

Zagadnienie 1

INTERAKCJA CZŁOWIEK MASZYNA

Programowanie w środowisku LabVIEW – pliki konfiguracyjne ini, CSV, XML.

Celem ćwiczenia jest zapoznanie studenta z metodyką programowania, projektowania i tworzenia aplikacji. Podstawowym zadaniem jest przybliżenie zagadnień i problemów występujących w systemach SCADA. W trakcie ćwiczenia student nabędzie podstawowe informacje dotyczące środowiska i umiejętności posługiwania się nim w procesie tworzenia aplikacji pomiarowych. Niniejsze zagadnienie dotyczy wykorzystania plików konfiguracyjnych, w zakresie tworzenia, zapisu o odczytu.



**Zakład Systemów Informacyjno-
Pomiarowych**

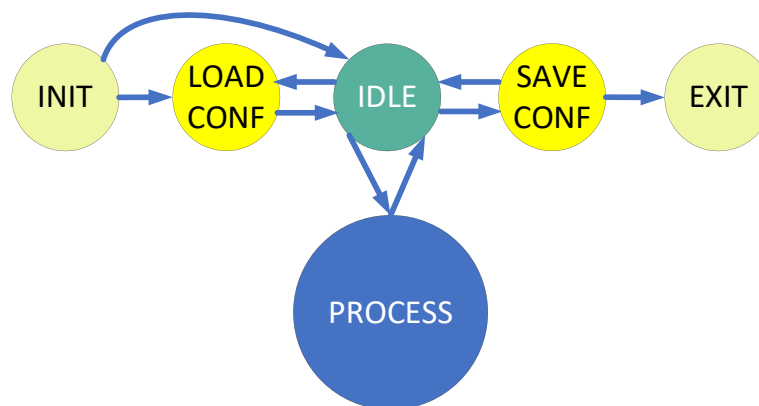
IETiSIP, Wydział Elektryczny, PW



W materiałach do przedmiotu znajdują się przykładowe programy, które mogą okazać się pomocne.

1. Zagadnienie: Zapis i odczyt plików konfiguracyjnych: .ini, CSV, XML. Zadanie jest swego rodzaju rozszerzeniem prostego resetowania nastaw płyty czołowej do wartości domyślnych i ich synchronizacji w kodzie aplikacji. Pliki konfiguracyjne są powszechnie wykorzystywane w aparaturze pomiarowo-sterującej czy też w infrastrukturze interfejsów komunikacyjnych.

2. Możliwe rozwiązanie: Dodatkowe, dedykowane funkcje w aplikacji oscyloskopu/ generatora. W najprostszej wersji odczyt i zapis może być realizowany każdorazowo przy odpowiednio, rozpoczęciu/zakończeniu pracy programu. Warto jednak dodać funkcję zapisu/odczytu w dowolnym momencie pracy aplikacji. W takim razie schemat działania programu może wyglądać jak na rysunku 1.



Rys. 1. Schemat działania programu z możliwością odczytu/zapisu konfiguracji.

Schemat przedstawiony na rysunku 1 zawiera dodatkowe stany związane z odczytem i zapisem konfiguracji. Ich położenie na diagramie pozwala na wywołanie zarówno na początku działania programu, jak i na żądanie w trakcie zasadniczej pracy aplikacji. Dodatkowa, bezpośrednia ścieżka przejścia ze stanu INIT do IDLE daje alternatywną możliwość warunkowego pominięcia załadowania danych z pliku konfiguracyjnego i np.: wykorzystania wartości domyślnych.

Rozwiązanie zadania należy realizować w projekcie LabVIEW. Dobrze jest też dodać syntetyczną instrukcję obsługi.

3. Przykładowy format danych:

- plik ini:

```
[CHAN1] - (sekcja)
WzmY = 1 - (klucz)
WzmX = 0.001
Tryb = AC
```



Zakład Systemów Informacyjno-
Pomiarowych

IETiSIP, Wydział Elektryczny, PW



Trigger = NORMAL

[CHAN2]

WzmY = 2

WzmX = 0.005

Tryb = DC

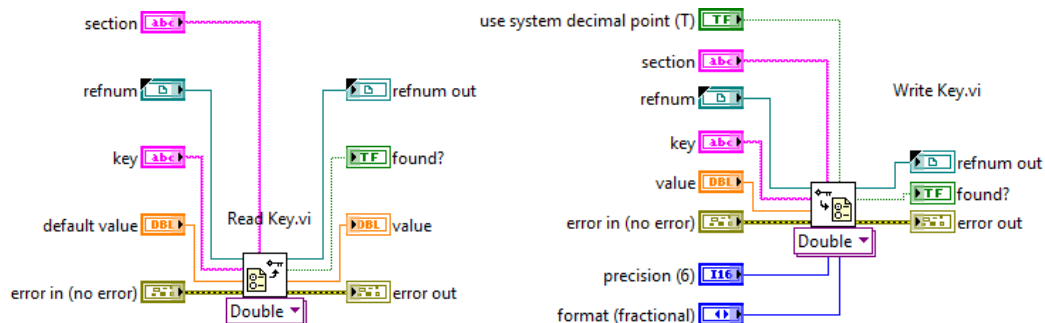
Trigger = AUTO

[Opcje]

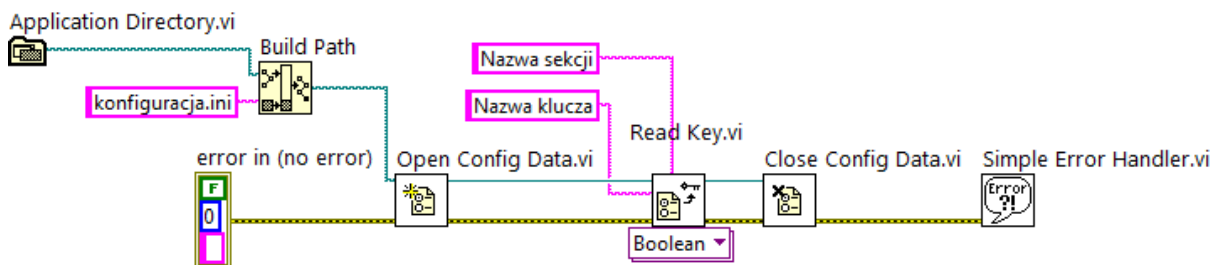
Plik = TRUE

Przydatne funkcje:

- Open Config Data.vi / Close Config Data.vi
- Get Section Names.vi
- Read Key.vi / Write Key.vi



Rys. 2. Wejścia i wyjścia funkcji Read Key.vi/Write Key.vi



Rys. 3. Standardowy odczyt klucza z sekcji w pliku konfiguracyjnym (klucz typu boolean).

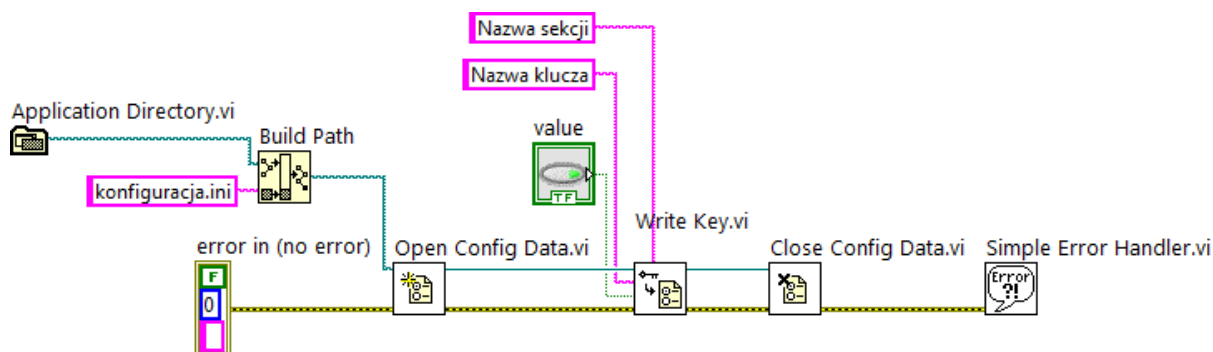
W trybie odczytu pliku konfiguracyjnego warto wykorzystać funkcję **Get Section Name.vi** umieszczając ją zaraz po funkcji **Open Config Data.vi** a przed **Read Key.vi**. Zwróci ona na wyjściu w tablicy nazwy wszystkich sekcji w pliku. Dzięki temu można (wykorzystując dodatkowo pętlę **for**) odczytywać sukcesywnie kolejno wszystkie/wybrane klucze z poszczególnych sekcji.



Zakład Systemów Informatycznych
Pomiarowych

IETiSIP, Wydział Elektryczny, PW





Rys. 4. Standardowy zapis klucza z sekcji w pliku konfiguracyjnym (klucz typu boolean).

Terminale **refnum/refnum out** przechowują "uchwyt" do pliku konfiguracyjnego, który dostępny jest po wywołaniu funkcji **Open Config Data.vi**. Terminal **section** zawiera nazwę sekcji (np: CHAN1), a **key** zawiera nazwę klucza do odczytania/zapisania (np: Tryb). Sekcje można odczytywać/zapisywać w pętli **for** co ułatwia rozbudowywanie pliku ini. Terminal **value** zawiera wartość odczytanego/zapisywanego klucza. Obie funkcje mają dodatkowy terminal widoczny na dole do ustalenia typu przechowywanej zmiennej (terminal ten wyświetla się po wybraniu menu kontekstowego: **Visible Items->Polymorphic VI Selector**)

- plik CSV

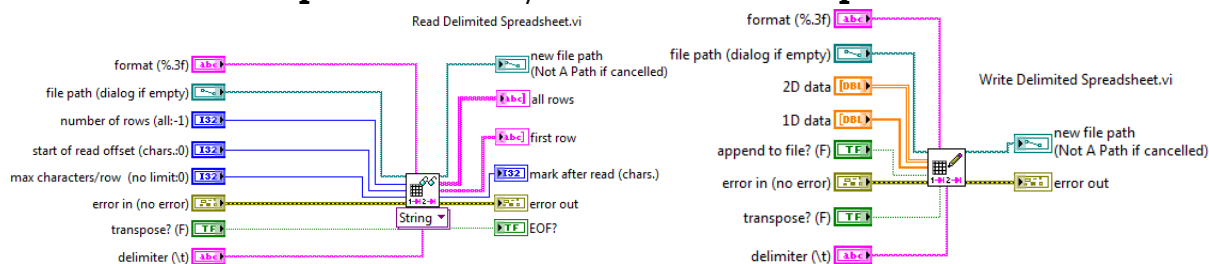
1,0.001,AC,NORMAL - (parametry kanału 1 tak jak w pliku ini)

2,0.005,DC,AUTO - (parametry kanału 1 tak jak w pliku ini)

TRUE - (opcje tak jak w pliku ini)

Do odczytu/zapisu danych z/do pliku CSV można wykorzystać funkcje:

Read Delimited Spreadsheet.vi/Write Delimited Spreadsheet.vi



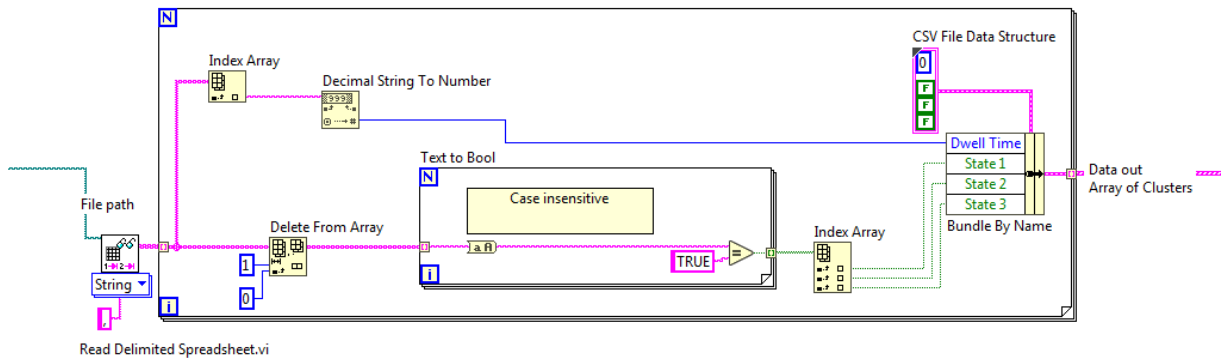
Rys. 5. Wejścia i wyjścia funkcji do obsługi plików CSV. Warto zwrócić uwagę na **delimiter** oraz **Polymorphic VI Selector** w funkcji **Read Delimited Spreadsheet.vi**.



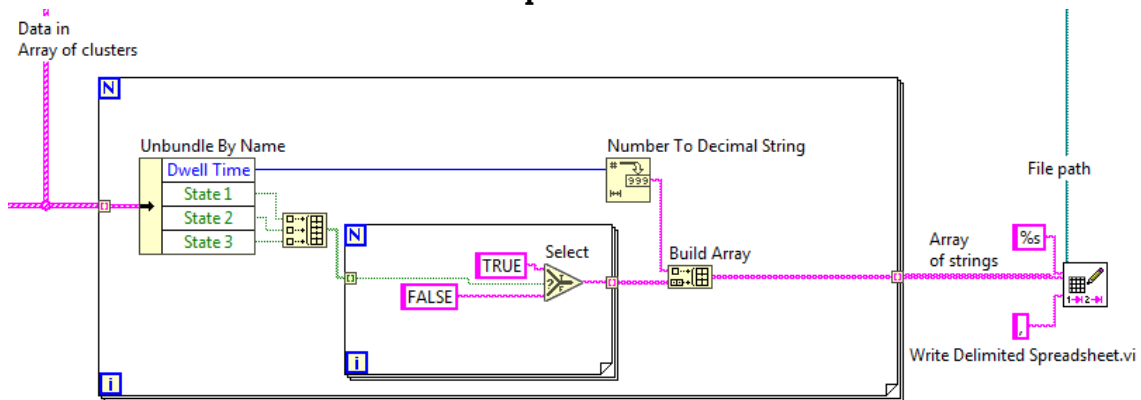
Zakład Systemów Informacyjno-Pomiarowych

IETiSIP, Wydział Elektryczny, PW



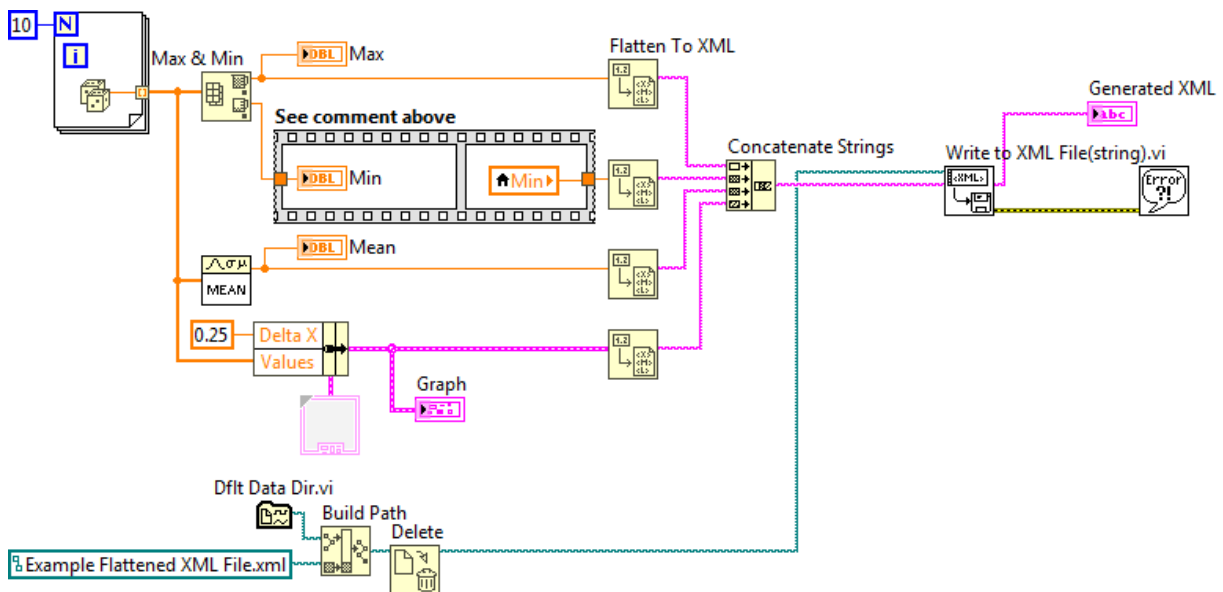


Rys. 6. Przykładowy fragment kodu ilustrujący użycie funkcji **Read Delimited Spreadsheet.vi**.

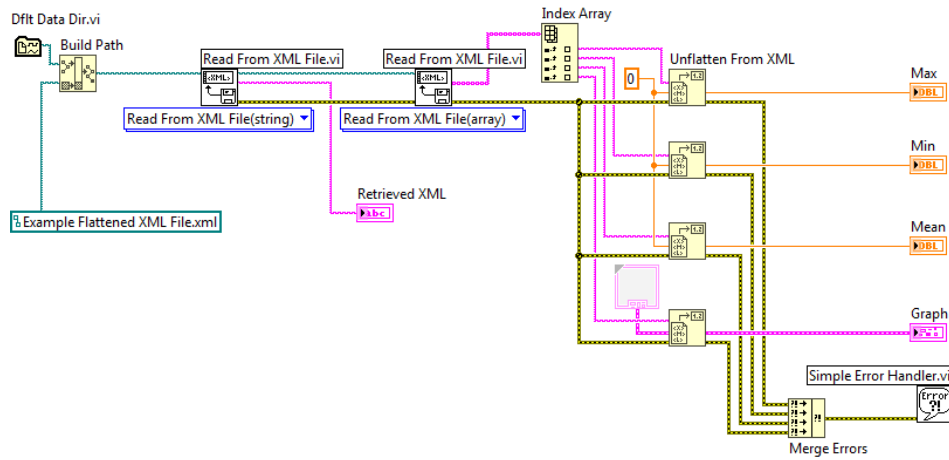


Rys. 7. Przykładowy fragment kodu ilustrujący użycie funkcji **Write Delimited Spreadsheet.vi**.

- plik XML



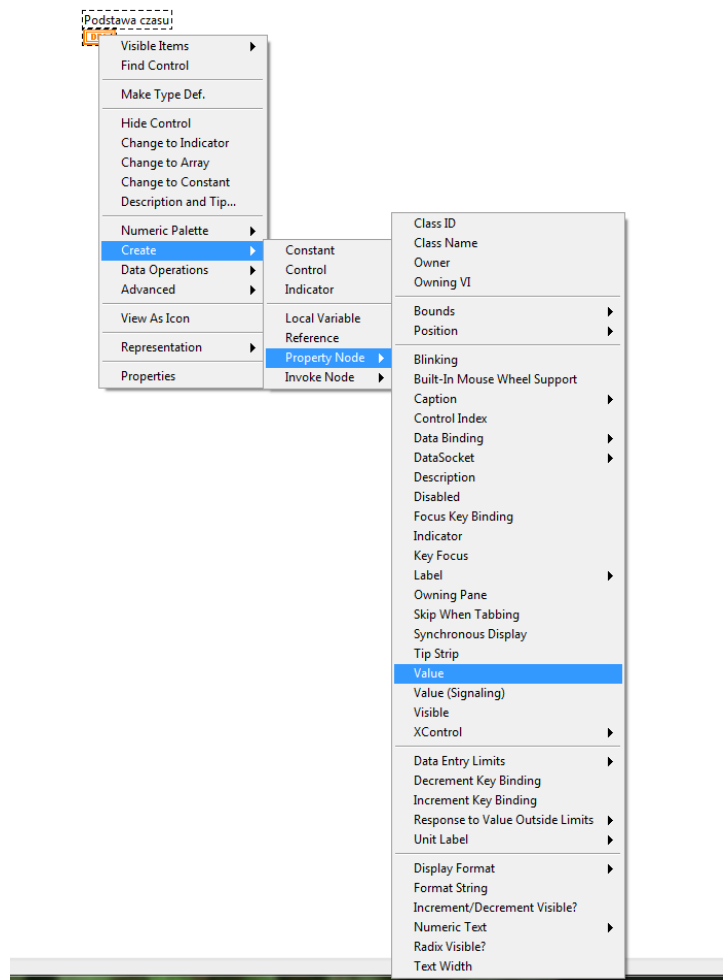
Rys. 8. Przykład zapisu danych do pliku XML. Wartość zmiennej Min otrzymuje stosowną etykietę, zaś Max i Mean - nie.



Rys. 9. Przykład odczytu danych z pliku XML.

Po odczytaniu danych z pliku należy uaktualnić nastawy na płycie czołowej. Do zmiany wartości nastaw i „synchronizacji” wartości w kodzie i na płycie czołowej można użyć **Property Node-> Value** albo **Value(signaling)** (dodatkowo generuje zdarzenie od zmiany wartości i w konsekwencji aktualizuje generowany przebieg).





Rys. 10. Lista rozwijana **Property Node**.



Zakład Systemów Informacyjno-
Pomiarowych

IETiSIP, Wydział Elektryczny, PW

