

Ćwiczenie

Programowanie w środowisku LabVIEW - Obsługa autonomicznych przyrządów pomiarowych – pomiar mocy z wykorzystaniem analizatora mocy.

Celem ćwiczenia jest zapoznanie studenta z metodyką programowania, projektowania i tworzenia aplikacji, których zadaniem jest przybliżenie zagadnień i problemów występujących w systemach pomiarowych. W trakcie ćwiczenia Student nabeździe podstawowe informacje dotyczące środowiska i umiejętności posługiwania się nim oraz podstawy programowania przyrządów pomiarowych.



**Zakład Systemów Informacyjno-
Pomiarowych**

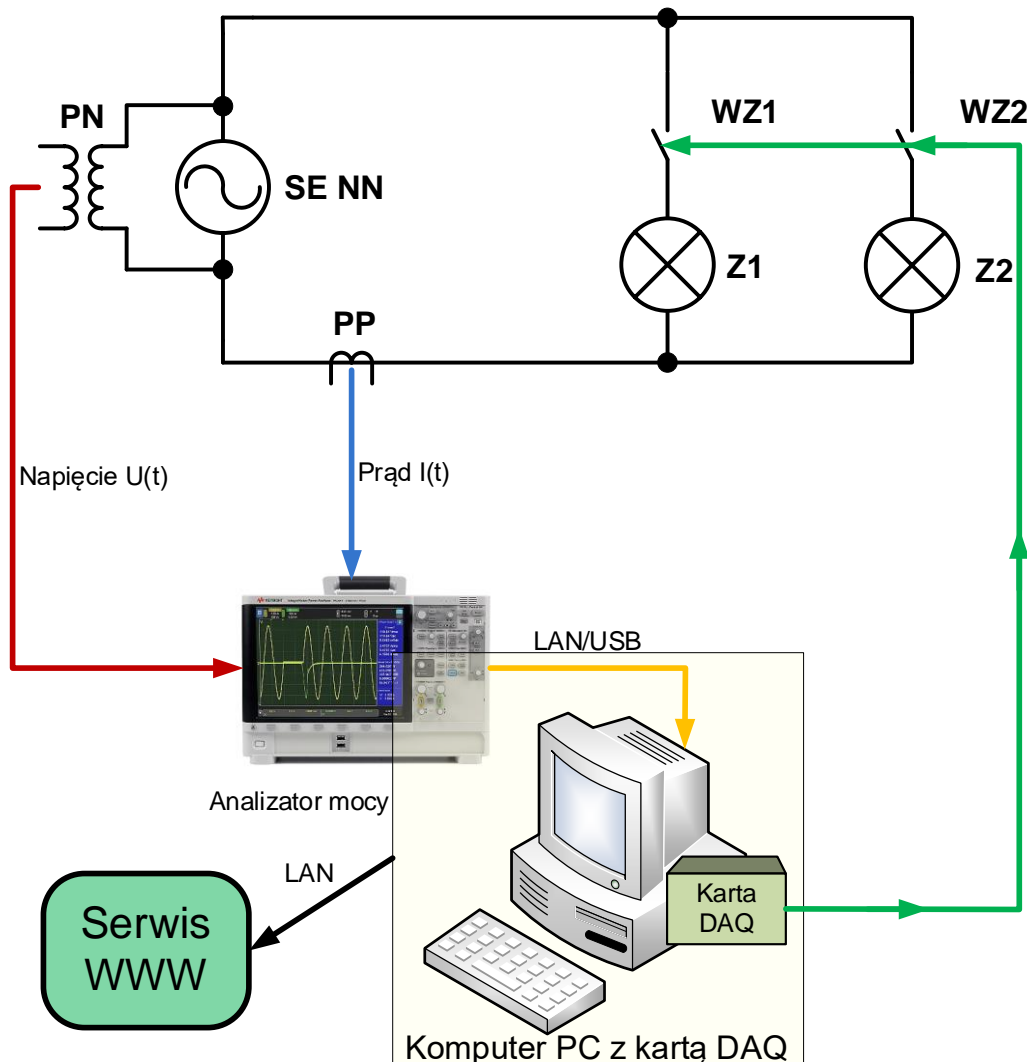
IETiSIP, Wydział Elektryczny, PW



1. Cel ćwiczenia

Celem ćwiczenia jest praktyczne wprowadzenie i zapoznanie Studentów z możliwościami, jakie oferują obecnie środowiska programistyczne przeznaczone do tworzenia systemów pomiarowych, w zakresie obsługi przyrządów autonomicznych. W ramach ćwiczenia przygotowane zostały zadania, w trakcie realizacji, których studenci zdobywają wiedzę dotyczącą sposobów łączenia przyrządów autonomicznych z komputerem, konfiguracji przyrządów w środowisku programistycznym, możliwości podłączenia wielu przyrządów tworzących system pomiarowy. Zasadniczym elementem ćwiczenia jest programowanie przyrządu pomiarowego z poziomu środowiska programistycznego. Dodatkowo Studenci zdobywają umiejętności w podstawowym zakresie obejmującym budowę, modyfikację, a nawet tworzenie własnego sterownika przyrządu pomiarowego.

2. Układ pomiarowy



Rysunek 1. Schemat układu pomiarowego

Zakład Systemów Informacyjno-Pomiarowych



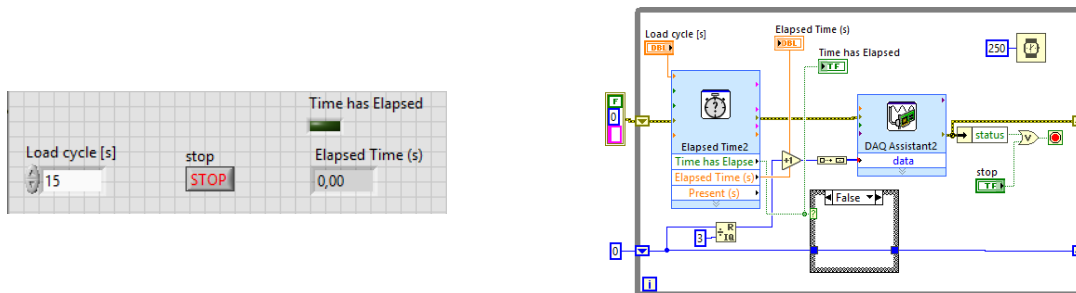
IETiSIP, Wydział Elektryczny, PW



Układ pomiarowy składa się z komputera klasy PC pełniącego funkcję kontrolera i umożliwiającego uruchamianie aplikacji sterujących oraz komunikacje z przyrządem pomiarowym. Dodatkowo za pomocą również komputera możliwa jest cykliczna zamiana obciążenia włączanego w tor pomiarowy analizatora mocy. Analizator mocy połączony jest z komputerem poprzez sieć komputerową. Dane pomiarowe umieszczane są każdorazowo w zasobach WWW, skąd można je pobrać i przetworzyć.

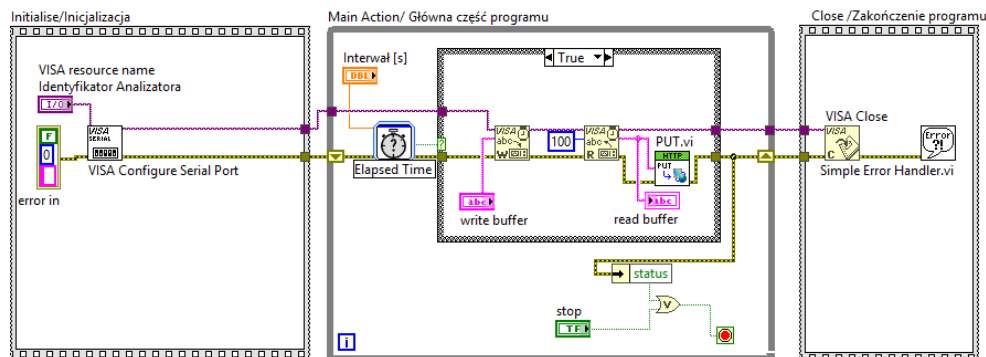
3.Oprogramowanie sterujące analizatorem

Układ pomiarowy sterowany jest automatycznie za pomocą dwóch programów uruchomionych na komputerze. Pierwszy z nich odpowiada za sterowanie obciążeniem układu pomiarowego, w tym przypadku są to dwie żarówki niezależnie dołączane i odłączane od obwodu obciążenia. Sam program jest bardzo prosty i sprowadza się do wysyłania cyklicznie wartości 1, 2 i 3 na wyjścia portu cyfrowego (DO) karty DAQ. Jest to równoznaczne z włączeniem tylko jednej, następnie tylko drugiej i na koniec oby razem żarówek. Interfejs użytkownika pozwala na wybranie przedziału czasu, po którym następuje przełączenie żarówki czyli zmiana obciążenia. Płytkę czołową wraz z kodem przedstawia rysunek 2.



Rysunek 2. Program sterujący obciążeniem.

Zasadniczy program sterujący analizatorem jest dość złożony ale można go zilustrować za pomocą rysunku 3.



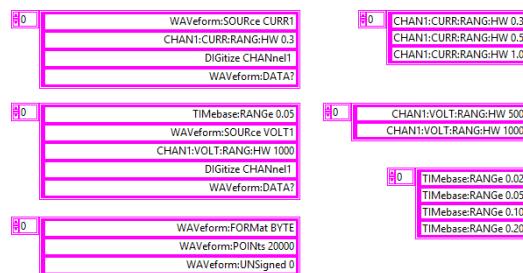
Rysunek 3. Program sterujący analizatorem.

Zakład Systemów Informacyjno-Pomiarowych



IETiSIP, Wydział Elektryczny, PW

Sama struktura aplikacji sterującej jest bardzo prosta i typowa. A początku przyrząd pomiarowy jest konfigurowany na ustaloną, początkową wartość podstawy czasu oraz wzmocnienie toru Y, niezależnie dla sygnału prądowego i napięciowego. Konfiguracja podstawy czasu oraz liczby próbek przekłada się na częstotliwość próbkowania. Następnie w pętli głównej programu ustalone na wstępie parametry są cyklicznie modyfikowane. Dotyczy to zakresu prądowego, zakresu napięciowego i podstawy czasu. Ze względu na zdalny dostęp do przyrządu operacja ta oprócz możliwości porównania otrzymanych, w poszczególnych konfiguracjach wyników daje informację o faktycznej pracy układu pomiarowego (rejestrowane przebiegi znacząco różnią się między sobą). Zestaw rozkazów inicjujących oraz zakres zmian parametrów ilustruje rysunek 4.

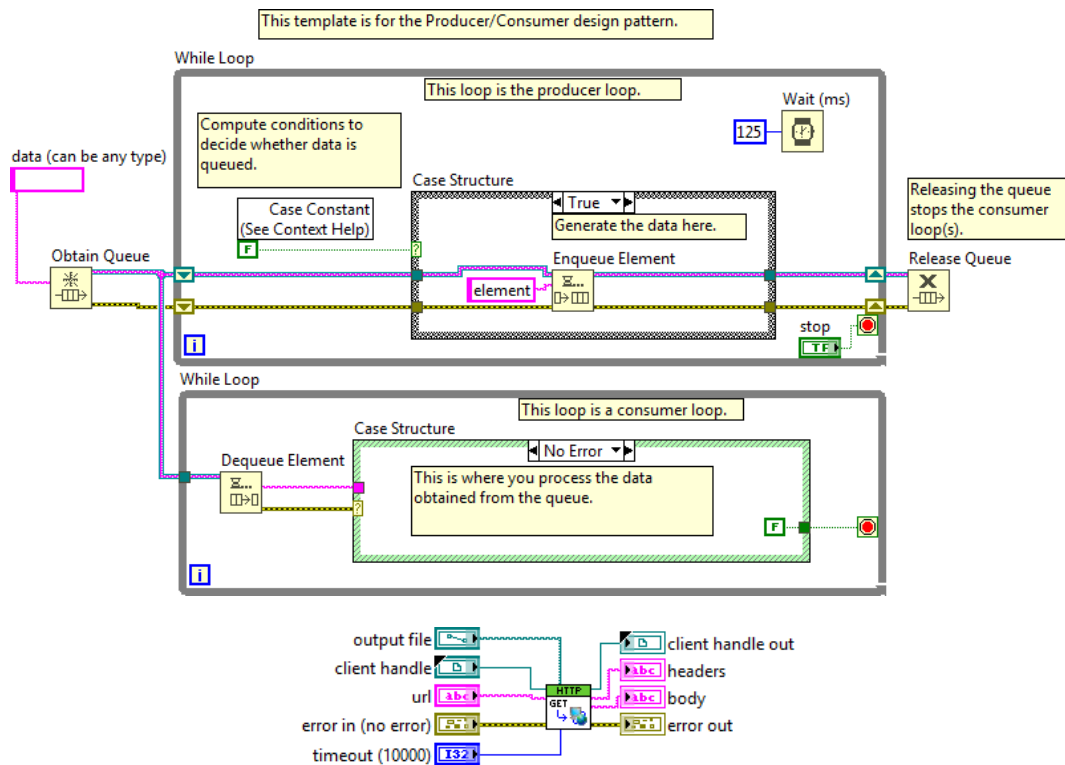


Rysunek 4. Zestaw instrukcji inicjujących (po lewej) oraz zakres zmian parametrów (po prawej)

Z lewej strony na rysunku 4 przedstawione są trzy zestawy rozkazów inicjujących przyrząd pomiarowy. Dotyczą one, patrząc od góry konfiguracji toru prądowego (kanał 1), konfiguracji toru napięciowego wraz z dodanym rozkazem konfigurującym podstawę czasu oraz konfigurację formatu danych i liczby próbek w buforze. Zestaw instrukcji dla sygnału napięciowego oraz prądowego zawiera dodatkowo rozkazy próbkowania i pobrania sygnału do komputera (**DIG CHAN1** oraz **WAV:DATA?**). Wysłanie rozkazów inicjujących pozwala już na odczyt przebiegów napięcia i prądu. Następnie program przechodzi do zasadniczego trybu pracy, w którym cyklicznie zmienia ustawienia niezależnie toru napięciowego, prądowego oraz podstawy czasu (zestaw rozkazów po prawej stronie rysunku 4). Wybrane zakresy nie powodują przekroczenia zakresu pomiarowego przyrządu w układzie pomiarowym ćwiczenia. Po odczytaniu przebiegów z przyrządu, są one każdorazowo zapisywane do pliku i wraz ze zrzutem oryginalnej zawartości ekranu analizatora wysyłane na serwer www (zasoby IETiSIP). Operacje te reprezentuje funkcja **PUT.vi** na rysunku 3. Z zasobów www dane można pobrać funkcją komplementarną do **PUT.vi** tj. **GET.vi** i po przetworzeniu wyświetlić oraz wykorzystać do obliczeń mocy na komputerach/stacjach



„klienckich”. Program można oprzeć na szablonie producent konsument, w którym funkcja `GET.vi` odpowiadać będzie za odczyt danych (rysunek 5).



Rysunek 5. Przykładowy punkt startowy programu do odczytu danych pomiarowych i wyznaczania mocy.

4. Obliczenia - pomiar mocy czynnej

Wielkością wyjściową przy obliczaniu mocy czynnej jest moc chwilowa w obwodzie jako iloczyn chwilowych wartości napięcia i prądu (1).

$$p(t) = u(t) * i(t) \quad (1)$$

Jeśli moc chwilowa zostanie uśredniona za pojedynczy okres w wyniku otrzymuje się definicyjną postać mocy czynnej (2).

$$P = \frac{1}{T} \int_0^T p(t) dt = \frac{1}{T} \int_0^T u(t) * i(t) dt \quad (2)$$

W dziedzinie sygnałów dyskretnych powyższa zależność przyjmuje następującą postać (3).

$$P = \frac{1}{N} \sum_{n=0}^{N-1} u(n) * i(n) \quad (3)$$



Zakład Systemów Informatycznych
Pomiarowych



IETiSIP, Wydział Elektryczny, PW

Liczbę próbek do obliczeń należy ustalić tak aby wziąć pod uwagę dokładnie pojedynczy okres napięcia (i prądu) sieciowego.

5. Realizacja ćwiczenia

Zasada działania układu pomiarowego jest następująca: układ w sposób ciągły dokonuje rejestracji napięcia i prądu w obwodzie. Rejestracje są powtarzane cyklicznie z interwałem minutowym. Równocześnie i również cyklicznie zmieniane jest obciążenie. Rejestracje dokonywane są dla różnych wzmocnień torów Y (niezależnie dla prądu i napięcia) oraz z różnym okresem próbkowania. Dane pomiarowe wysyłane są na serwis WWW skąd można je pobrać. Adres zasobu jest następujący:

http://sip.iem.pw.edu.pl/power/PA2203_Coef_Data.dat

Dodatkowo, rejestrowana jest również zawartość ekranu analizatora mocy. Jest ona również umieszczana w zasobach WWW, pod adresem:

http://sip.iem.pw.edu.pl/power/PA2203_display.png

Wygląd ekranu zapisany jest w postaci pliku graficznego, który można wykorzystać dla porównania z własnymi przekształceniami sygnałów pomiarowych.

Przebiegi napięcia i prądu mają postać surowych wartości wyjściowych przetwornika A/C, dlatego też dane pomiarowe zawierają dodatkowo współczynniki skalujące amplitudowe i czasowe, które pozwalają przejść na wielkości fizyczne napięcia, prądu oraz czasu. W celu wyeliminowania zagrożenia, że pobrane z zasobów WWW przebiegi napięcia i prądu pochodzą z różnych pomiarów wszystkie dane włączone zostały do pojedynczego pliku **PA2203_Coef_Data.dat**. Jego zawartość przedstawia rysunek 6:

Nagłówek #1 4 bajty Liczba danych	współczynniki skalujące format współczynników <i>double</i>	Nagłówek #2 4 bajty Liczba danych	Próbki przebiegu prądu format próbek <i>I16</i>	Nagłówek #3 4 bajty Liczba danych	Próbki przebiegu napięcia format próbek <i>I16</i>
---	--	---	--	---	---

Rysunek 6. Format danych pomiarowych.

Plik z danymi pomiarowymi ma postać binarną i składa się z trzech bloków. Każdy blok z kolei rozpoczyna się 4-bajtowym nagłówkiem zawierającym informację o liczbie danych/próbek w zasadniczej części bloku. I tak:

- blok pierwszy zawiera 4 współczynniki. Nagłówek pierwszego bloku zawiera liczbę 4, co może ułatwić jego rozkodowanie. Format danych w tym bloku to **double** czyli długość tablicy ze współczynnikami wynosi 4x8 bajtów. Zawartość tej tablicy to kolejno:



Zakład Systemów Informatycznych
Pomiarowych

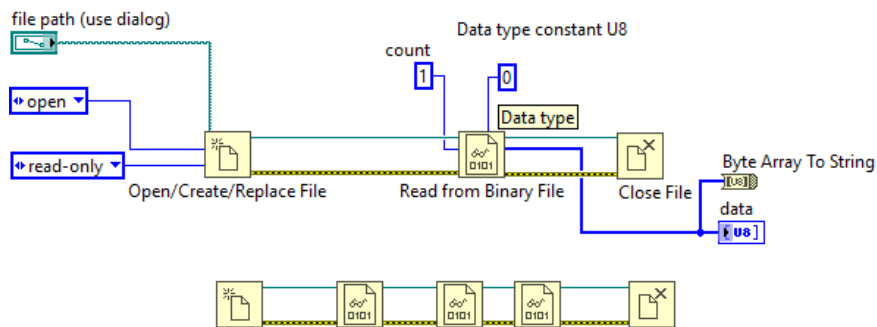


IETiSIP, Wydział Elektryczny, PW

- o współczynnik skalujący dla napięcia
- o współczynnik skalujący dla prądu
- o okres próbkowania
- o liczba próbek

Na podstawie ww. parametrów można wyznaczyć częstotliwość próbkowania i dobrać tak liczbę próbek sygnału aby opisywały one dokładnie pojedynczy okres sygnału (ewentualnie jego wielokrotność)

- blok drugi zawiera przebieg prądu w formacie I16, czyli całkowitoliczbowym z 2 bajtami na próbkę. Informacja o liczbie próbek zapisana jest 4-bajtowym nagłówku tego bloku.
- blok trzeci zawiera przebieg napięcia w formacie I16, czyli całkowitoliczbowym z 2 bajtami na próbkę. Informacja o liczbie próbek zapisana jest 4-bajtowym nagłówku tego bloku. Rozmiary bloków drugiego i trzeciego są takie same.



Rysunek 7. Odczyt danych.

Dane odczytać można za pomocą wywołania z rysunku 7. Funkcja **Read from Binary File** musi mieć ustawiony bajtowy typ danych. Można to zrobić podłączając stałą typu U8. Wyjście **data** funkcji **Read from Binary File** może być odczytywane binarnie (bezpośrednio jako tablica bajtowa) albo po wykorzystaniu funkcji **Byte Array To String** jako ciąg tekstowy – **string**. Wszystko zależy od tego jak będzie wygodniej (wykorzystywane funkcje). Odczyt można zrealizować za jednym razem, czytając cały plik. W tym celu trzeba ustawić liczbę bajtów do odczytu **count** na -1 (rysunek 7). W dalszej kolejności należy podzielić tak odczytane dane na poszczególne bloki i osobno je przekonwertować. Można też operację tę wykonać krokowo:

- odczytać pierwszy nagłówek (4 bajty) zawierający informację o długości pierwszego bloku w bajtach
- na podstawie odczytanej w poprzednim kroku informacji o liczbie bajtów w bloku pierwszym, odczytać taką właśnie porcję danych (w bajtach) zawierającą współczynniki skalujące



- odczytać drugi nagłówek (4 bajty) zawierający informację o długości drugiego bloku w bajtach
- na podstawie odczytanej w poprzednim kroku informacji o liczbie bajtów w bloku drugim, odczytać taką właśnie porcję danych (w bajtach) zawierającą przebieg prądu
- odczytać trzeci nagłówek (4 bajty) zawierający informację o długości trzeciego bloku w bajtach
- na podstawie odczytanej w poprzednim kroku informacji o liczbie bajtów w bloku trzecim, odczytać taką właśnie porcję danych (w bajtach) zawierającą przebieg napięcia

6. Wskazówki do realizacji ćwiczenia

- Pobieranie danych z serwisu WWW należy wykonać za pomocą funkcji GET.vi – instrukcja pobierania danych analizatora mocy
- sposób realizacji konwersji danych opisany jest w instrukcji dotyczącej obsługi sprzętowego symulatora przyrządu pomiarowego (z interfejsem GPIB)
- wyznaczenie mocy opisane jest w instrukcji opisującej pomiary w systemie rozproszonym

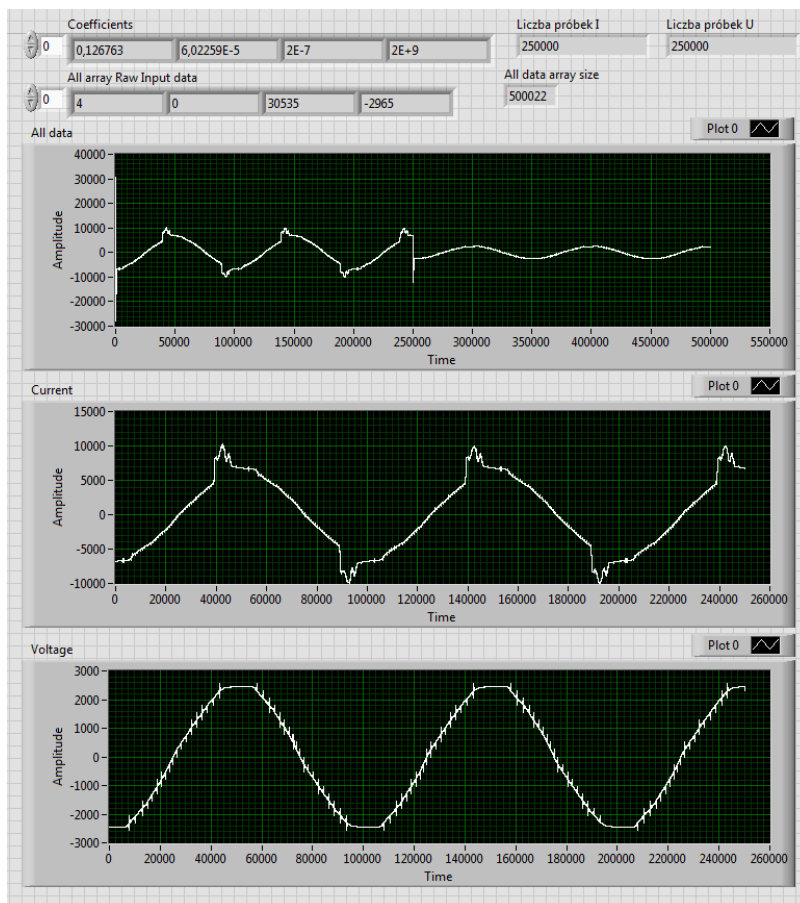
W ćwiczeniu należy wyznaczyć przebieg mocy w funkcji czasu wytwarzanej w obwodzie elektrycznym, w którym obciążenie stanowią przełączane żarówki (rysunek 1). Należy także wyznaczyć współczynnik THD oraz widma częstotliwościowe napięcia i prądu – tu z pomocą może przyjść instrukcja do ćwiczenia opisującego symulator oscyloskopu/generatora przebiegów okresowych. Przebiegi napięcia i prądu mogą wymagać filtracji. Przykładowe, zarejestrowane przebiegi i wartości współczynników skalitycznych przedstawiono na rysunku 8.



**Zakład Systemów Informacyjno-
Pomiarowych**

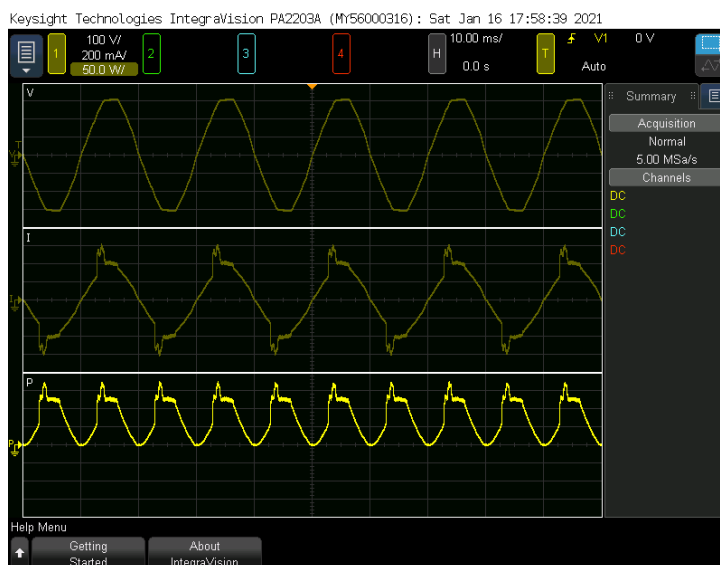


IETiSIP, Wydział Elektryczny, PW



Rysunek 8. Przykładowe zarejestrowane przebiegi.

Na rysunku 9 przedstawiono zaś widok analizatora, który można potraktować jako ekran kontrolny.



Rysunek 9. Przykładowy pomiar analizatorem PA2203.



Zakład Systemów Informacyjno-Pomiarowych



IETiSIP, Wydział Elektryczny, PW

Dane odbierane z przyrządu mogą mieć postać ciągu ASCII bądź danych binarnych (składających się z wartości jedno lub wielobajtowych). Wynika to z aktualnej konfiguracji przyrządu. Dla ciągów ASCII otrzymane wyniki można przekonwertować do postaci zmiennoprzecinkowej wykorzystując funkcję **Extract Numbers**, która będzie odpowiedzialna za przekonwertowanie odpowiedzi symulatora do postaci tablicy elementów **double**. W przypadku konfiguracji binarnej najczęściej wyniki mają postać formatu binarnego jednobajtowego lub dwubajtowego. W tym drugim przypadku ustalić należy dodatkowo porządek danych **little** bądź **big endian**.

Przykładowe formaty danych:

Floating Point ASCII (default)

```
#header<space>num0,num1,...,num127,...<EOF>
```

The floating point format used is [+][-]1.2345[E[+][-]0]

8-bit Unsigned Binary

```
#header<Byte 0><Byte 1>...<Byte127>...<EOF>
```

16-bit Signed Binary (NORMal byte order)

```
#header<MSB 0><LSB 0>...<MSB 127><LSB 127>...<EOF>
```

16-bit Signed Binary (SWAPped byte order)

```
#header<LSB 0><MSB 0>...<LSB 127><MSB 127>...<EOF>
```

Jak widać nagłówki może przybierać różne formy, ale najistotniejsze jest to, że zawiera użyteczne informacje. W przypadku formatów binarnych często podawana jest liczba bajtów jaką zajmują dane zawarte w buforze odpowiedzi oraz współczynniki skalujące, gdyż dane przesyłane w postaci binarnej „surowej” zawierają wyniki konwersji ADC. Współczynniki pozwalają na otrzymanie wartości w jednostkach wielkości mierzonej. W przypadku formatu ASCII konwersja jest bardzo prosta i wymaga jedynie funkcji **Extract Numbers**. W przypadku formatów binarnych konwersję można wykonać korzystając z funkcji **Type Cast** (Rysunek 10). Poniższe przykłady dotyczą typowego wykorzystania funkcji do obsługi interfejsu GPIB/LXI i wykorzystują ciągi tekstowe (pojedynczy element takiego ciągu to bajt co zapewnia wygodną i uniwersalną formę przechowywania danych). W przypadku danych przekazywanych w formacie natywnie binarnym (I8/U8) postępowanie przy ewentualnej konieczności konwersji oparte jest na tych samych zasadach. Dotyczy to przypadku kiedy dane wynikowe są np.:

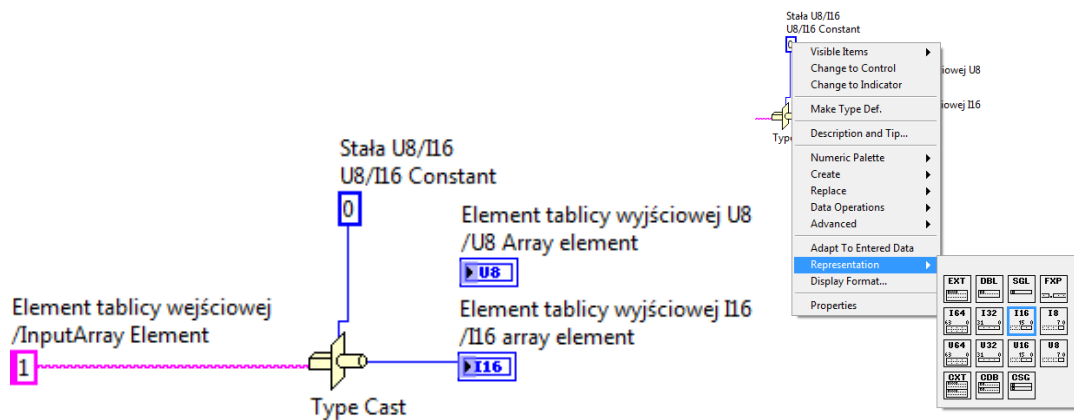


Zakład Systemów Informatycznych
Pomiarowych



IETiSIP, Wydział Elektryczny, PW

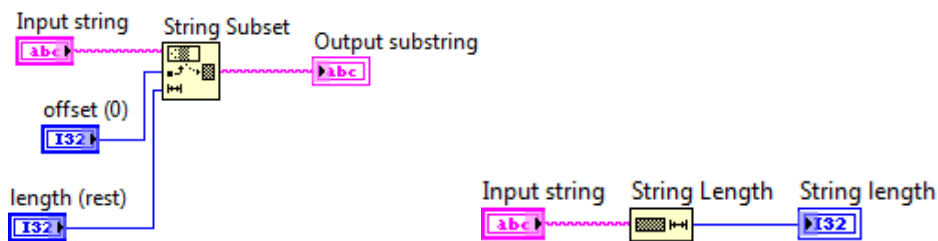
dwubajtowe czyli jedna próbka zajmuje dwa bajty. Należy tylko zastosować odpowiednie funkcje.



Rysunek 10. Argumenty funkcji **Type Cast**. Typ elementów wyjściowych zależy od typu stałej (jej wartość jest nieistotna).

W przypadku tablicy alfanumerycznej (ASCII) otrzymywanej dynamicznie na wyjściu funkcji **GPIB Read** (i zawierającej wartości kolejnych próbek przebiegu), nie jest dostępny mechanizm automatycznego indeksowania tej tablicy. W celu przeprowadzenia konwersji można wykonać poniższą sekwencję kroków (Przedstawione rozwiązanie jest jedynie propozycją, gdyż zapewne zadanie konwersji ma inne rozwiązania):

1) Usunięcie nagłówka za pomocą funkcji **String Subset** (Rysunek 11). Głównym wejściem tej funkcji jest **Input String**, zaś wyjściem **Output Substring**. Parametrami sterującymi operacją kopiowania ciągu z wejścia na wyjście funkcji są **offset** oraz **length**. Warto zauważyć, że ich domyślne wartości wynoszą odpowiednio **0** oraz **rest** (wszystkie pozostałe) co może być ułatwieniem przy korzystaniu z tej funkcji.

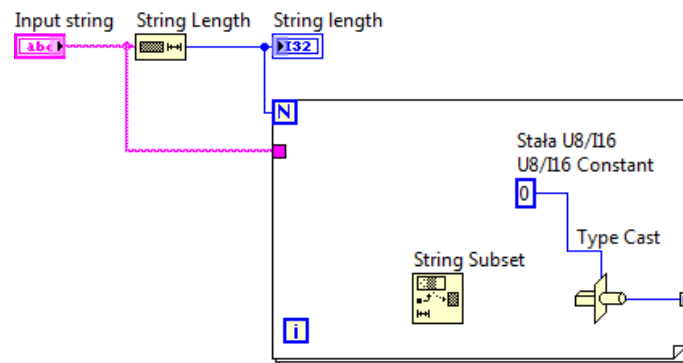


Rysunek 11. Funkcje **String Subset** oraz **String Length**.

W analogiczny sposób można usunąć ostatni znak (znak końca) w tablicy odpowiedzi symulatora (<LF> - patrz format odpowiedzi symulatora).

2) Po usunięciu nagłówka i znaku końca można przystąpić do zasadniczej konwersji. Należy jeszcze wyznaczyć liczbę elementów tablicy

alfanumerycznej (funkcja **String Length**) i wykorzystać ją do ustawienia liczby iteracji pętli **for**, w której wykonywana będzie konwersja (Rysunek 12).



Rysunek 12. Pętla „konwertująca” **string** do **U8/I16**.

W pętli wykorzystano raz jeszcze funkcję **String Subset**, tym razem do pobierania pojedynczych elementów tablicy alfanumerycznej (**length** = 1, a parametr **offset** należy skojarzyć ze zmienną indeksującą **i** pętli **for**). Na wyjściu pętli dostępne jest automatyczne indeksowanie (wyjście funkcji **Type Cast**), stąd dane automatycznie umieszczane są w tablicy i są gotowe do wykorzystania w programie.

W analogiczny sposób można przekonwertować dane do formatu **I16**. Tu należy pamiętać o tym, że podwaja się liczba bajtów w tablicy przy niezmiennej liczbie elementów (porównaj nagłówek **U8** i **I16**). Należy modyfikować wartość parametru **offset** pamiętając, że każda próbka zapisana jest na dwóch bajtach.

