



Fotografia 1. Komputery Panelowe National Instruments

HMI z użyciem NI LabView Touch Panel Module

W artykule opisano budowę typowego interfejsu HMI oraz pokazano, jak zaimplementować tę architekturę za pomocą biblioteki LabView Touch Panel Module.

Dodatkowe informacje:

National Instruments Poland
Salzburg Center
ul. Grójecka 5, 02-025 Warszawa
Tel.: 22 328 90 10, Faks: 22 331 96 40
ni.poland@ni.com, www.ni.com
Infolinia: 800 889 897
Wsparcie techniczne: techsupport@ni.com

W aplikacjach przemysłowych (i nie tylko), w których operator ma obsługiwać maszynę, konieczne jest przygotowanie mechanizmu, który umożliwiałby użytkownikowi konfigurację i monitorowanie stanu danej maszyny. Typowym rozwiązaniem spełniającym te wymagania i powszechnie stosowanym obecnie w przemyśle są interfejsy HMI – Human Machine Interface.

Do implementacji HMI świetnie nadają się komputery panelowe National Instruments, które zawierają niezawodny LCD z interfejsem dotykowym, zestaw zintegrowanych wejść i wyjść oraz są przystosowane do obsługi LabVIEW (graficzne środowisko programowania) i modułu LabVIEW Touch Panel. Niniejszy artykuł opisuje typową architekturę aplikacji HMI, stworzoną przy użyciu ww. oprogramowania.

Aby skrócić czas projektowania przy tworzeniu HMI, warto użyć dostępnych architektur referencyjnych i bibliotek przykładów.

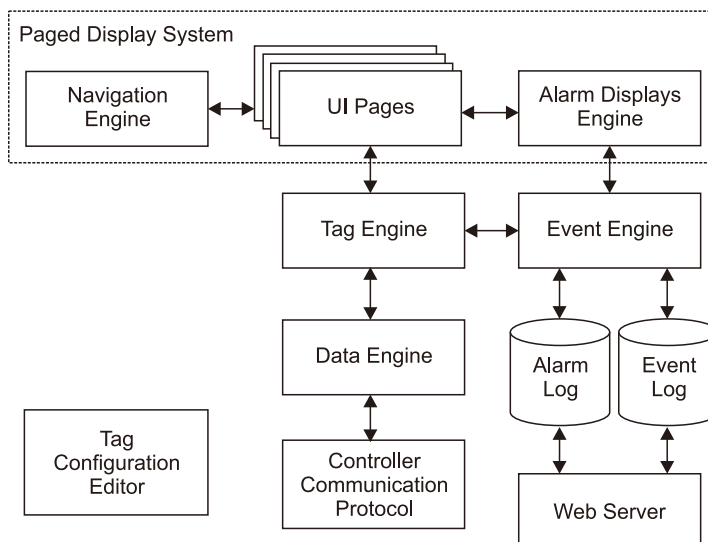
Typowa architektura HMI

Na rysunku 2 widać diagram typowej architektury HMI. Pokazuje on zależności pomiędzy poszczególnymi komponentami programowymi.

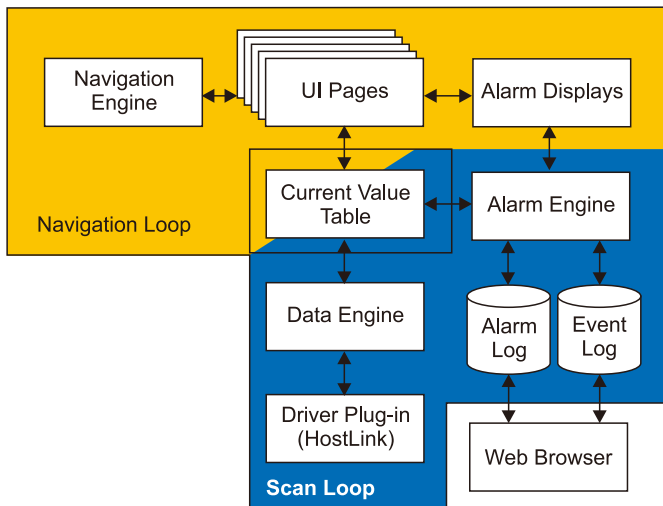
Ponieważ większość złożonych aplikacji posługuje się setkami zmiennych, trudno jest je wszystkie zaprezentować jednocześnie na jednym ekranie. Część zmiennych

odnosi się do wejść lub wyjść, część do stanu urządzenia, a część z nich to parametry kalibracyjne lub dane konfiguracyjne. Dobrze wykonany interfejs HMI powinien uwzględniać podział zmiennych na grupy, w zależności od tego, z czym są powiązane. Poszczególne grupy powinny być prezentowane na oddzielnych stronach, a poruszanie się pomiędzy stronami nimi powinno być zorganizowane za pomocą odpowiednich przycisków nawigacyjnych. Zmianę stron obsługuje w tle odpowiedni mechanizm (Navigation Engine).

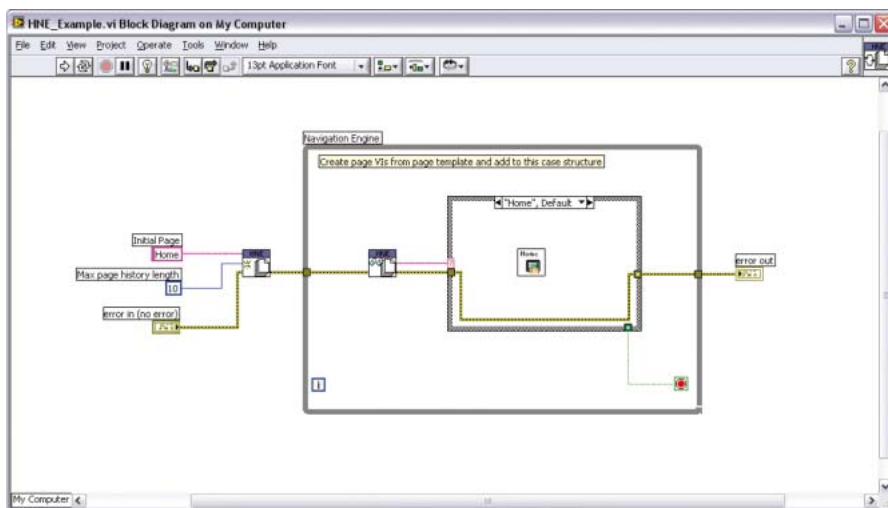
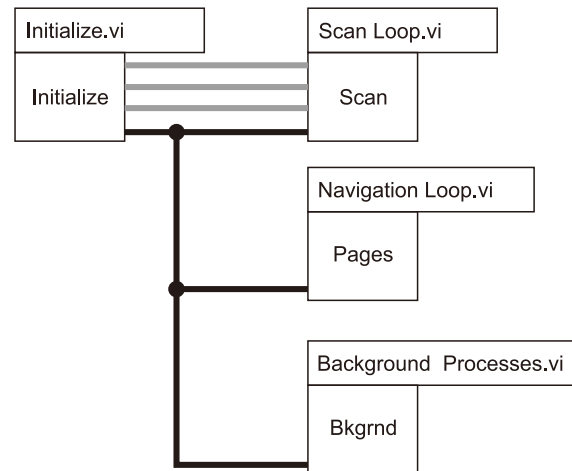
Do wyświetlania zwykłych wartości, takich jak liczby całkowite, ułamkowe i ciągi znaków można wykorzystywać standardowe wskaźniki. Jednakże, prezentacja alarmu wymaga dodatkowej obsługi. Zazwyczaj realizowane jest to przez zaprezentowanie podsumowania alarmów oraz stopki. Podsumowanie najczęściej wyświetlane jest na oddzielnej stronie, na której znajduje się pełna lista alarmów zaś stopka jest często prezentowana w postaci pojedynczej linii



Rysunek 2. Typowa architektura HMI



Rysunek 3. Typowa architektura HMI w LabVIEW



Rysunek 4. Przykład diagramu blokowego pętli nawigacyjnej

ki na dole ekranu, na której sygnalizowane są najnowsze alarmy. Opisane powyżej funkcje realizowane są przez blok Alarm Displays Engine.

Dostęp do danych, rejestru zdarzeń i alarmów może być również oferowany zdalnie za pomocą serwera WWW, poprzez przeglądarkę internetową.

Wprowadzanie danych

Problem w korzystaniu z wielu interfejsów HMI stanowi często wprowadzanie danych. Ponieważ częstokroć są one pozbawione myszki ani klawiatury, wpisywanie liczb czy ciągów znaków dokonywane jest za pomocą ekranu dotykowego, a w przypadku gdy urządzenie znajduje się na hali fabrycznej – również w rękawiczkach roboczych. Dlatego konieczne jest odpowiednie powiększenie wszelkich pól do wprowadzania danych, a wyświetlane klawiatury muszą być dostosowane do rodzaju wpisywanych informacji. Przykładowo, jeśli wybrane pole przyjmuje jedynie liczby, warto wyświetlić jedynie odpowiednio dużą klawiaturę numeryczną.

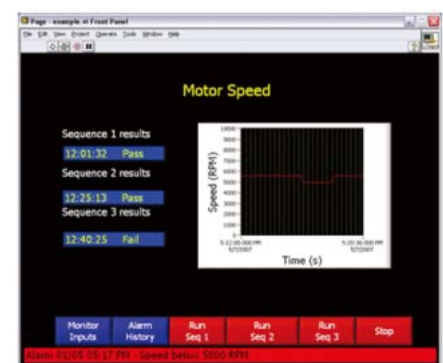
Warto Należy też zwrócić uwagę na to, jak zmienne są przechowywane i udostęp-

niane. Obsługą tych zadań zajmuje się odpowiedni mechanizm (Tag Engine), którego algorytm działania wygląda następująco:

- Mechanizm prezentacji stron odczytuje wartości zmiennych w celu ich wyświetlenia.
- Operator korzysta z obiektów do wprowadzania danych, aby zmienić wartości zmiennych, które następnie są przekazywane do reszty systemu.
- Mechanizm obsługi zdarzeń monitoruje wartości zmiennych sprawdzając, czy nie spełniają one określonych wcześniej warunków, które mogłyby wywołać alarm lub wymusić inne działanie.
- Mechanizm obsługi danych przekazuje nowe wartości zmiennych do sterownika maszyny.

Mechanizm zdarzeń

Mechanizm ten odpowiada za porównywanie wybranych grup zmiennych z wcześniej zdefiniowanymi warunkami. Te warunki mogą być określone jako przynależność do zbioru, równowartość itp. Jeśli których z warunków zostanie spełniony, mechanizm rejestruje tę sytuację. Niektó-



Rysunek 5. Przykładowa strona HMI (panel frontowy wirtualnego instrumentu LabVIEW)

re ze zdarzeń mogą wymagać interwencji i najczęściej wywołują alarmy. Dane związane z konkretnym alarmem są przesyłane do mechanizmu wyświetlania alarmów, który określa sposób, w jaki zostaną one zaprezentowane operatorowi. Jeśli wartość zmiennej wróci do dozwolonego zakresu, alarm jest anulowany.

Mechanizm obsługi danych pozwala na wymianę wartości zmiennych z kontrolerem za pomocą odpowiedniego, wybranego protokołu. W praktyce, sterowniki PLC najczęściej korzystają z własnościowych, zamkniętych protokołów lub powszechnych w przemyśle, takich jak np. Modbus. Coraz częściej stosuje się komunikację opartą o protokoły bazujące na Ethernetie.

Narzędzie do edytowania zmiennych pozwala na tworzenie, konfigurowanie i zarządzanie listą tagów aplikacji. W wyniku jego działania można wyeksportować plik konfiguracyjny, który pozwala interfejsowi HMI na zainicjalizowanie mechanizmu obsługi zmiennych.

Implementacja typowej architektury HMI

Moduł LabVIEW Touch Panel pozwala na tworzenie aplikacji HMI w środowisku graficznym pracującym pod kontrolą sys-

temu operacyjnego Windows. Gotowy projekt może być w łatwy sposób przeniesiony i uruchomiony na komputerze panelowym NI. Dla uproszczenia, typową strukturę aplikacji HMI można podzielić na trzy pętle programowe: nawigacji, skanowania i procesów tła. Zostały one przedstawione na **rysunku 3**.

Pętla nawigacyjna zawiera obsługę stron interfejsu HMI i jest odpowiedzialna za ich organizację oraz nawigację pomiędzy nimi. Pętla skanowania zawiera mechanizmy danych, alarmów i sprzętowe sterowniki interfejsów komunikacyjnych. Jest odpowiedzialna za wymianę danych pomiędzy HMI i kontrolerem maszyny oraz za sprawdzanie czy wystąpiły jakieś alarmy. Pętla procesów tła zawiera wszystkie pozostałe komponenty działające równoległe w tle, potrzebne do poprawnego działania aplikacji.

Inne komponenty, które mogą być ewentualnie zastosowane w aplikacji HMI, nie współdziałają bezpośrednio z aplikacją. Przykładowo, zarówno serwer WWW, jak i edytor konfiguracji zmiennych wymieniają dane z aplikacją za pomocą systemu plików – odpowiednio stron internetowych oraz plików konfiguracyjnych.

Pętla nawigacyjna

Mechanizm ten jest najczęściej implementowany za pomocą prostej maszyny stanów w zapętłonej procedurze z szeregiem instrukcji warunkowych. Każda z takich instrukcji obejmuje jedną stronę, wyświetlaną na ekranie HMI po jej wywołaniu. Przykład pętli nawigacyjnej widać na **rysunku 4**. Zastosowano w nim bibliotekę referencyjną HNE (HMI Navigation Engine), zaprojektowaną specjalnie do zarządzania stronami interfejsów HMI i nawigacją.

Dokładna dokumentacja HNE znajduje się w internecie, pod adresem: <http://zone.ni.com/devzone/cda/epd/p/id/5329>

Każda ze stron HMI jest tworzona jako instrument wirtualny w LabVIEW i pozwala monitorować oraz kontrolować konkretny proces lub subprocess zachodzący w maszynie. Najbardziej powszechnie stosowane elementy znajdujące się na takich stronach to przyciski nawigacyjne, wskaźniki numeryczne, wykresy, obrazki oraz kontrolki dwustanowe. Dobrym przykładem takiej strony jest **rysunek 5**.

Na schemacie blokowym strony stosuje się mechanizmy bazujące na obsłudze zdarzeń, co jest możliwe dzięki użyciu referencyjnej biblioteki AMC (Asynchronous Message Communication). Pozwala ona uzyskać responsywny interfejs użytkownika. Przykładowy schemat blokowy opracowany z użyciem metod bazujących na bibliotece AMC pokazano na **rysunku 6**.

Prezentacja alarmów

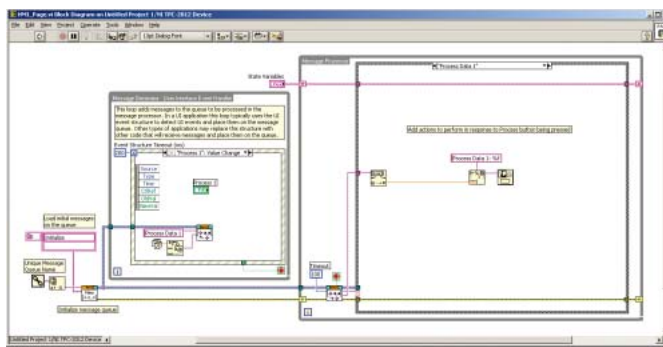
Na **rys. 3** obsługa wyświetlania alarmów została ujęta w pętli nawigacyjnej. Zrobiono tak, ponieważ bardzo często alarmy prezentowane są po prostu w postaci specjalnej strony interfejsu HMI, której działanie polega na wyświetlaniu informacji o najnowszych wydarzeniach zachodzących w maszynie. Innym popularnym sposobem prezentacji alarmów jest wyświetlanie jednolitej stopki, która w razie potrzeby pojawia się na dowolnej ze stron interfejsu HMI. Firma National Instruments przygotowała specjalną bibliotekę referencyjną do obsługi alarmów, która pozwala na ich wygodne monitorowanie, wyświetlanie i rejestrowanie. Biblioteka ta to TAE (Touch Panel Alarm Engine) i zawiera interfejs programistyczny pozwalający na dostęp do danych o alarmach, ich formatowanie i prezentację na stronach HMI. Szczegółowe informacje o jej działaniu można znaleźć w Internecie pod adresem: <http://zone.ni.com/devzone/cda/epd/p/id/5332>

Pętla skanująca

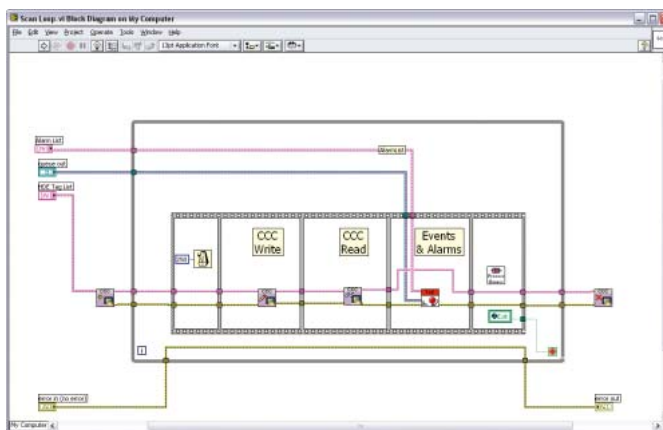
Ma ona postać sekwencji procesów, które wykonywane są w pętli z określoną częstotliwością. Mogą do nich należeć także procesy obsługujące dane i alarmy. Przykładową pętlę skanującą zaprezentowano na **rysunku 7**. Została ona zbudowana z użyciem biblioteki referencyjnej CCC (CVT Client Communication), która pozwala na uruchomienie mechanizmu obsługi danych i zawiera odniesienia do biblioteki TAE w celu obsługi alarmów.

Biblioteka CCC ułatwia wymianę zmiennych pomiędzy interfejsem HMI i sterownikiem maszyny. CCC pobiera dane z kontrolera i aktualizuje je w pamięci HMI, dzięki czemu są one zawsze aktualne i dostępne lokalnie. Sposób zaimplementowania pętli CCC w pętli skanującej HMI został zilustrowany na **rysunku 8**.

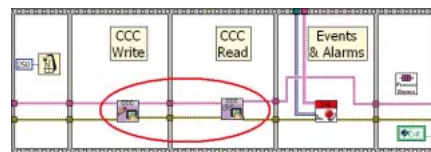
Mechanizm obsługi alarmów działa zgodnie z opisanymi wcześniej zasadami. Zastosowanie biblioteki TAE pozwala łatwo



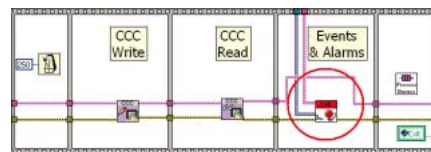
Rysunek 6. Przykładowy diagram blokowy wykonany z użyciem biblioteki AMC



Rysunek 7. Przykład diagramu blokowego pętli skanującej



Rysunek 8. Przyrządy wirtualne CCC w pętli skanującej aplikacji HMI



Rysunek 9. Miejsce implementacji mechanizmu alarmów w pętli skanującej

prezentować nowe alarmy, klasyfikować je jako aktywne, a następnie zapisywać na dysku. Mechanizm ten powinien być uruchamiany po aktualizacji wartości wszystkich zmiennych. Zilustrowano to na **rysunku 9**.

Tablica aktualnych wartości

Mechanizm wymiany zmiennych pomiędzy poszczególnymi elementami zaimplementowany jest za pomocą biblioteki referencyjnej CVT (Current Value Table). Tworzy ona centralne miejsce, w którym gromadzone są wszystkie zmienne i zapewnia interfejs programistyczny, umożliwiający łatwy dostęp do nich. Sposób jej działania został opisany w internecie pod adresem: <http://zone.ni.com/devzone/cda/epd/p/id/5326>