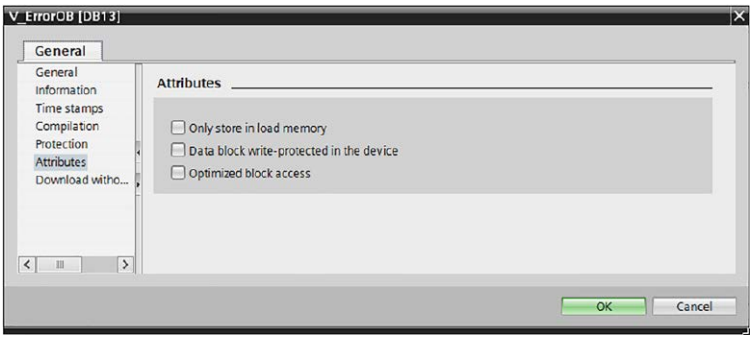
Programowanie paneli HMI (4)

Dobrze napisany program powinien się charakteryzować działaniem zgodnym z założeniami, ale także jak największą odpornością na błędy. Jednak sytuacje awaryjne zawsze będą się pojawiały podczas działania maszyny. Ważne jest, aby w prosty i jednoznaczny sposób poinformować dział utrzymania ruchu, co się aktualnie dzieje z maszyną. Do realizacji tego celu panele operatorskie HMI przewidują funkcjonalność zwaną alarmami. W ten sposób możemy wyświetlić zdarzenia i stany pracy.

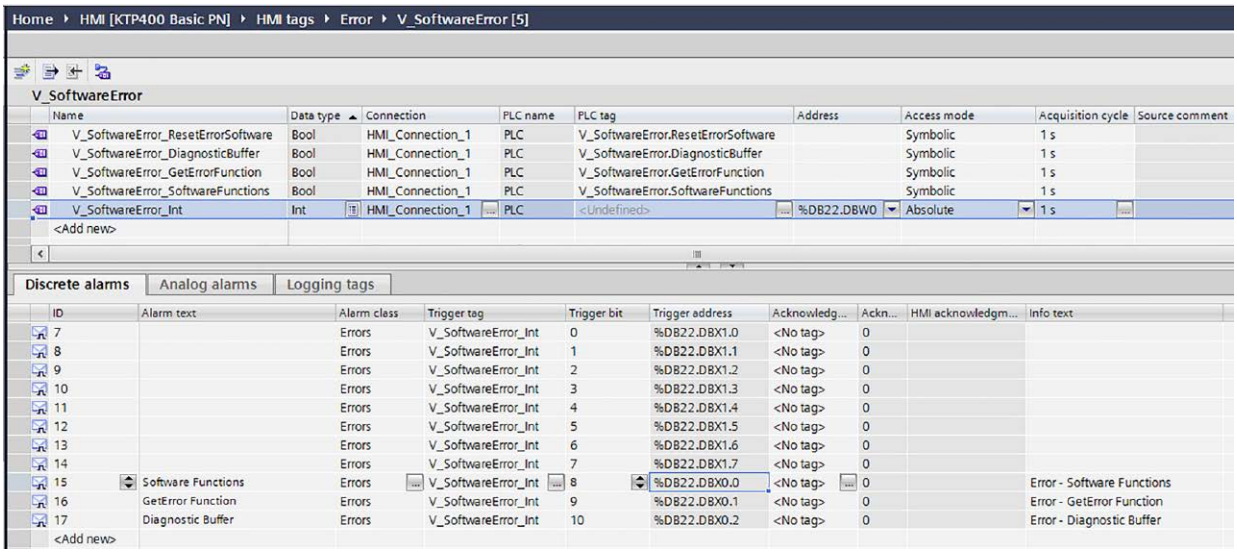


Rysunek 1. Odznaczenie opcji *Optimized block access* w *Attributes*

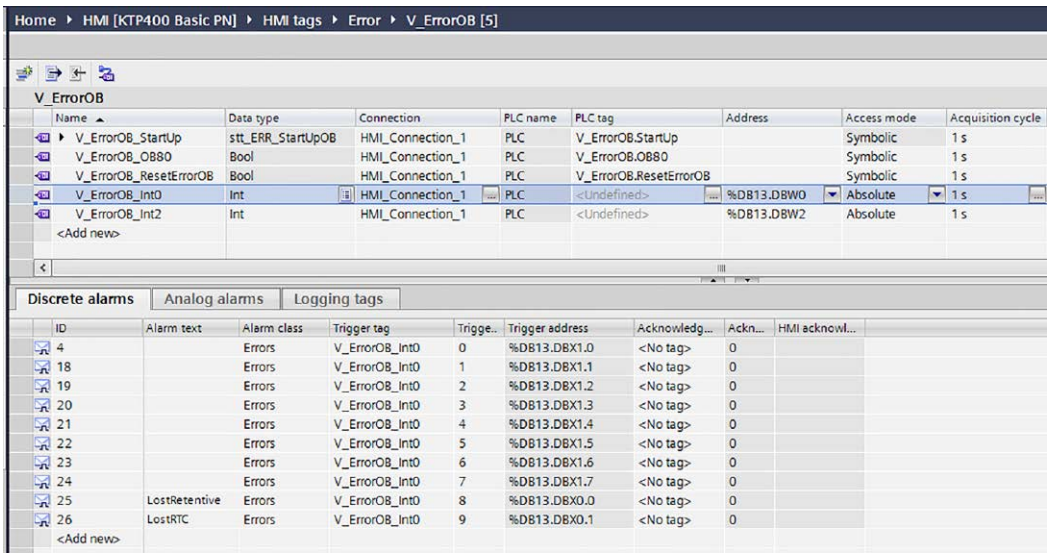
Alarmy można podzielić na dwie grupy:

1. Dyskretne, w których zmiana stanu sygnału cyfrowego może powodować alarm.
2. Analogowe, które wskazują, czy wartość zmiennej (np. typu *Int*) przekroczyła dopuszczalny limit (górny, dolny).

Zaczniemy od przygotowania zmiennych do wyświetlenia. W projekcie dla PLC należy przejść do folderu *01Visualization*, a następnie *Error*. Znajdują się tam dwa bloki danych. Dla każdego z nich należy wyłączyć



Rysunek 2. Dodanie do tablicy HMI tags o nazwie *V_SoftwareError* tagu *V_SoftwareError_Int* typu *Int*



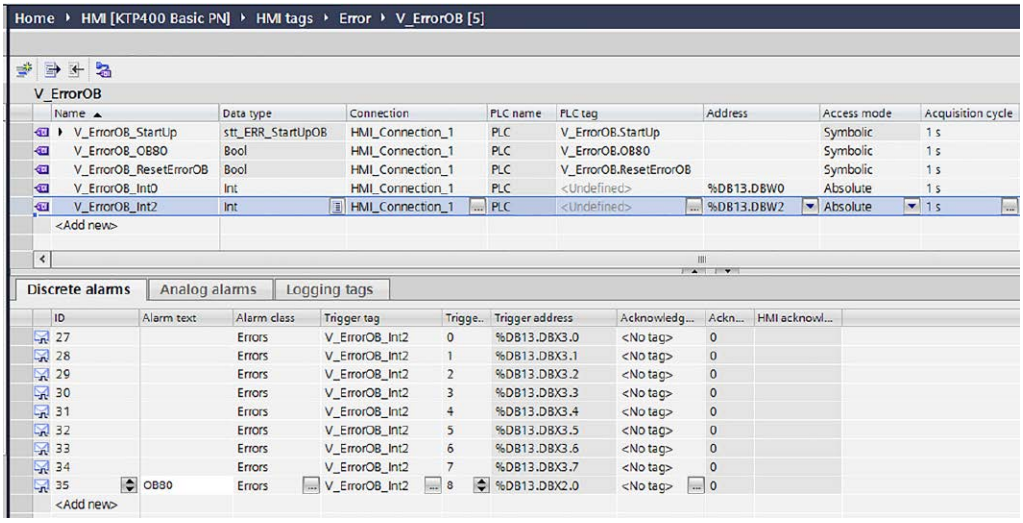
Rysunek 3. Dodanie zmiennej *V_ErrorOB_Int0* do tablicy *V_ErrorOB*

optymalizację poprzez wejście do właściwości. Przedstawia to **rysunek 1**. Należy odznaczyć w *Attributes* opcję *Optimized block access*. Taka konfiguracja wynika z faktu, że wyświetlenie alarmu na panelu HMI dla zmiennej typu Bool jest możliwe, gdy posiada ona fizyczny adres. Jedynie zmienne typu Int można wyświetlić jako alarm. Wówczas nie trzeba wyłączać optymalizacji.

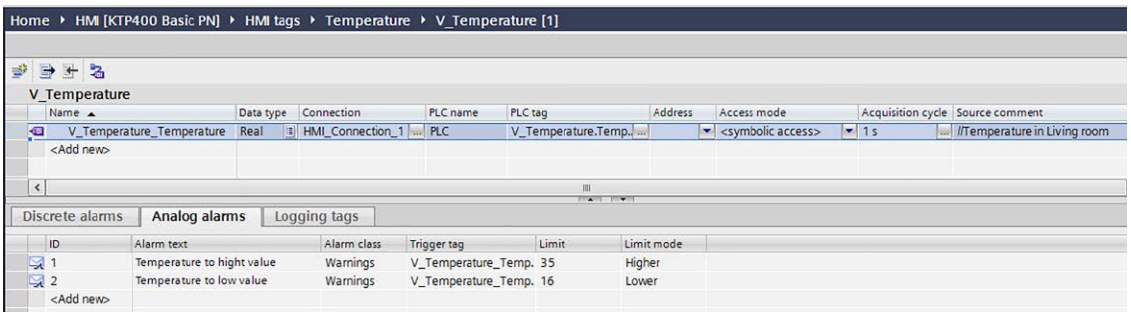
Alarm analogowy zostanie pokazany na przykładzie temperatury. W folderze *01Visualization* znajduje się folder *Temperature*. Jest

tam blok o nazwie *V_Temperature*. Podobnie jak poprzednio, w HMI tags tworzymy nowy folder o nazwie *Temperature* oraz tablicę o nazwie *V_Temperature*. Należy skopiować do tej tablicy zmienną z bloku danych *V_Temperature*.

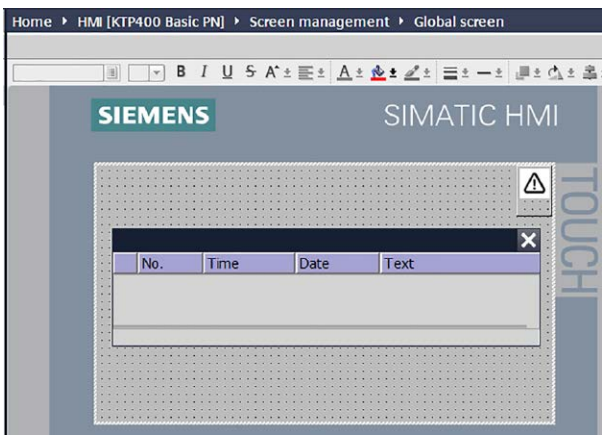
Przed przystąpieniem do pracy należy przygotujemy sobie *HMI tags*. Wcześniej wspomniane bloki danych, czyli *V_ErrorOB* oraz *V_SoftwareError*, a więc należy utworzyć sobie dwie tablice o takich samych nazwach. Wszystkie zmienne znajdujące się w tych blokach danych należy odpowiednio skopiować do utworzonych tablic *HMI tags*.



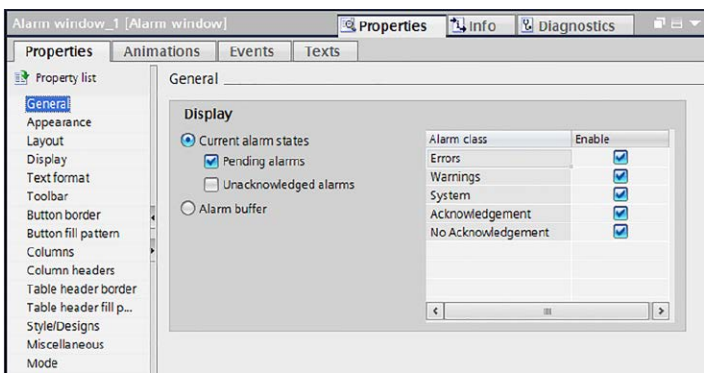
Rysunek 4. Dodanie zmiennej *V_ErrorOB_Int2* do tablicy *V_ErrorOB*



Rysunek 5. Konfiguracja alarmu analogowego



Rysunek 6. Wygląd okna *Global screen*



Rysunek 7. Konfiguracja okna alarmu

Konfiguracja zmiennych do alarmów

Do tablicy HMI tags o nazwie *V_SoftwareError* należy dodać jeszcze jeden tag np. o nazwie *V_SoftwareError_Int* typu *Int* co przedstawia **rysunek 2**. W kolumnie *Access mode* należy wybrać *Absolute*. Zmiennej został przypisany adres fizyczny %DB22.DBW0. Więc ta zmienna zawiera wszystkie cztery zmienne znajdujące się w bloku danych *V_SoftwareError*. Takie rozwiązanie jest konieczne, ponieważ tylko zmiennym typu Int lub Word można przypisać alarmy.

Poniżej w zakładce *Discrete alarm* widzisz zmienną *V_SoftwareError_Int* podzieloną na poszczególne bity. Błąd *Software Functions* posiada adres fizyczny %DB22.DBX0.0. Zmiana wartości z 0 na 1 na tym adresie spowoduje pojawienie się komunikatu o błędzie o treści, która została umieszczona w kolumnie *Alarm text*.

W kolumnie *Info text* możesz dopisać dodatkowe informacje/komentarze o błędzie.

Analogicznie do tablicy *V_ErrorOB* dodajemy kolejne zmienne o nazwach *V_ErrorOB_Int0* oraz *V_ErrorOB_Int2*. Przedstawiają to **rysunek 3** oraz **rysunek 4**. Na poszczególnych bitach tych zmiennych zostały ustawione alarmy.

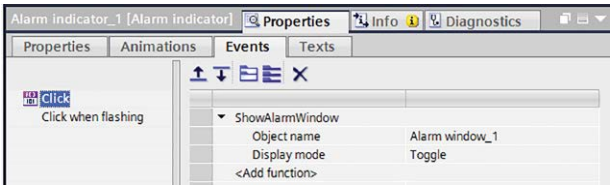
W tym przypadku wszystkie błędy zostały dodane do klasy *Error* (kolumna *Alarm class*), jednak można też w tej kolumnie wybrać *Warning* lub alarm z potwierdzeniem lub bez potwierdzenia.

W przypadku alarmu analogowego wykorzystamy temperaturę. Konfiguracja została przedstawiona na **rysunku 5**. W zakładce *Analog alarms* zostały skonfigurowane dwa alarmy w zależności od temperatury. Po przekroczeniu wartości 35° pojawi się błąd przekroczenia górnej granicy, natomiast po spadku temperatury poniżej wartości 16°, to zostanie wyświetlony alarm z powodu przekroczenia dolnej granicy.

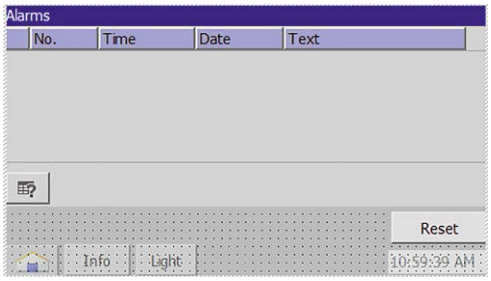
Ekrany

Alarmy musimy gdzieś wyświetlić, zatem będą potrzebne nam dodatkowe ekrany. Zaczniemy od przygotowania globalnego szablonu, aby pojawienie się błędu powodowało automatyczne wyświetlenie komunikatu. Dodatkowo użyjemy obiektu, który będzie wyświetlał liczbę aktualnych błędów.

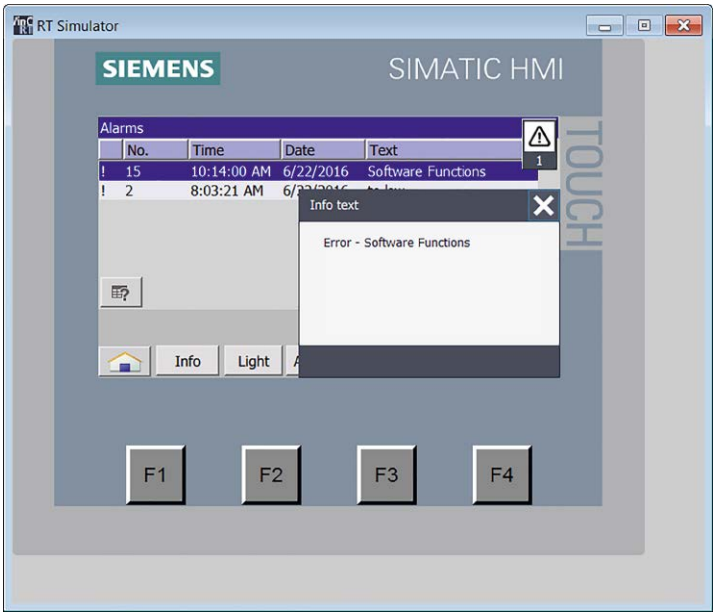
Przejdźmy do *Screen management* i następnie *Global screen*. Widok tego okna został przedstawiony na rysunku 6. Zaczniemy od dodania obiektu znajdującego się na środku tego ekranu. Z prawej



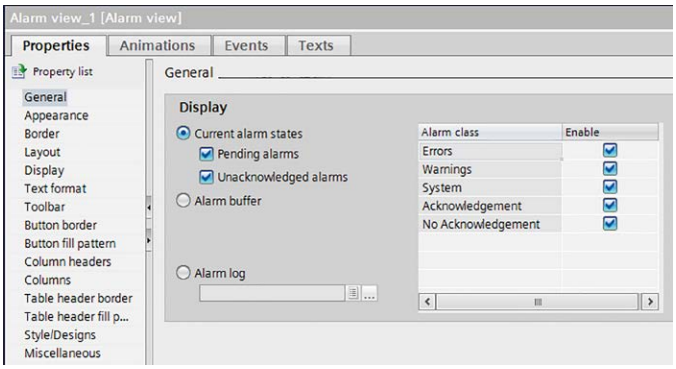
Rysunek 8. Konfiguracja zdarzenia aktywowanego po wybraniu *Alarm indicator*



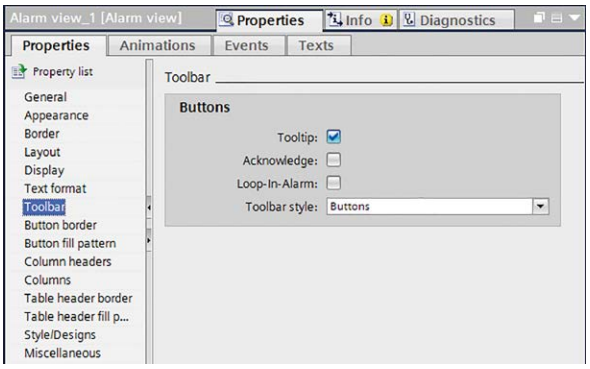
Rysunek 9. Dodanie w folderze *Screens* ekranu o nazwie *Alarms*



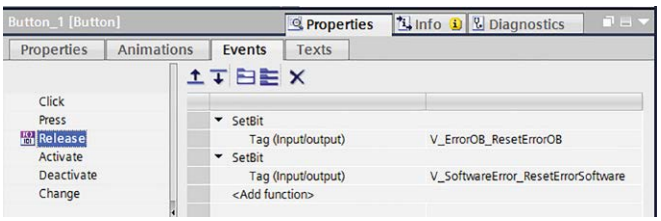
Rysunek 13. Przykładowy widok błędu



Rysunek 10. Konfiguracja obiektu *Alarm view* – właściwości wyświetlania



Rysunek 11. Konfiguracja obiektu *Alarm view* – ustalenie wyglądu



Rysunek 12. Przypisanie zdarzeń do przycisku *Reset*

strony w karcie *Toolbox* znajduje się obiekt *Alarm window*. Należy przeciągnąć ten obiekt i umieścić na ekranie *Global screen* w podobny sposób, jak na rysunku 5. Ten obiekt posłuży do wyświetlania pojawiających się zdarzeń. Trzeba przeprowadzić tylko konfigurację tego obiektu, co przedstawia **rysunek 7**.

W zakładce *General* znajduje się pole *Display*. Zaznaczamy opcję *Pending alarms* i zaznaczamy wszystkie klasy. Pozostałe ustawienia domyślne są poprawne do naszego zastosowania.

W prawym górnym rogu umieszczamy obiekt *Alarm indicator*, który będzie wyświetlał liczbę błędów. We właściwościach dla tego obiektu zaznaczamy wszystkie klasy. Ustawimy też dla tego obiektu zdarzenie, które uaktywni się po naciśnięciu *Alarm indicator*. Konfiguracja została przedstawiona na **rysunku 8**. Ten obiekt posiada tylko jedno zdarzenie, które należy powiązać z obiektem *Alarm window*.

W folderze Screens dodać należy kolejny ekran o nazwie *Alarms*, co przedstawia **rysunek 9**. Standardowo na samej górze dodajemy nagłówek, co należy wykonać analogicznie jak w poprzednich odcinkach cyklu.

Następnie z karty Toolbox dodajemy obiekt Alarm view. Służy on także do wyświetlania alarmów, podobnie jak *Alarm window*. Konfiguracja tego obiektu została przedstawiona na **rysunku 10** oraz **rysunku 11**.

Rysunek 10 przedstawia sposób, w który konfigurujemy alarmy, jakie mają być wyświetlane w *Alarm view*. Zaznaczamy *Pending alarms* oraz *Unacknowledged alarms*, jak również wszystkie klasy alarmów.

Rysunek 11 przedstawia konfiguracje zakładki Toolbar. Zaznaczamy w polu *Buttons* „ptaszek" przy *Tooltip*. To spowoduje dodanie przycisku na obiekcie *Alarm view*. Naciśnięci tego przycisku spowoduje pojawienie się dodatkowego tekstu, który znajduje się w kolumnie *Info text*.

Pozostało nam dodanie jeszcze przycisku *Reset*, aby kasować pojawiające się błędy. Do tego przycisku należy przypisać zdarzenia, co przedstawia **rysunek 12**. Do zdarzenia *Relese* dodajemy funkcje *SetBit*, aby ustawić tagi odpowiedzialne za reset błędów po stronie kodu w sterowniku PLC.

Pozostało tylko do szablony MyTemplete dodać przycisk Alarms i przypisać do niego zdarzenie aktywacji ekranu *Alarms* po naciśnięciu tego przycisku.

W ten sposób została przygotowana część związana z wyświetlaniem alarmów i ewentualnie innych zdarzeń. W jednym z następnych odcinków pokaże, jak zrobić rejestrację tych zdarzeń w postaci pliku tekstowego.

Można teraz wykonać kompilację projektu i jego uruchomienie. Następnie po stronie kodu w sterowniku PLC trzeba zasymulować pojawienie się błędu. Można do tego celu użyć tablic monitorujących. Przykładowy widok błędu przedstawia **rysunek 13**.

Tomasz Gilewski
tomasz.gilewski@mistrzplc.pl
www.mistrzplc.pl