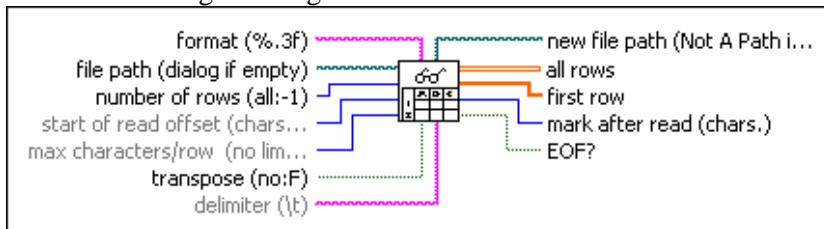
new file path returns the path to the file.

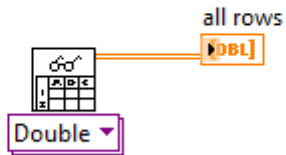
Mamy tutaj wszystkie informacje, jakie dotyczą tego bloku.

Na wyjściu all rows naciśnij prawym klawiszem i wybierz **Create ->Indicator**.

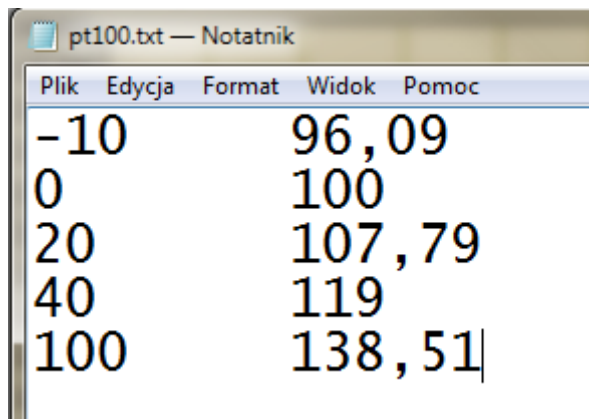
Dane w pliku tekstowym mogą być zapisywane w kolumnach lub w wierszach. Nasz program wymaga danych zapisanych w ciągu, dlatego musimy stworzyć kontrolkę by móc transponować dane z kolumn na wiersze, które nadają się do dalszego przetwarzania.

- Aktywuj okno Block Diagram.
- Wstaw ikonę odczytu. W oknie „Functions” kliknij „Programming”, następnie kliknij „File I/O” (symbol dyskiety), wybierz blok „Read From Spreadsheet File.vi” i umieść go na diagramie.





Tworzymy plik tekstowy z danymi wejściowymi do programu, kolumny rozdzielamy znakami tabulacji.



Proszę pojedynczą strzałką uruchomić program, a następnie wskazać plik z danymi.

Na Front Panelu pojawią się dane w postaci.

The screenshot shows a data table on a front panel titled 'all rows'. The table has 5 columns and 10 rows. The first two columns contain the data from the text file, with values formatted with commas as thousands separators. The last two columns contain zeros.

all rows				
-10,000	96,090	0,000	0,000	
0,000	100,000	0,000	0,000	
20,000	107,790	0,000	0,000	
40,000	119,000	0,000	0,000	
100,000	138,510	0,000	0,000	
0,000	0,000	0,000	0,000	
0,000	0,000	0,000	0,000	
0,000	0,000	0,000	0,000	
0,000	0,000	0,000	0,000	

Można tablicę transponować, w tym celu na wejściu bloku **Read From Spreadsheet File** o nazwie **transpose** stworzyć stałą (**TRUE**) Na wyjściu transpose naciśnij prawym klawiszem i wybierz **Create ->Constans**. Następnie klikając w stałą można zmienić jej wartość z **False** na **True**.

Po ponownym uruchomieniu programu na Front Panelu tablica zostanie odwrócona.

all rows

-10,000	0,000	20,000	40,000
96,090	100,000	107,790	119,000
0,000	0,000	0,000	0,000
0,000	0,000	0,000	0,000
0,000	0,000	0,000	0,000
0,000	0,000	0,000	0,000
0,000	0,000	0,000	0,000
0,000	0,000	0,000	0,000
0,000	0,000	0,000	0,000

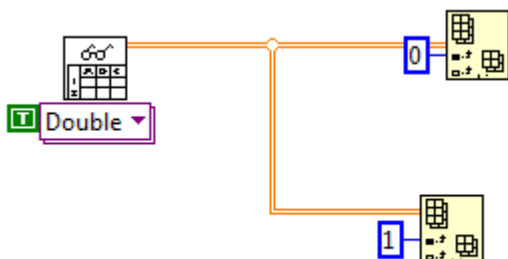
2. Rozdzielenie zmiennym x,y

Kolejnym etapem naszego ćwiczenia jest rozdzielenie zmiennych typu double na zmienne x,y. Pomoże nam w tym bloczek o nazwie **Index Array Function**. . Bloczek znajdziemy klikając prawy klawiszem myszki na puste pole w naszym projekcie następnie zakładka **Programming** -> **Array** -> : **Index Array**. Innym sposobem odnalezienia bloczku jest oczywiście skorzystanie z opcji **Search**. Umieszczamy 2 takie same bloczki, jeden będzie odpowiedzialny za zmienne x, drugie za zmienne y. Następnie wybrane przez nas bloki musimy połączyć. Dokładną informację odnośnie połączenia możemy wyszukać w opcji **Help**.

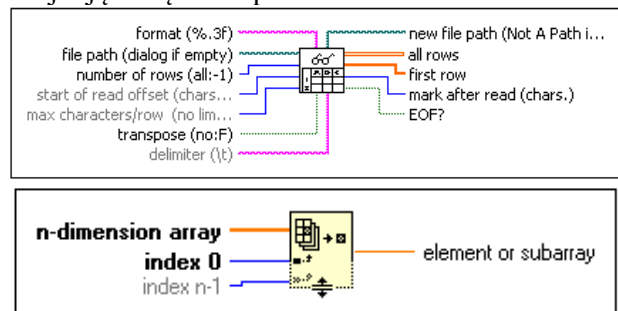
3. Łączenie bloków szpulką.

W bloku pierwszym, czyli **Read From Spreadsheet File VI** podłączamy kabelek do wejścia "all rows", jest to pierwsze wejście z prawej strony. Dzięki temu możemy odczytać wszystkie wiersze z naszymi danymi. Drugą stronę kabelka podpinamy do obu bloków **Index Array Function** na wejście "n-dimension array" również pierwsze wejście jednak z lewej strony. W tym bloku musimy również wybrać indeksowanie zmiennej. Dla jednego bloku wybieramy 0 (dla bloku odpowiedzialnego za zmienna x) dla drugiego 1. Indeksowanie robimy za pomocą bloku **Numeric Constant**. Znajdziemy go w zakładce **Programming** -> **Numeric** -> : **Numeric Constant**. Ażeby nadać konkretną wartość zaznaczamy w okienku liczbę i podajemy przez nas wybrana.

Schemat podłączenia po pierwszych krokach powinien wyglądać tak:

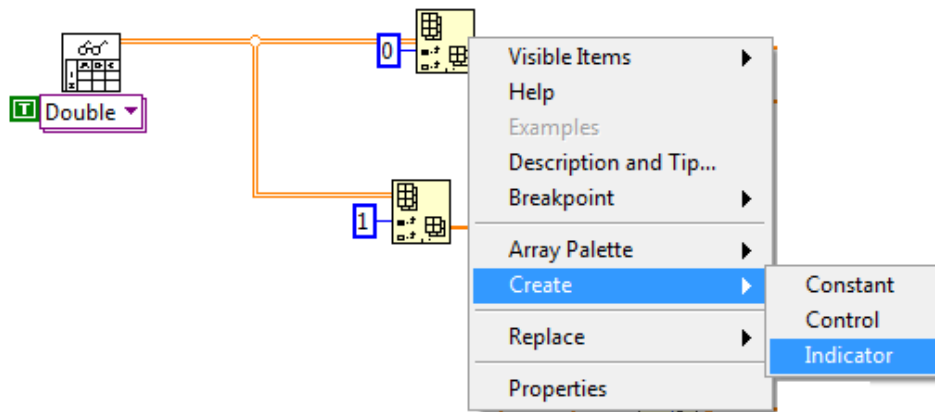


Informację o wejściach i wyjściach bloków znajdujące się w Helpie.

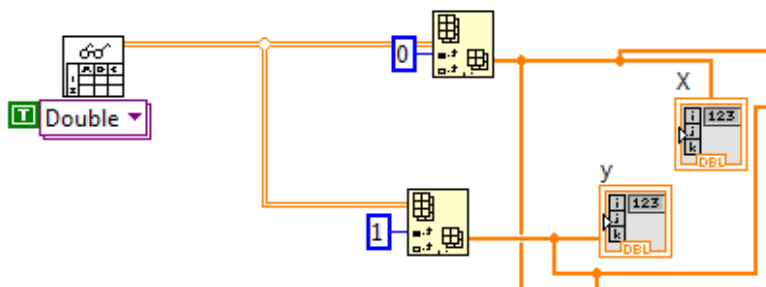


4. Tworzenie bloku Indicator.

Następny etap to blok zajmujący się wypisaniem naszych zmiennych jest on połączony z blokiem **Index Array Function**. Tworzymy go za pomocą kliknięcia prawym klawiszem myszy na blok **Index Array** następnie zakładka **Create** -> **Indicator**.

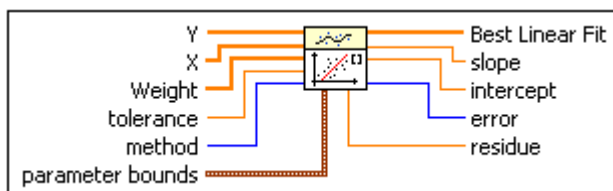


W pole nazwa wpisujemy nazwę naszej zmiennej. Czyli dla bloku o indeksie 0 - x, a dla 1-y.



5. Tworzenie bloku Linear Fit.

Następnie tworzymy blok który zlinearyzuje nam naszą charakterystykę. Znajdziemy go w zakładkach **Mathematics** -> **Fitting** -> : **Linear Fit**. Bardzo ważne jest tutaj prawidłowe podpięcie. Jeżeli zrobimy to nieprawidłowo charakterystyka będzie odwrócona. Schemat podpięcia wygląda następująco:



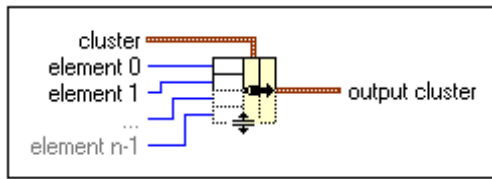
Do wejścia Y podłączamy kabelek od zmiennych y, do wejścia X podłączamy kabelek od zmiennych x.

6. Wykreślenie charakterystyki.

6.1. Blok Bundle.

Pomiary Elektryczne Wielkości Nielektrycznych – Ćw. 2

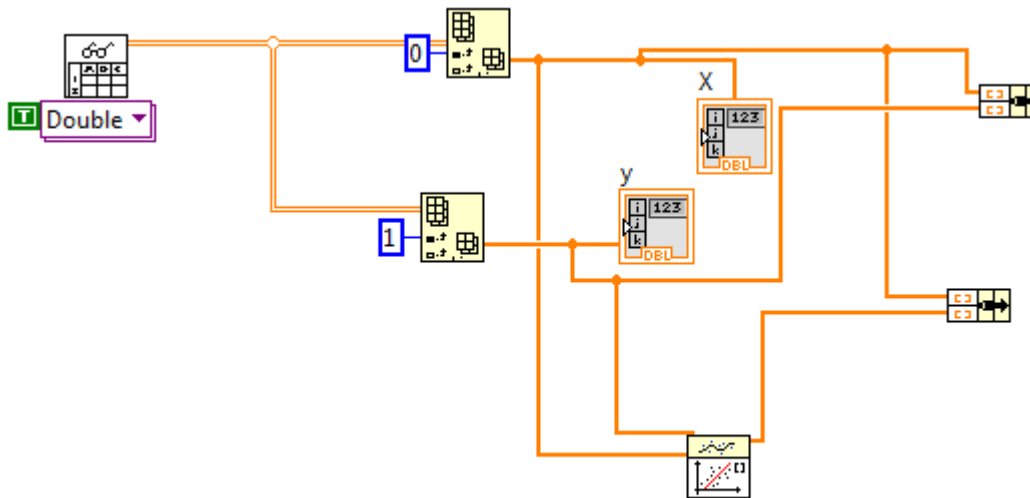
1 etap to stworzenie 2 bloków **Bundle**. Będzie on odpowiedzialny za połączenie zmiennych x, y . Bloczek możemy znaleźć w zakładce **Programming** -> **Cluster, Class & Variant** -> : **Bundle**.



W **pierwszym** blozku do wejścia element **0** - podpinamy zmienną x , do wejścia element **1** podpinamy zmienną y .

W **drugim** blozku do wejścia element **0** - podpinamy zmienną x , zaś do wejścia element **1** podpinamy zlinearyzowane nasze dane czyli z nóżki **Best Linear Fit** z bloku : **Linear Fit** prowadzimy kabel.

Schemat nasz powinien wyglądać następująco:



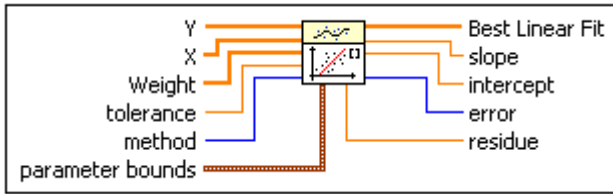
6.2. Blok Build XY Graph.

2 etap tworzenia wykresu to umieszczenie diagramu zajmującego się wykreśleniem jego przed tym jednak musimy połączyć oba bloki **Bundle** tak abyśmy na jednym wykresie mogli uzyskać obie charakterystyki: rzeczywistą jak i zlinearyzowaną. Połączenie **Bundli** robimy za pomocą bloku **Build Array**. Znajduje się on w zakładce: zakładka **Programming** -> **Array** -> : **Build Array**. Musimy w tym bloku stworzyć dwa osobne wejścia. Robimy to za pomocą kliknięcia prawym myszy na bloczek, a następnie **Add Input**. Do wejścia pierwszego podłączamy pierwszy bloczek **Bundle**, do drugiego kolejny. Po połączeniu **Bundli** możemy stworzyć bloczek związany bezpośrednio z charakterystyką. Przechodzimy do okna **Front Panel** na puste pole klikamy prawym myszy i następnie **Modern** -> **Graph indicators** -> **XY Graph**. Wracamy do diagramu, jeśli oprócz **XY Graph** pojawi się niebieskie okno **Build XY Graph** musimy go usunąć, następnie blok wykresu **XY Graph** łączymy z wcześniej stworzonym blokiem **Build Array**.

7. Wyznaczanie stałej oraz czułości.

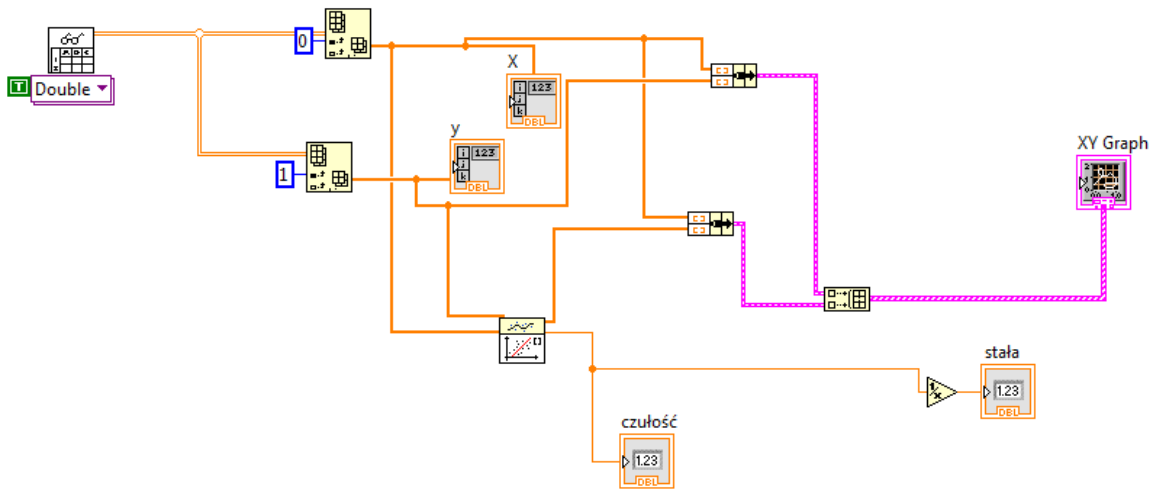
Pomiary Elektryczne Wielkości Nielektrycznych – Ćw. 2

Ażeby wyznaczyć czułość musimy wrócić się do bloczku : **Linear Fit**. Najeżdżamy myszką na



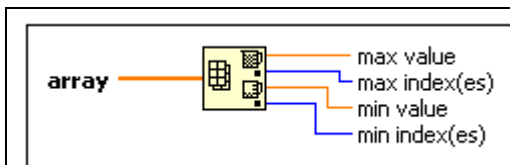
wyjście slope i klikając prawym klawiszem myszy wybieramy opcję **Create** -> **Indicator** następnie wyjście slope łączymy z naszym nowym bloczkiem. Warto również blok podpisać ażeby później nam się nie mylił. Stała jest to odwrotność czułości. Tworzymy ją

kopiując blok wyświetlający czułość, zmieniamy nazwę na stała oraz przed blokiem stawiamy blok odwracający. Znajduje się on w zakładce **Mathematics** -> **Numeric** -> : **Reciprocal**. Obecny nasz schemat powinien wyglądać tak:



8. Wyznaczanie ΔY max.

Tworzymy blok odejmujący, znajduje się on w zakładce **Mathematics** -> **Numeric** -> : **Subtract**. Do wejścia górnego podpinamy wyjście **Best Linear Fit** z bloku **Linear Fit**. Do wejścia dolnego podpinamy naszego Y-greka. Dla przejrzystości projektu najlepiej blok odejmujący przenieść na sam dół naszego schematu ponieważ dalej będą dobudowywane kolejne bloki. Następnie do bloku odejmującego podpinamy blok który wyciąga nam wartość bezwzględna z liczby znajduje się on w zakładce **Mathematics** -> **Numeric** -> : **Absolute Value**. Do bloku **Absolute Value** podpinamy blok **Array Max & Min**. Znajduje się on w zakładce **Programming** -> **Array** -> : **Array Max & Min**.



Następnie w bloku **Array Max & Min** klikamy prawym klawiszem myszki na nóżkę **max value** a następnie **Create** -> **Indicator**. W taki sposób tworzymy ΔY max. Warto blok podpisać dla czytelności schematu.

9. Wyznaczanie zakresu Y.

Kolejny etap to stworzenie bloku który nam wypisze różnicę pomiędzy maksymalną a minimalną wartością Y-greka. Tworzymy bloczek **Array Max & Min** ewentualnie możemy skopiować wcześniej stworzony, na zasadzie zaznaczenia oraz użycia skrótów klawiszowych CTRL+C, CTRL+V, Do wejścia **array** podpinamy nasze wartości Y. Następnie tworzymy blok odejmujący czyli wchodzimy

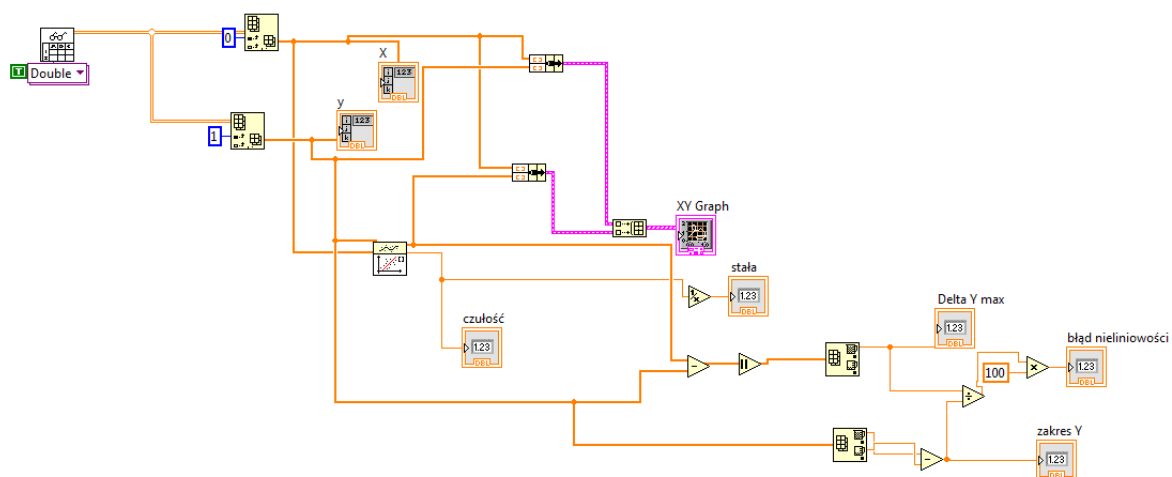
Pomiary Elektryczne Wielkości Nielektrycznych – Ćw. 2

w zakładkę **Mathematics** -> **Numeric** -> : **Subtract**. Do wejścia górnego bloku różnicy podpinamy nóżkę **max value** bloku : **Array Max & Min** do wejścia dolnego podpinamy nóżkę **min value**. Na wyjście z bloku **Subtract** klikamy prawym klawiszem myszy a następnie **Create** -> **Indicator**. Blok **Indicator** łączymy z blokiem odejmującym. W taki sposób powstaje nam wartość różnicy pomiędzy maksymalną a minimalną wartością Y-greka. Dla czytelności schematu zaleca się podpisania bloku.

10. Wyznaczenie błędu nieliniowości.

Tworzymy blok związany z dzieleniem. Znajdziemy go **Mathematics** -> **Numeric** -> : **Divide**. Następnie do górnego wejścia podpinamy wartość ΔY max do dolnego **zakres Y**. Wynik ten następnie musimy pomnożyć przez 100, także tworzymy blok związany z mnożeniem **Mathematics** -> **Numeric** -> : **Multiply**. Do jednego z wejść podpinamy wynik dzielenia na drugą nóżkę klikamy prawym klawiszem myszki i **Create** -> **Constant**. Wynik mnożenia to błąd nieliniowości. Na wyjście klikamy **Create** -> **Indicator**. Dla czytelności schematu zaleca się podpisania bloku.

Cały nasz projekt powinien wyglądać następująco:

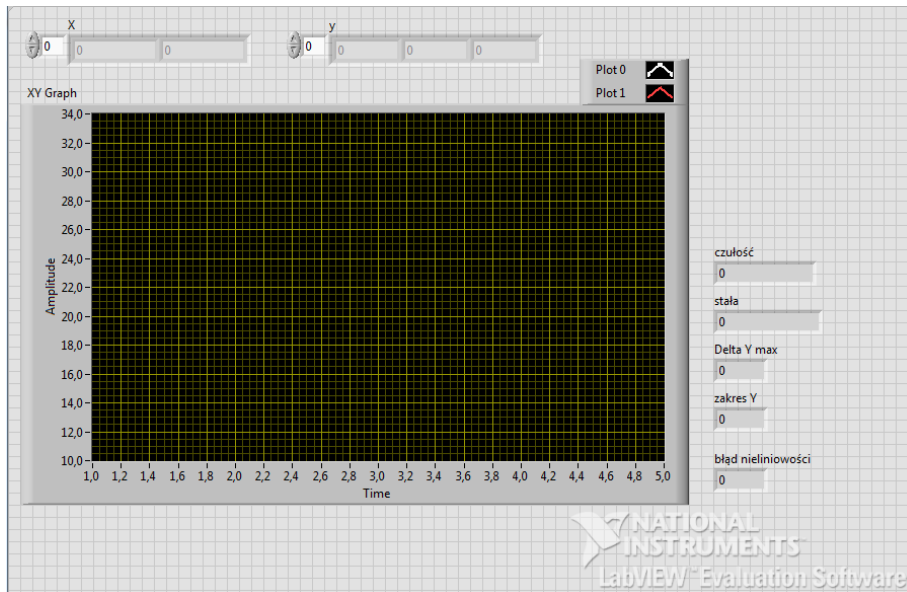


11. Testowanie.

Następny proces to testowanie układu. Zanim jednak to zrealizujemy musimy stworzyć szatę graficzną na Front panelu. Elementy jakie tutaj znajdują się pochodzą od bloczków z diagramu. Jeżeli wszystkie wypisy zostały prawidłowo nazwane panel powinien być czytelny.

Przykładowe rozłożenie okien w panelu:

Pomiary Elektryczne Wielkości Nielektrycznych – Ćw. 2

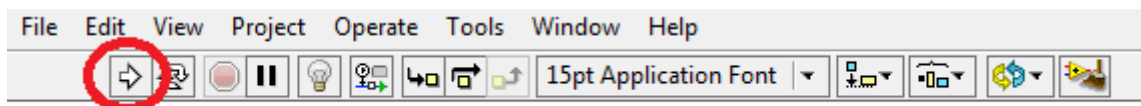


Dla przykładu i sprawdzenia działalności programu należy stworzyć plik txt o dowolnej nazwie z następującymi wartościami:

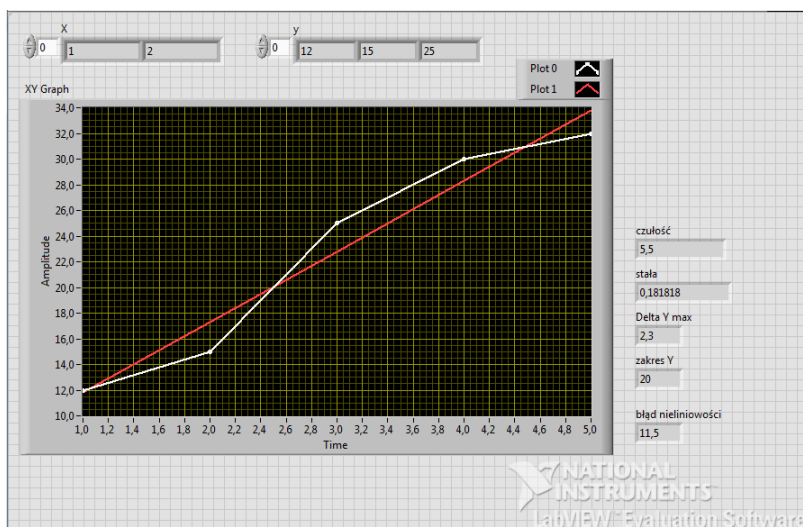
1	12
2	15
3	25
4	30
5	32

Liczby oddzielony są tabulatorami, każda kolejna para liczb znajduje się w następnym wierszy.

Załadowanie pliku do naszego programu odbywa się poprzez naciśnięcie przycisku run.



Po wczytaniu wartości jeżeli nasz program działa prawidłowo Front Panel powinien wyglądać następująco:



Pomiary Elektryczne Wielkości Nielektrycznych – Ćw. 2

Program należy zapisać na dysku C://.

Pytania kontrolne:

1. Co to jest panel, blok diagram, jakie spełniają funkcję w LabVIEW?
2. Czym różnią się obiekty Controls i Indicators?
3. O czym informuje grubość ramki, kolor w LabVIEW?
4. Wymień struktury występujące w LabVIEW.

Literatura:

1. LabVIEW – User Manual, Measurement Manual, National Instruments 2003 (pdf).
2. Świsulski D. Komputerowa technika Pomiarowa – oprogramowanie wirtualnych przyrządów w LabVIEW. Agenda wyd. PAK, Warszawa 2005.
3. Lesiak P., Świsulski D. Komputerowa technika pomiarowa w przykładach. Agenda wyd. PAK, Warszawa 2002.
4. Tłaczała W. Środowisko LabVIEW w eksperymencie wspomaganym komputerowo. WNT, Warszawa 2002.
5. Winiński W., Organizacja komputerowych systemów pomiarowych, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 1997.
6. Marks-Wojciechowska Z., Pacholski K., Kulesza W.: Systemy pomiarowe, Politechnika Łódzka, 1999.
7. Świsulski D.: Systemy pomiarowe - laboratorium, Politechnika Gdańska, 2001.
8. Winiński W., Stanik B., Nowak J.: Graficzne, zintegrowane środowiska programowe do projektowania komputerowych systemów pomiarowo-kontrolnych. Mikom, Warszawa 2001.
9. Lesiak P., Gołąbek P. Laboratorium aparatury pomiarowo-diagnostycznej cz.2-Komputerowe systemy pomiarowo-diagnostyczne. Pol. Radomska, Radom 2005.
10. Świsulski D. Systemy pomiarowe. Wyd. Pol. Gdańskiej, Gdańsk 2004.
11. Winiński W.: Wirtualne przyrządy pomiarowe. O.W. Pol. Warszawskiej, Warszawa 2003.
12. Winiński W.: Organizacja komputerowych systemów pomiarowych. O.W. Pol. Warszawskiej, Warszawa 1997.
13. Rak R. Wirtualny przyrząd pomiarowy – realne narzędzie współczesnej metrologii. O.W. Pol. Warszawskiej, Warszawa 2003.
14. Tłaczała W. Środowisko LabVIEW w eksperymencie wspomaganym komputerowo. WNT, Warszawa 2002.komputerowo. WNT, Warszawa 2002.