

Programowanie paneli HMI (4)

Przyszedł czas na zajęcie się pracą z ekranami. Jest to podstawowa umiejętność, ponieważ ogólna idea wizualizacji polega na pokazywaniu odpowiednich wartości oraz ich zmian. To właśnie na ekranach umieszcza się elementy biblioteczne, aby przedstawić operatorowi i obsłudze informacje o danym składniku procesu.

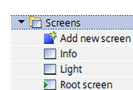
Pierwszy krok to dodanie do projektu nowych ekranów, na których zostanie zbudowana wizualizacja o aktualnym czasie i dacie oraz temperaturze otoczenia w mieszkaniu. Te informacje zostaną umieszczone na ekranie o nazwie *Info*. Kolejny ekran, który będzie nam potrzebny, to *Light*. Posłuży on do wyświetlenia informacji o aktualnym stanie sygnałów wejściowych w sterowniku oraz umieszczenia przycisków do włączenia/wyłączenia oświetlenia w każdym z pomieszczeń.

Z drzewa projektu rozwijamy *Screens* i wykorzystujemy *Add new screen*. Dodajemy dwa ekrany i zmieniamy im nazwy zgodnie z tym, co zostało napisane wcześniej. W ten sposób w projekcie mamy aktualnie trzy ekrany, co pokazano na **rysunku 1**. Ekran startowy to *Root screen*, co jest symbolicznie zaznaczone zielonym trójkątem. Oczywiście istnieje możliwość zmiany. Każdy ekran w projekcie można ustawić, aby był startowym.

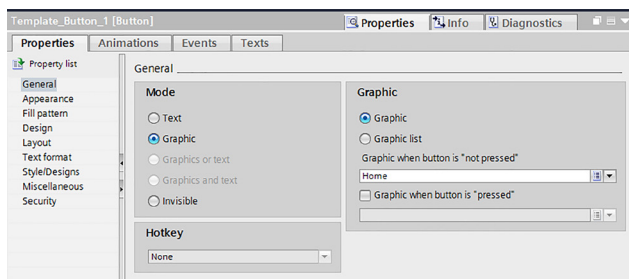
Zacniemy od utworzenia szablonu (tz. *template*), który będzie dostępny na każdym z ekranów. Zostaną tam umieszczone obiekty, które będą potrzebne do przechodzenia pomiędzy ekranami. Jednym słowem zrobimy menu. Z drzewa projektu wybieramy *Screen management* i następnie *Templates*. Wybieramy *Add new template*. Następnie zmieniamy nazwę na *MyTemplate*. Z prawej strony z karty *Toolbox* dodajemy z zakładki *Elements* obiekt *Button*. Umieszczamy go w lewym dolnym narożniku. Przechodzimy do właściwości tego obiektu poprzez jego zaznaczenie. Przedstawia to **rysunek 2**.

Przechodzimy do zakładki *Properties* w zakładce *Properties*. Następnie w *General* z pola *Mode* wybieramy *Graphic* i następnie w polu *Graphic* wybieramy z listy rozwijanej ikonę *Home*. Spowoduje to, że zamiast tekstu na przycisku zostanie umieszczona grafika.

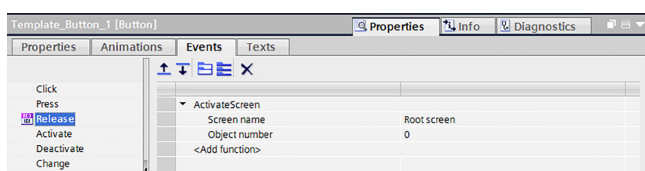
Kolejna czynność, którą należy zrobić dla tego przycisku, to utworzenie zdarzenia. Będzie ono polegało na tym, że po naciśnięciu tego przycisku nastąpi zmiana ekranu na ekran startowy (*Root screen*). Aby utworzyć tę funkcjonalność, trzeba wybrać zakładkę *Event*, co przedstawia **rysunek 3**. Po lewej stronie są dostępne zdarzenia, które są powiązane z przyciskiem. Skorzystamy z *Release*, czyli dopiero



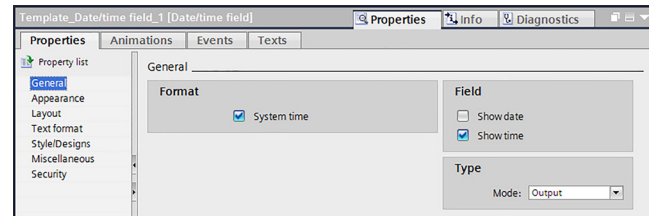
Rysunek 1. Trzy ekrany umieszczone w projekcie



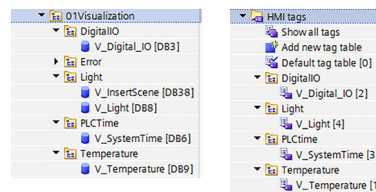
Rysunek 2. Właściwości obiektu Button



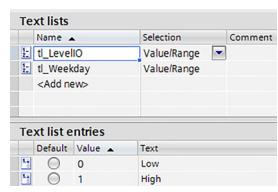
Rysunek 3. Wybranie zakładki Event



Rysunek 4. Konfiguracja obiektu Data fields



Rysunek 5. Bloki danych używane w tej części kursu



Rysunek 6. Utworzenie nowych grup danych

Rysunek 7. Powiązanie poziomów logicznych z opisami

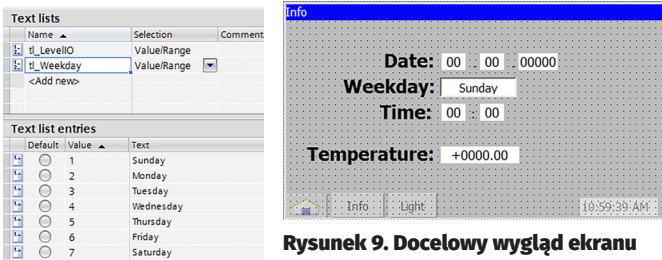
w momencie zwolnienia przycisku nastąpi wykonanie akcji, czyli zmiana ekranu. Akcje wybieramy z listy *<Add function>* lub można też wpisać ręcznie, jeżeli znamy jej nazwę. Nas interesuje aktywizowanie określonego ekranu, dlatego wykorzystamy *ActivateScreen*. Pozostało w polu *Screen name* wybrać także z rozwijanej listy nazwę ekranu. Wybieramy *Root screen*. Analogicznie dodajemy do szablonu dwa przyciski, którym należy przypisać odpowiednie nazwy *Info* oraz *Light*. Pozostało jeszcze dla każdego z nich utworzyć zdarzenia, aby po naciśnięciu nastąpiło przejście do ekranu, który wskazuje nazwa przycisku.

W prawym dolnym rogu umieścimy obiekt *Data/Time field* z zakładki *Elements*. Konfiguracja tego obiektu została przedstawiona na **rysunku 4**. Ten obiekt wykorzystamy do wyświetlania aktualnego czasu systemowego. Dlatego w polu *Field* zostało tylko zaznaczone *Show time*. Aby czas systemowy był poprawnie synchronizowany, w drzewie projektu należy skonfigurować zakładkę *Connection*. Należy tylko zmienić w kolumnie *HMI time synchronization mode* na *Slave*.

Ekran

We wstępie napisałem o ekranach, z którymi będziemy pracować w tej części kursu. Przed przystąpieniem do realizacji tego zadania należy jeszcze przygotować kilka rzeczy.

Hmi tags Przed przystąpieniem do tworzenia ekranów na początku należy przygotować dane. W części dla PLC jest warstwa wizualizacji (*01Visualization*) zawierająca bloki danych ze zmiennymi, które zostaną wyświetlone w celu prezentacji wartości lub ich zmiany. Bloki danych, które zostaną użyte w tej części kursu, zostały przedstawione na **rysunku 5**. Na rysunku 5 zamieszczono pięć bloków DB. Skorzystamy tylko z czterech, ponieważ blok *V_InsertScene* nie będzie potrzebny w tej części kursu. W katalogu *HMI tags* tworzymy nowe grupy zgodnie z **rysunkiem 6**. Do każdej grupy dodajemy nową tablicę tagów o nazwie takiej samej, jak nazwa bloku danych. Wówczas pozostanie nam tylko skopiowanie danych z odpowiednich bloków danych do tablicy tagów o takiej samej nazwie.



Rysunek 9. Docelowy wygląd ekranu Info

Rysunek 8. Ekran zawierający dni tygodnia

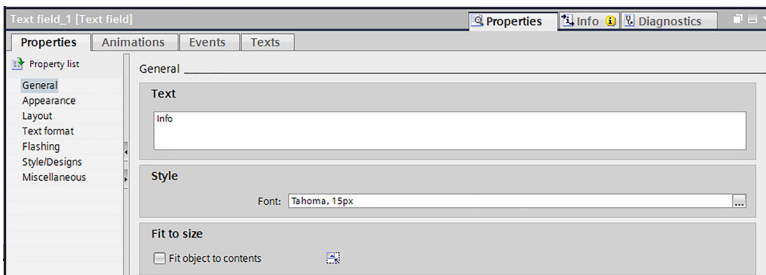
Text list Informacje na wizualizacji można przedstawiać w postaci liczb.

Kolejny sposób to prezentacja informacji w postaci tekstowej (wyrazy, zdania). Istnieje też możliwość powiązania wartości liczbowych z tekstem. W zależności od aktualnej wartości ukaże się odpowiedni ciąg

znaków. Do tego celu służą tz. *Text list*. Takie powiązanie wykorzystamy na ekranie *Info*, aby przedstawiać w postaci słowa aktualny dzień tygodnia oraz na ekranie *Light* w celu przedstawienia aktualnego stanu każdego z wejść i wyjść cyfrowych PLC. Tworzenie list tekstowych odbywa się w *Text and graphic lists*, co przedstawia **rysunek 7** oraz **rysunek 8**.

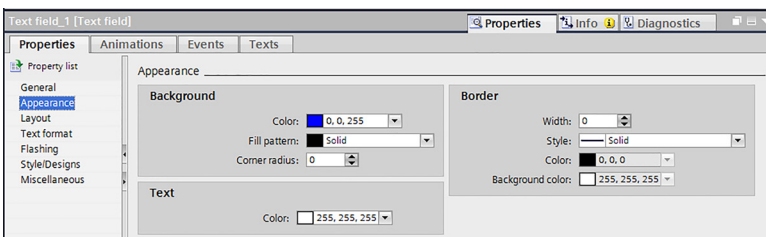
Rysunek 7 pokazuje powiązanie poziomów logicznych z ich opisem, natomiast rysunek 8 zawiera dni tygodnia.

Info Zacniemy od przygotowania ekranu, który dodaliśmy do projektu pod nazwą *Info*. Docelowy wygląd przedstawiono na **rysunku 9**. Na początku dodamy do ekranu obiekt *Text field* z *Basic objects*. Należy to wykonać pięć razy (nagłówki *Info*, *Date*, *Weekday*, *Time*, *Temperature*). Zacniemy od przygotowania nagłówka *Info*, który będzie informacją, na którym ekranie znajduje się aktualnie użytkownik. W pierwszej kolejności należy zaznaczyć obiekt, który chcemy edytować. Zacniemy od zakładki *General*, co ilustruje **rysunek 10**.



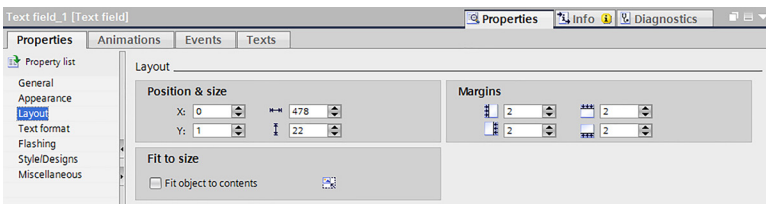
Rysunek 10. Edytowanie obiektu – zakładka General

W polu *Text* należy zmienić nazwę na *Info* oraz w polu *Fit to size* odznaczyć *Fit object to contents*. Przechodzimy do kolejnej zakładki, czyli *Appearance*. Ta zakładka została pokazana na **rysunku 11**. W polu *Background* należy zmienić kolor na niebieski, natomiast w polu *Text* zmieniamy kolor na biały. Pozostało nam tylko przejść do zakładki *Layout*, co przedstawia **rysunek 12**. Ta zakładka służy do ustawienia pozycji i wielkości pola tekstowego. My jednak ustawimy to ręcznie za pomocą myszki. Należy tylko w polu *Fit to size* odznaczyć *Fit object to contents*. Pozostało nam tylko ręczne ustawienie obiektu pola tekstowego, aby przypominało nagłówek.



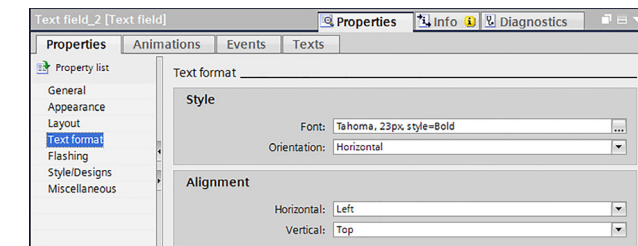
Rysunek 11. Edytowanie obiektu – zakładka Appearance

Pozostałe obiekty pól tekstowych należy zmienić odpowiednio nazwy (zakładka *General* i następnie pole *Text*). Zmiany wielkości czcionki lub jej typu dokonuje się w zakładce *Text format*, co przedstawia **rysunek 13**. Zmieniamy tylko rozmiar czcionki i dodajemy pogrubienie, więc należy zmienić pole *Font*. Analogiczne zmiany robimy dla pozostałych pól tekstowych. Wówczas pozostanie nam tylko odpowiednio rozmieszczać te obiekty.



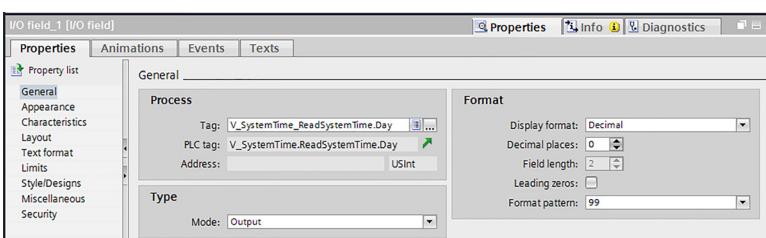
Rysunek 12. Ustawienie pozycji i wielkości pola tekstowego – zakładka Layout

Do wyświetlenia poszczególnych składników daty i czasu potrzebujemy obiektów znajdujących się w *Elements*, który nazywa się dokładnie *I/O field*. Należy dodać łącznie sześć takich obiektów. Jedynie do wyświetlenia dnia tygodnia będzie na potrzebny obiekt o nazwie *Symbolic I/O*. Zacniemy od parametryzacji obiektu, który będzie wyświetlał aktualny dzień miesiąca. Po zaznaczeniu obiektu *I/O field* przechodzimy do jego właściwości, które zostały pokazane na **rysunku 14**. W polu *Process* należy wskazać tag, który powinien być wyświetlany. W tym przypadku będzie to dzień, więc przycisk zawierający trzy kropki. Pojawi się dodatkowe okno, gdzie z katalogu *HMI tags* odszukujemy tablicę o nazwie *V_SystemTime*. Ten proces ilustruje **rysunek 15**.



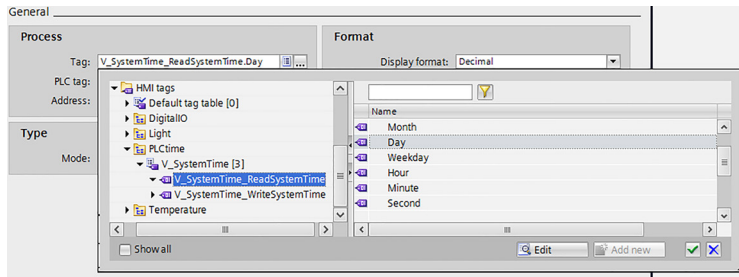
Rysunek 13. Zamiana wielkości czcionki lub jej typu

Z tej tablicy wybieramy tag o nazwie *V_SystemTime_ReadSystemTime* (który w tym wypadku jest strukturą) i wybieramy *Day*. Ponieważ ten obiekt będzie tylko do wyświetlania aktualnego dnia tygodnia, w polu *Type* zmieniamy z listy rozwijanej *Mode* na *Output*. Z tego względu, że dzień miesiąca może być tylko jedno- lub dwucyfrowy, w *Format pattern* (pole *Format*) wpisujemy *99*. W podobny sposób należy wybrać odpowiednie składniki z tagu *V_SystemTime_ReadSystemTime* dla pozostałych obiektów *I/O field*. Trzeba pamiętać, że rok jest czterocyfrowy, więc odpowiednio należy ustawić *Format pattern* (pole *Format*) wpisujemy *9999*.

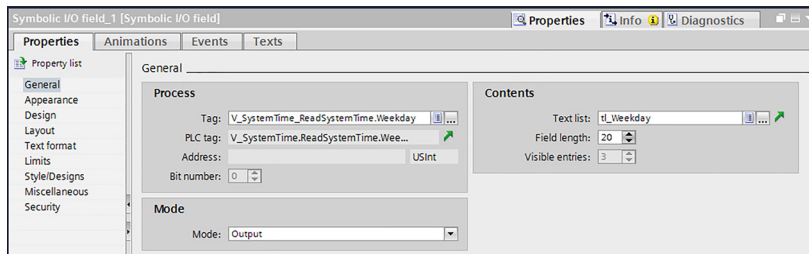


Rysunek 14. Właściwości pola I/O field

Założyliśmy, że dzień tygodnia będziemy wyświetlać w postaci tekstu. Do tego celu musimy koniecznie użyć obiektu *Symbolic I/O*. Parametryzację przedstawia **rysunek 16**. Zmiana tekstu nastąpi w zależności od wartości tagu. Dlatego w polu *Process* wybieramy odpowiedni tag. W polu *Contents* wybieramy utworzoną przez nas wcześniej listę tekstową o nazwie *tl_Weekday*. W przypadku



Rysunek 15. Wstawienie tagu, który powinien być wyświetlany



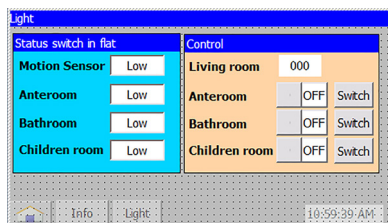
Rysunek 16. Parametryzacja Symbolic I/O

wyświetlania temperatury, poza wskazaniem tagu z tablicy *V_Temperature* należy zmienić *Format pattern* na *s9999.99*. Temperatura może być dodatnia lub ujemna, dlatego literka *s* powoduje dodanie znaku. Wystarczy dokładność dwóch miejsc po przecinku, więc tak przygotowano wzór do wyświetlania temperatury. W ten sposób ekran *Info* został przygotowany.

Light Pozostało nam przygotować ekran *Light*, który powinien wyglądać jak na **rysunku 17**. Zaczynamy od przygotowania nagłówka ekranu, czyli *Light*. Postępujemy analogicznie do poprzedniego ekranu. Następnie możemy przejść do dodania pól tekstowych opisujących nazwy sygnałów wejściowych (umieszczamy z lewej strony ekranu *Light*) oraz pól tekstowych do opisu sygnałów wyjściowych (umieszczamy z prawej strony ekranu *Light*). Zmieniamy rozmiar czcionki na *15 px* oraz pogrubiamy. Poziomy sygnałów wejściowych zostaną wyświetlone w postaci tekstu. Z tego powodu dodajemy obiekty *Symbolic I/O*. Przykładowa konfiguracja dla czujnika ruchu (*Motion Sensor*) została przedstawiona na **rysunku 18**.

Tagi znajdują się tym razem w *V_Digital*. Są tam dwa tagi typu *Array* (czyli typ tablicowy). Ponieważ czujnik ruchu jest podłączony do wejścia cyfrowego o adresie *0*, został wskazany w polu *Tag* element zerowy. Tym razem w polu *Text list* trzeba wybrać listę o nazwie *tl_LevelIO*. Analogicznie postępujemy z pozostałymi polami *Symbolic I/O*. Należy tylko wskazać kolejne indeksy dla *V_Digital_IO_Input*. W wypadku sygnałów wyjściowych dla *Living room* dodajemy obiekt *Field IO*. Konfiguracja została przedstawiona na **rysunku 19**. Jako tag podpinamy *V_Light_LivingRoom*. W polu *Mode* został użyty typ jako *Input/output*. Wynika to z faktu, że to pole powinno wyświetlać aktualną wartość ustawionego natężenia oświetlenia, jak również powinna być możliwość wprowadzenia nowej wartości. Dla *Anteroom* wstawiamy obiekt *Switch* oraz *Button*. Za pomocą tych dwóch obiektów użytkownik będzie miał możliwość zmiany włączenia oświetlenia w przedpokoju. Konfiguracje obiektu *Switch* przedstawia **rysunek 20**.

Podobnie jak w przypadku innych obiektów, tak samo obiekt *Switch* wymaga powiązania z tagiem. W tym wypadku będzie to *V_Light_Anteroom*. Dla każdego z dostępnych stanów (stan niski, stan wysoki)



Rysunek 17. Wygląd ekranu Light

można w polu *Label* ustawić etykietę. Dla obiektu *Button* zmieniamy nazwę na *Switch*. Następnie we właściwościach tego obiektu trzeba przejść do zakładki *Events*. W zakładce *Release* należy wybrać zdarzenie *InvertBit* i wybrać także tag *V_Light_Anteroom*. Dzięki temu naciśnięcie przycisku będzie powodowało zmianę wartości tagu, a tym samym zmianę stanu oświetlenia. Analogicznie należy skonfigurować obiekty i powiązać z tagami dla *Bathroom* i *Children room*.

Ostatnia rzecz, jaką należy zrobić w celu poprawienia wyglądu, to zgrupowanie obiektów. Na początku zaznaczamy wszystkie obiekty, które zostały dodane na ekranie *Light* (poza nagłówkiem). Następnie przechodzimy do właściwości i wybieramy *Miscellaneous*. Tutaj ustawiamy w polu *Layer* warstwę *1-Layer_1*. Ogólna zasada jest taka, że im większy numer warstwy to obiekty znajdują się bliżej użytkownika. Zatem obiekty znajdujące się na niższych warstwach są przysłaniane.

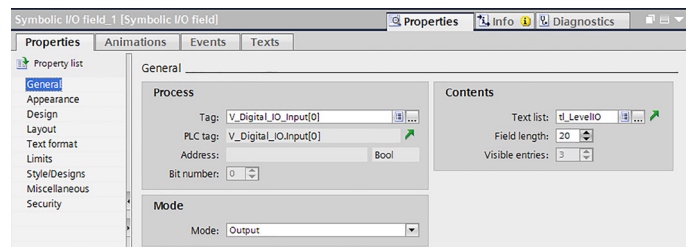
Do zgrupowania obiektów użyjemy obiektu o nazwie *Rectangle*. Należy dodać dwa takie obiekty. Każdy z nich powinien znajdować się na warstwie *0 - Layer_0*. W zakładce *Appearance* należy ustawić odpowiednie kolory (pole *Color*). Do każdego obiektu *Rectangle* trzeba przygotować nagłówek z opisem, co znajduje się na danej „zakładce”. Nagłówek przygotowujemy analogicznie, jak do nagłówków do ekranu. Nagłówek również powinien znajdować się na warstwie *1-Layer_1*.

Pozostało tylko skompilować cały projekt i wgrać na panel HMI (lub uruchomić symulator). Przetestuj informacje wyświetlane na każdym z ekranów. Sprawdź, czy stany sygnałów odpowiadają tym, które są prezentowane na HMI. Czas prawdopodobnie nie będzie się zgadzał. Tym zajmiemy się w kolejnych artykułach.

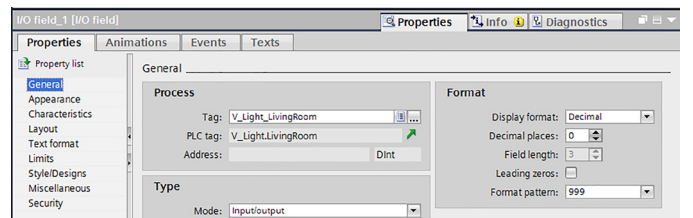
Ćwiczenie

Na ekranie *Light* w części *Status switch in flat* dodać pola *IO field* i wyświetlać wartości wejść w postaci liczbowej (0 lub 1).

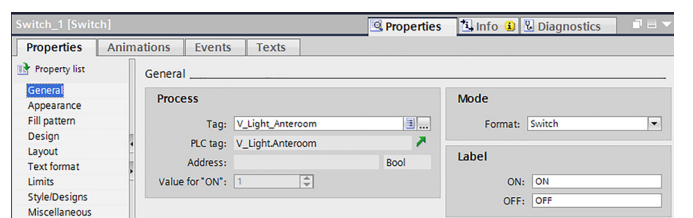
Tomasz Gilewski
www.mistrzplc.pl
 tomasz.gilewski@mistrzplc.pl



Rysunek 18. Przykładowa konfiguracja dla czujnika ruchu



Rysunek 19. Konfiguracja Living room



Rysunek 20. Konfiguracja obiektu Switch