

# Zagadnienie 5

## INTERAKCJA CZŁOWIEK MASZYNA

### Programowanie w środowisku LabVIEW – podstawowa aplikacja SCADA.

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 struktury do przechowywania bieżącego stanu wewnętrznego aplikacji (urządzenia).



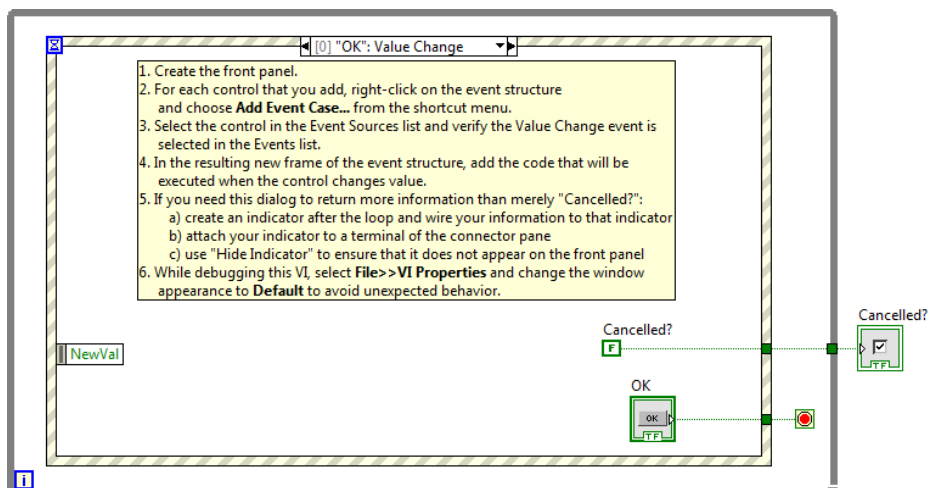
**Zakład Systemów Informacyjno-  
Pomiarowych**

**IETiSIP, Wydział Elektryczny, PW**

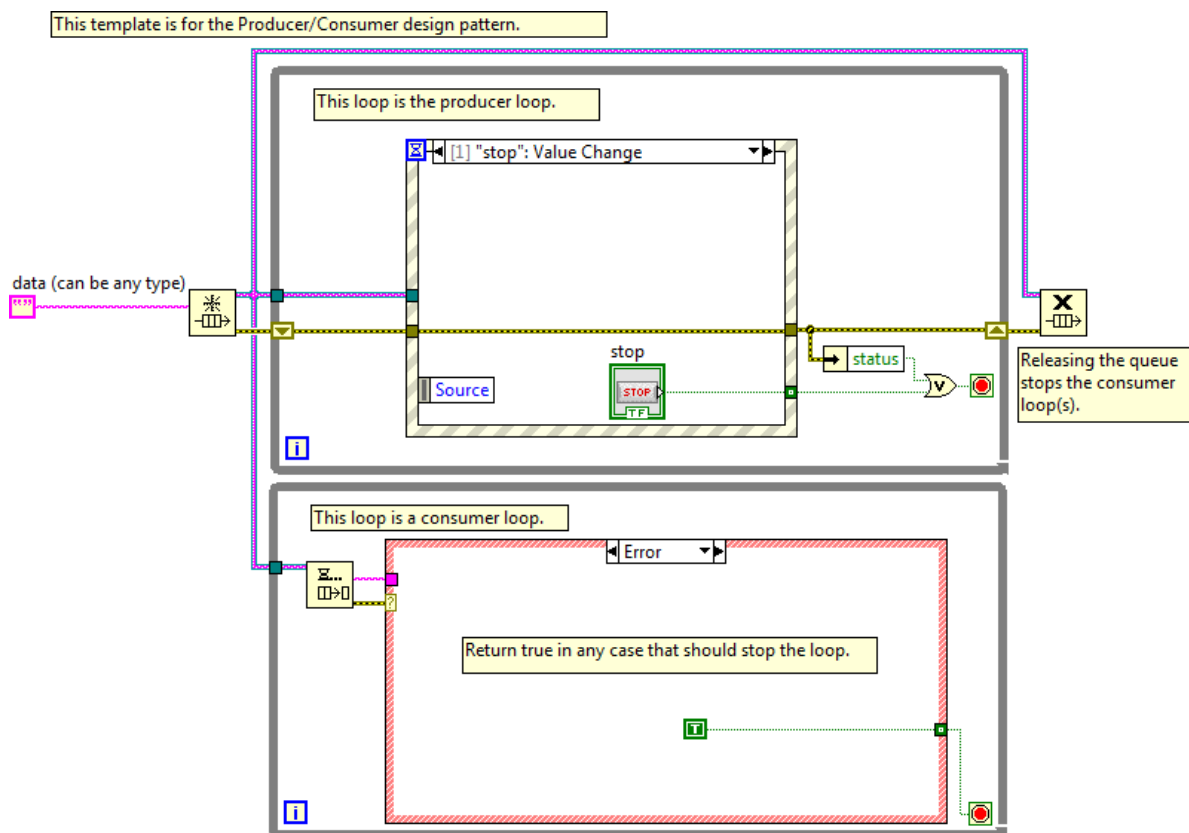


W materiałach do przedmiotu znajdują się przykładowe programy, które mogą okazać się pomocne.

1. Zagadnienie: Proszę zbudować aplikację postępując zgodnie z instrukcją zawartą w DSC\_getting\_startedAug07.pdf. Widok użytkownika/użytkowników oprzeć przynajmniej na **Event Structure**, a może nawet na Producencie/Konsumencie - rysunki 1 i 2.



Rys 1. Szablon **Event Structure** - domyślny szablon LabVIEW.



Rys 2. Szablon **Producer/Consumer** - domyślny szablon LabVIEW.

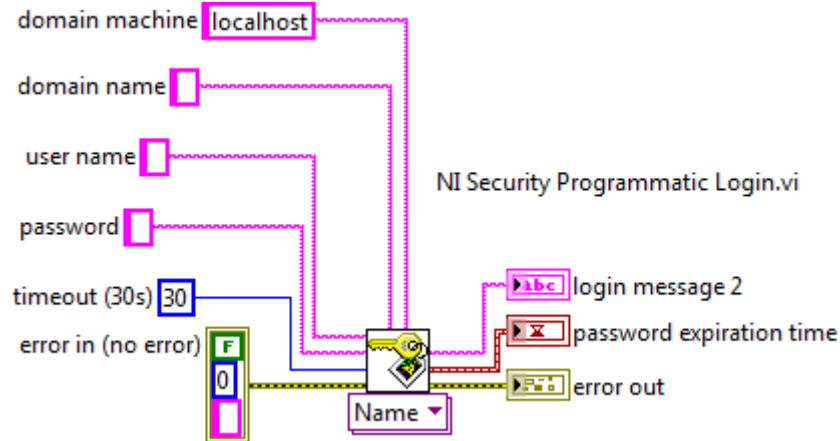


Zakład Systemów Informatycznych  
Pomiarowych

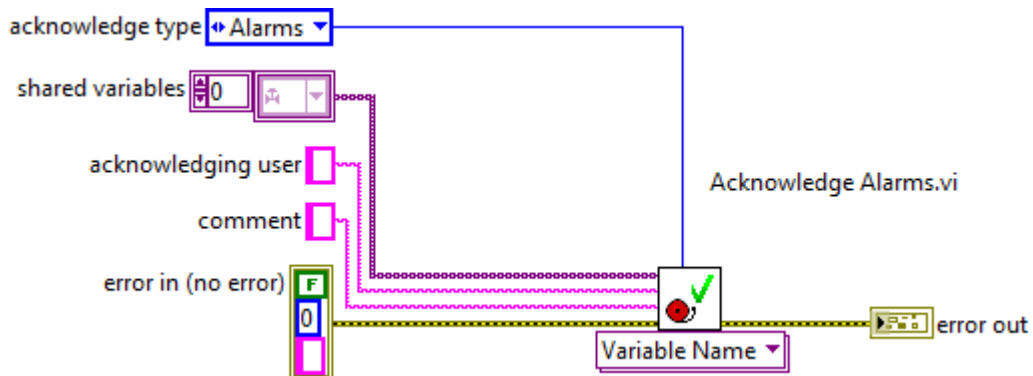
IETiSIP, Wydział Elektryczny, PW



Proszę zwrócić uwagę na ćwiczenia dodatkowe dotyczące definiowania praw dostępu i użytkowników, obsługi baz danych w zakresie podglądu alarmów i obserwacji danych historycznych. W tym celu mogą przydatne być funkcje z grup **Functions**-> DSC Module -> Security; **Functions** -> DSC Module -> Alarms & Events oraz **Controls** -> DSC Module -> Alarm Controls i **Controls** -> DSC Module -> Trend Controls.

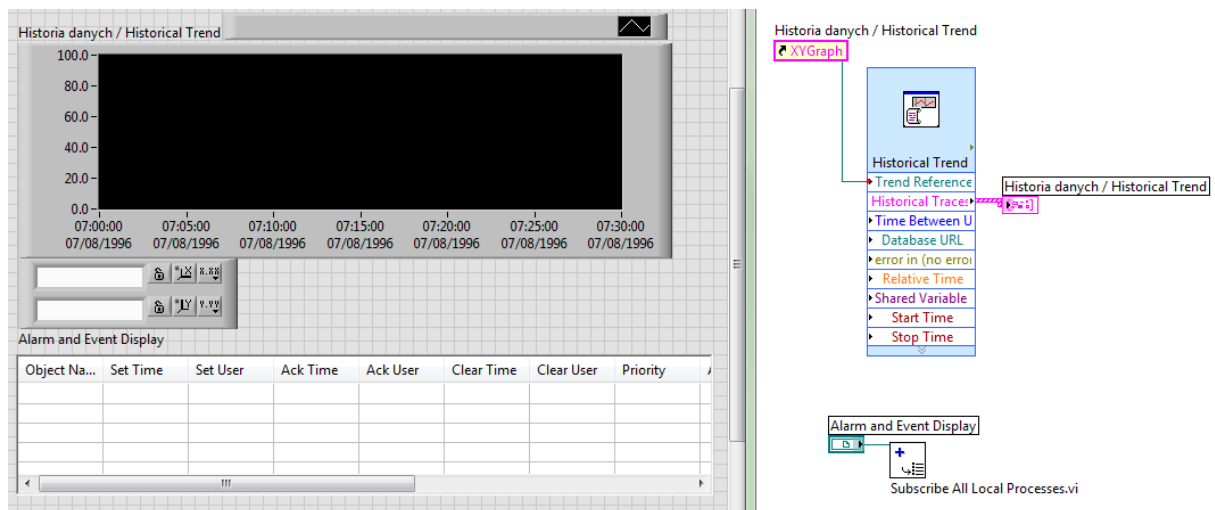


Rys 3. Funkcja **NI Security Programmatic Login.vi** - umożliwia zalogowanie użytkownika w trakcie działania programu.



Rys 4. Funkcja **Acknowledge Alarms.vi** - umożliwia potwierdzenie występującego alarmu.





Rys. 5. "Domyślne" fragmenty kodu (**Controls->NI DSC...**) do wizualizacji danych historycznych i podglądu alarmów.

Przeglądanie danych historycznych można wykonać za pomocą funkcji **Historical trend** (konfiguracja poprzez menu dla funkcji **Express** albo poprzez wejścia funkcji). Podgląd na danych najlepiej zrobić korzystając ze struktury **Event Structure** tak aby nie wysyłać redundantnych zapytań w pętli, a funkcję **Historical trend**.

UWAGA. Proszę zwrócić uwagę na fakt, że w plikach źródłowych serwera nie ma Event Structure (program/proces działa „w tle” i nie ujawnia interfejsu). Nie ma również enumeratorów, klastrów i zmiennych lokalnych/globalnych. Należy również unikać stosowania zmiennych lokalnych (local variables), węzłów własności (property nodes), węzłów wywołań (invoke nodes) i referencji (reference). Dodatkowo kod powinien być oparty na pojedynczej pętli **while** – jednym wątku (nie dwóch i więcej)

