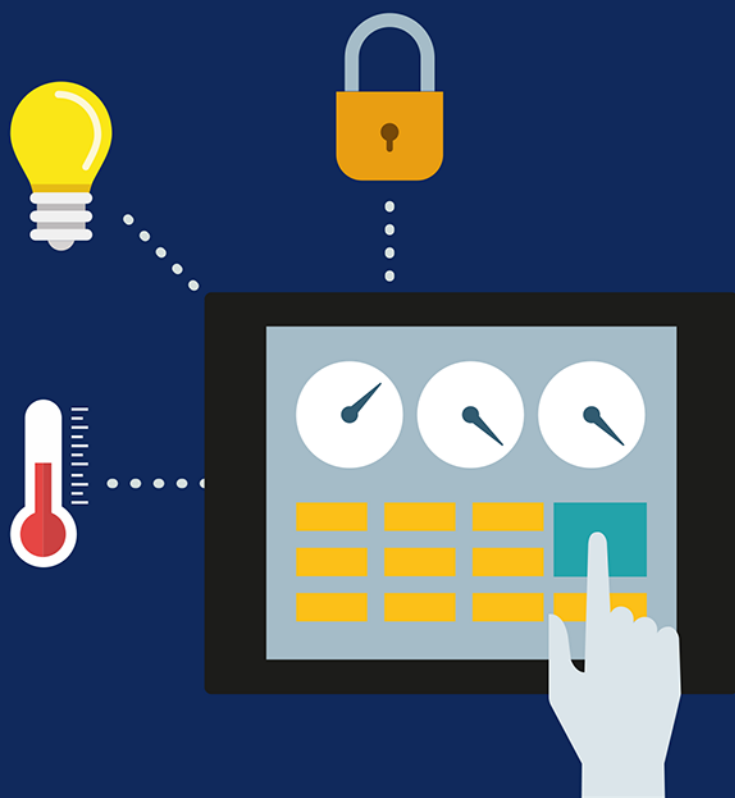


Tomasz Gilewski

TWORZENIE WIZUALIZACJI NA PANELE HMI FIRMY SIEMENS



- Dowiedz się, do czego służą i jak działają panele sterownicze
 - Poznaj funkcje paneli HMI firmy Siemens
- Przygotuj do pracy system zarządzania inteligentnym domem

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Małgorzata Kulik

Projekt okładki: Studio Gravite/Olsztyn
Obarek, Pokoński, Pazdrijowski, Zaprucki

Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock.

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
<http://helion.pl/user/opinie/twphsi>
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-3776-3

Copyright © Helion 2019

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Wstęp	7
Rozdział 1. Pierwsze uruchomienie	17
Konfiguracja serwera czasu	22
Ustawienie daty i czasu	22
Aktywacja sygnału dźwiękowego	25
Konfiguracja autostartu	25
Aktywacja zabezpieczenia hasłem	26
Transfer projektu	28
Konfiguracja ekranu	29
Rozdział 2. TIA Portal	31
Pomoc dostępna w środowisku	34
Menedżer licencji	35
Automatyczne aktualizacje	35
Rozdział 3. Pierwszy projekt	37
Projekt w TIA Portal	37
Dodanie panelu	38
Kreator ekranów	39
Konfiguracja HMI	47
Tworzenie ekranu	48
Wgranie projektu do urządzenia	49
Symulacja	53

Rozdział 4. Sterowanie inteligentnym domem 55

Schemat elektryczny	58
Warstwa sprzętowa	61
Moduł DataLog	61
Moduł Digital IO	66
Moduł HSC	66
Moduł PLC Time	67
Moduł PWM	68
Moduł Recipe	69
Warstwa wizualizacji	71
Moduł Digital IO	71
Warstwa Error	72
Moduł Light	73
Moduł PLC Time	74
Moduł Temperature	75
Warstwa błędów	75
Moduł GetError	75
Moduł OB	81
Moduł SoftwareError	84
Moduł Diagnostic Buffer	87
Moduł Check Error	87
Moduł Reset	88
Warstwa logiczna	90
Moduł ADC	91
Moduł DataLog	92
Moduł Digital IO	99
Moduł Light Control	100
Moduł PLC Time	119
Konfiguracja i uruchomienie	121

Rozdział 5. Ekrany 123

Dodawanie ekranów	123
Hmi tags	125
Text list	126
Info	127
Light	131

Rozdział 6. Alarmy	135
Konfiguracja zmiennych do alarmów	136
Ekrany	138
Rozdział 7. Użytkownicy	143
Przygotowanie	143
Konfiguracja zarządzania użytkownikami	144
Uruchomienie	146
Rozdział 8. Języki	149
Konfiguracja	149
Przełączanie ekranów	151
Uruchomienie	154
Podsumowanie	157
Książki	158
Kursy wideo	159
Bibliografia	163
Skorowidz	164

Rozdział 5.

Ekrany

W tym rozdziale zajmiemy się pracą z ekranami. Jest to podstawa, ponieważ ogólna idea wizualizacji polega na pokazywaniu odpowiednich wartości oraz tego, jak się zmieniają. To właśnie na ekranach umieszcza się elementy biblioteczne, aby przedstawić operatorowi i obsłudze informacje o danym składniku procesu.

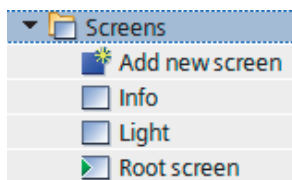
Dodawanie ekranów

Pierwszy krok to dodanie do projektu nowych ekranów, na których zostanie zbudowana wizualizacja pokazująca aktualny czas, datę 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 do 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 przedstawia rysunek 5.1.

RYSUNEK 5.1.

Lista ekranów
w projekcie

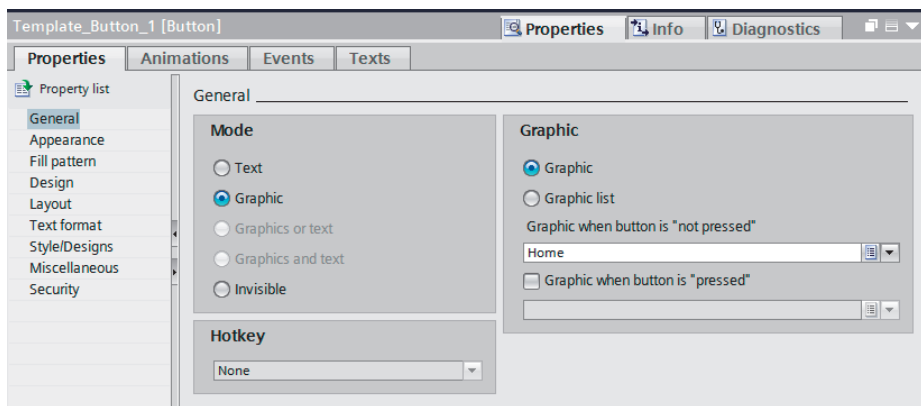


Ekran startowy to *Root screen*, co jest symbolicznie zaznaczone zielonym trójkątem. Oczywiście istnieje możliwość zmiany. Każdy ekran w projekcie można ustawić, aby był startowy.

Zacniemy od utworzenia szablonu (*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*.

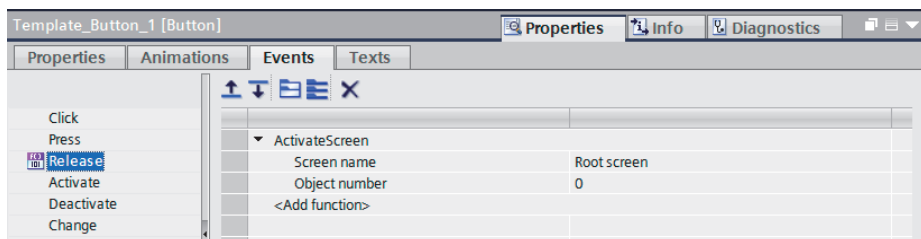
Następnie zmieniamy nazwę na *MyTemplate*. Z prawej strony środowiska TIA Portal, z karty *Toolbox* dodajemy z zakładki *Elements* obiekt *Button*. Umieszczamy go w lewym dolnym narożniku. Przechodzimy do właściwości tego obiektu poprzez jego zaznaczenie. Przedstawia to rysunek 5.2.



RYСУNEK 5.2. Edytowanie właściwości obiektu

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, jaką mamy wykonać 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 taką funkcjonalność utworzyć, trzeba wybrać zakładkę *Events*, co przedstawia rysunek 5.3.



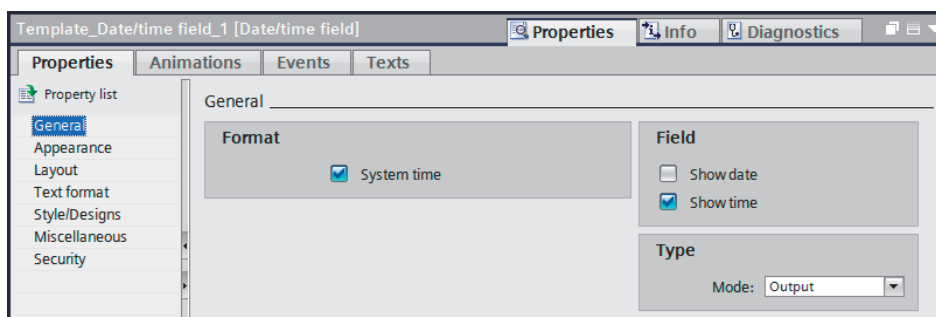
RYСУNEK 5.3. Zakładka *Events*

Po lewej stronie są dostępne zdarzenia, które są powiązane z przyciskiem. Skorzystamy z *Release*, więc dopiero w momencie zwolnienia przycisku nastąpi wykonanie akcji, czyli zmiana ekranu. Akcję wybieramy z listy *<Add function>* lub można też wpisać ją 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 odpowiednio 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 5.4.



RYСУNEK 5.4. Konfiguracja obiektu *Data/time field*

Ten obiekt wykorzystamy do wyświetlania aktualnego czasu systemowego. Dlatego w polu *Field* zostało tylko zaznaczone *Show time*.

Aby czas systemowy był poprawnie synchronizowany, to w drzewie projektu należy skonfigurować zakładkę *Connection*. Należy tylko zmienić w kolumnie *HMI time synchronization mode* na *Slave*.

Na wstępie napisałem o ekranach, z którymi będziemy pracować w tym rozdziale. Przed przystąpieniem do realizacji tego zadania należy jeszcze przygotować kilka rzeczy.

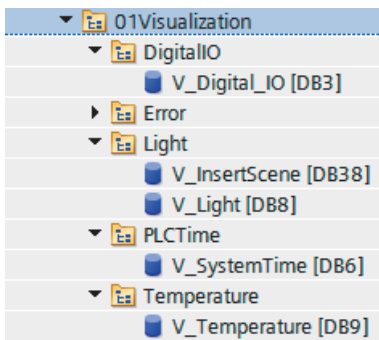
Hmi tags

Przed przystąpieniem do tworzenia ekranów 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 z warstwy wizualizacji zostały przedstawione na rysunku 5.5.

RYSUNEK 5.5.

*Bloki
wykorzystywane
w projekcie*

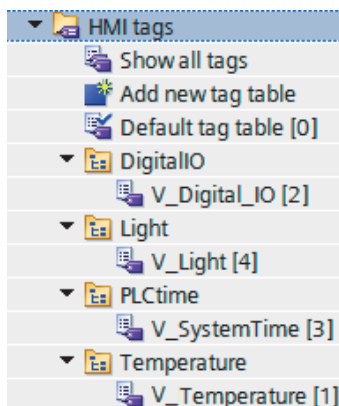


Na rysunku 5.5 widzisz pięć bloków DB. Skorzystamy tylko z czterech, ponieważ blok *V_InsertScene* nie będzie potrzebny w tym rozdziale.

W katalogu HMI tags tworzymy nowe grupy zgodnie z rysunkiem 5.6.

RYSUNEK 5.6.

*Struktura grup
w katalogu HMI
tags*



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.

Text list

Informacje można przedstawiać na wizualizacji w postaci liczb. To chyba naturalne i pewnie tak też podejrzewałeś. 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 tzw. *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 przedstawiają rysunki 5.7 i 5.8.

RYSUNEK 5.7.

Powiązanie stanów logicznych z ich opisem

Text lists			
	Name ▲	Selection	Comment
1	tl_LevelIO	Value/Range ▼	
2	tl_Weekday	Value/Range	
	<Add new>		

Text list entries			
	Default	Value ▲	Text
1	☐	0	Low
2	☐	1	High

RYSUNEK 5.8.

Dni tygodnia

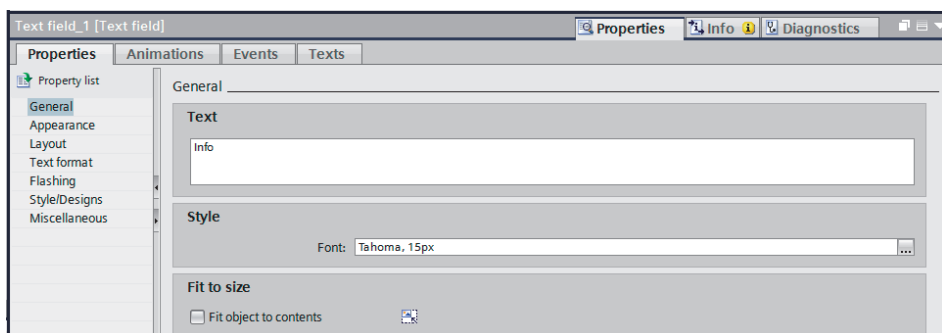
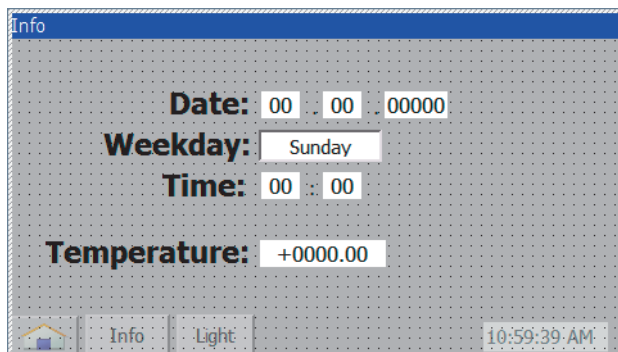
Text lists			
	Name ▲	Selection	Comment
1	tl_LevelIO	Value/Range	
2	tl_Weekday	Value/Range ▼	
	<Add new>		

Text list entries			
	Default	Value ▲	Text
1	☐	1	Sunday
2	☐	2	Monday
3	☐	3	Tuesday
4	☐	4	Wednesday
5	☐	5	Thursday
6	☐	6	Friday
7	☐	7	Saturday

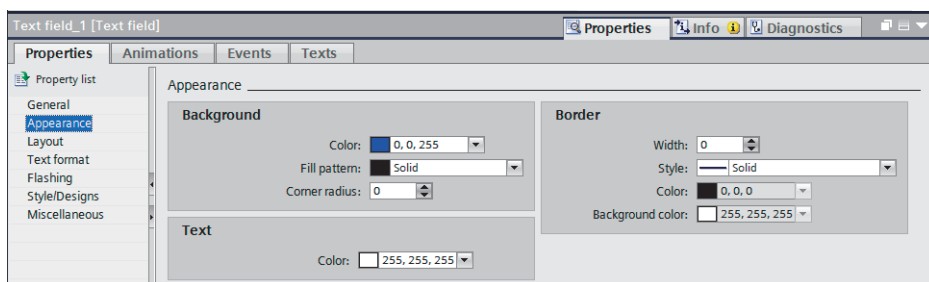
Info

Zacniemy od przygotowania ekranu, który dodaliśmy do projektu pod nazwą *Info*. Docelowy wygląd przedstawiono na rysunku 5.9.

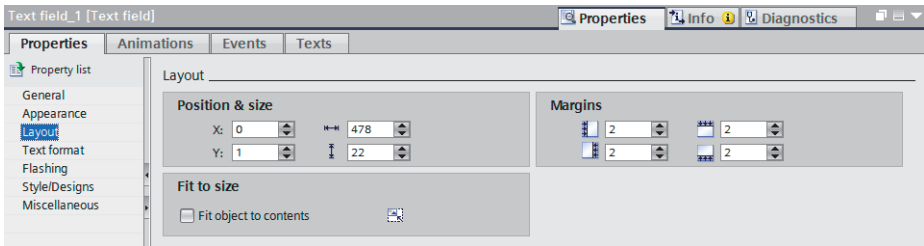
Na początku dodamy do ekranu obiekt *Text field* z *Basic objects*. Należy to wykonać pięć razy (nagłówek *Info*, *Date*, *Weekday*, *Time*, *Temperature*). Zacniemy od przygotowania nagłówka *Info*. Będzie on informacją, na którym ekranie znajduje się aktualnie użytkownik. W pierwszej kolejności należy zaznaczyć obiekt, który chcemy edytować. Zacniemy od zakładki *General*, co przedstawia rysunek 5.10.

RYSUNEK 5.9.Wygląd ekranu *Info***RYSUNEK 5.10.** Widok zakładki *General*

Trzeba w polu *Text* zmienić nazwę na *Info* oraz w polu *Fit to size* usunąć zaznaczenie *Fit object to contents*. Przechodzimy do kolejnej zakładki, czyli *Appearance*. Ta zakładka została przedstawiona na rysunku 5.11.

**RYSUNEK 5.11.** Widok zakładki *Appearance*

W tej zakładce należy zmienić w polu *Background* kolor na niebieski. Natomiast w polu *Text* zmieniamy kolor na biały. Pozostało nam tylko przejść do zakładki *Layout*, co przedstawia rysunek 5.12.

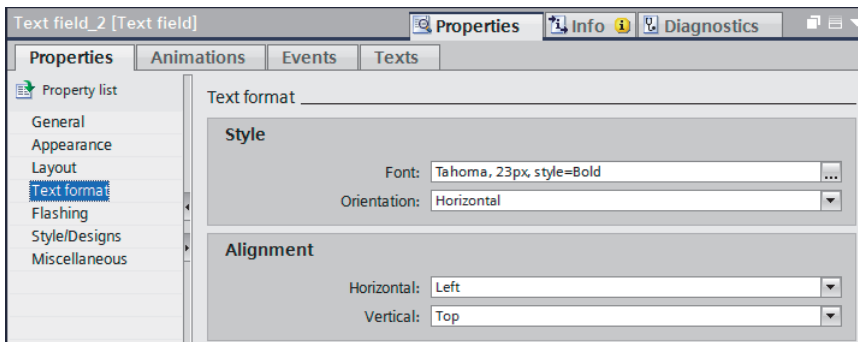


RYSUNEK 5.12. Widok zakładki Layout

Ta zakładka służy do ustawienia pozycji i rozmiaru pola tekstowego. My jednak ustawimy to ręcznie za pomocą myszy. Należy tylko w polu *Fit to size* usunąć zaznaczenie *Fit object to contents*.

Pozostało nam tylko ręczne ustawienie obiektu pola tekstowego, aby przypominało nagłówek.

Pozostałym obiektom pól tekstowych należy zmienić odpowiednio nazwy (zakładka *General* i następnie pole *Text*). Zmiany rozmiaru czcionki lub jej typu dokonuje się w zakładce *Text format*, co przedstawia rysunek 5.13.

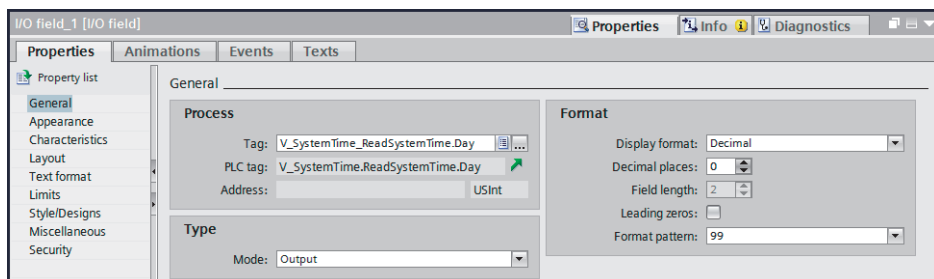


RYSUNEK 5.13. Widok zakładki Text format

Zmieniamy tylko rozmiar czcionki i dodajemy pogrubienie, więc należy zmienić pole *Font*. Analogiczne zmiany robimy dla pozostałych pól tekstowych. Wówczas pozostanie nam tylko odpowiednio porozmieszczać te obiekty.

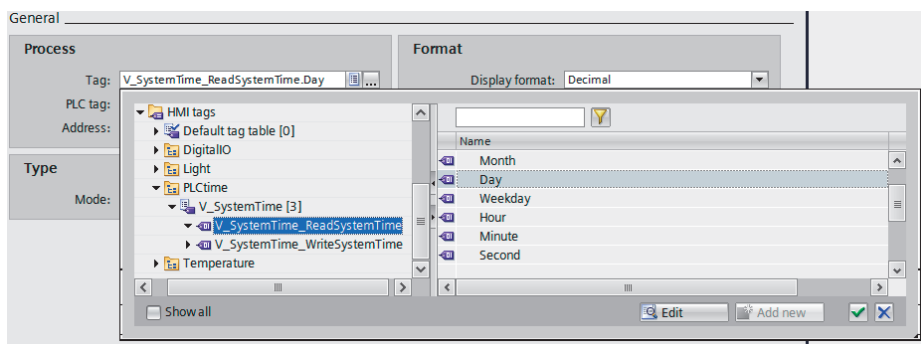
Do wyświetlenia poszczególnych składników daty i czasu potrzebujemy obiektów znajdujących się w *Elements*, który nazywa się dokładnie *I/O field*. Należy dodać łącznie sześć takich obiektów. Jedynie do wyświetlenia dnia tygodnia będzie na potrzeby obiekt o nazwie *Symbolic I/O*.

Zacniemy od parametryzacji obiektu, który będzie wyświetlał aktualny dzień miesiąca. Po zaznaczeniu obiektu *I/O field* przechodzimy do jego właściwości, które zostały przedstawione na rysunku 5.14.



RYСУNEK 5.14. Właściwości obiektu *I/O field*

W polu *Process* należy wskazać tag, który powinien być wyświetlany. W tym przypadku będzie to dzień, więc wciskamy przycisk zawierający trzy kropki. Pojawi się dodatkowe okno, gdzie w katalogu *HMI tags* znajdujemy tablicę o nazwie *V_SystemTime*. Ten proces przedstawia rysunek 5.15.

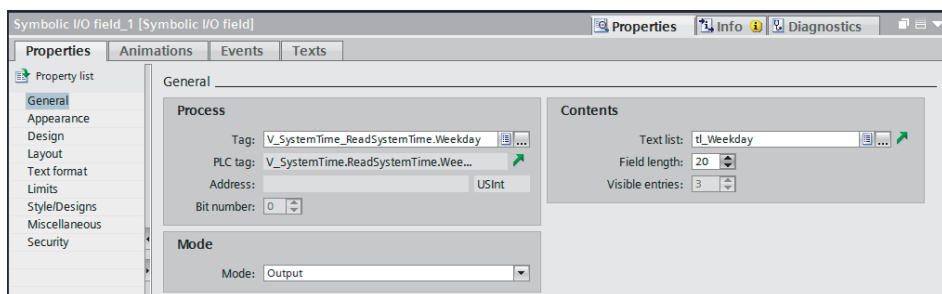


RYСУNEK 5.15. Zawartość tablicy *V_SystemTime*

Z tej tablicy wybieramy tag o nazwie *V_SystemTime_ReadSystemTime* (który w tym przypadku jest strukturą) i wybieramy *Day*. Ponieważ ten obiekt będzie służył jedynie do wyświetlania aktualnego dnia tygodnia, to w polu *Type* na liście rozwijanej zmieniamy *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.

Założyliśmy, że dzień tygodnia będziemy wyświetlać w postaci tekstu. Do tego celu musimy koniecznie użyć obiektu *Symbolic I/O*. Parametryzację przedstawia rysunek 5.16.



RYSUNEK 5.16. Właściwości obiektu *Symbolic I/O*

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*.

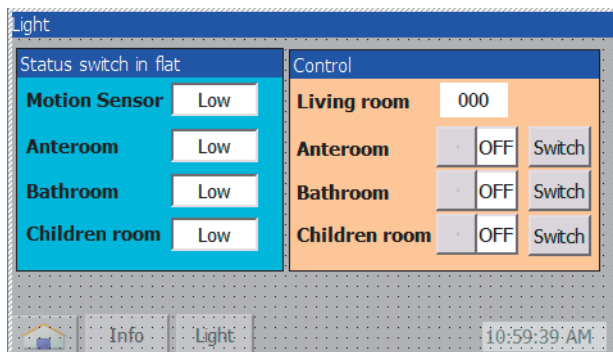
W przypadku wyświetlania temperatury poza wskazaniem tagu z tablicy *V_Temperature* należy zmienić *Format pattern* na *s9999.99*. Temperatura może być dodatnia lub ujemna, a literka *s* powoduje dodanie znaku. Wystarczy dokładność dwóch miejsc po przecinku, więc tak przygotowano wzór do wyświetlania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

Light

Pozostało nam przygotować ekran *Light*, który powinien wyglądać podobnie, jak przedstawiono na rysunku 5.17.

RYSUNEK 5.17.

Wygląd ekranu
Light

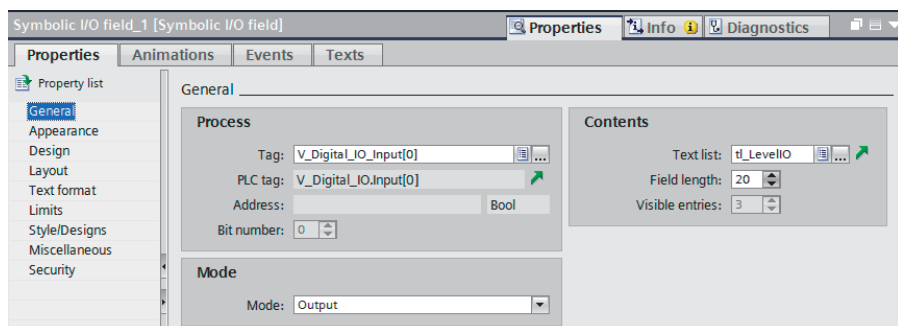


Zaczynamy od przygotowania nagłówka ekranu, czyli *Light*. Postępujemy analogicznie jak na poprzednim ekranie.

Następnie możemy przejść do dodania pól tekstowych opisujących nazwy sygnałów wejściowych (umieszczamy je z lewej strony ekranu *Light*) oraz pól tekstowych do opisu sygnałów

wyjściowych (umieszczamy je z prawej strony ekranu *Light*). Zmieniamy rozmiar czcionki na 15px oraz pogrubiamy.

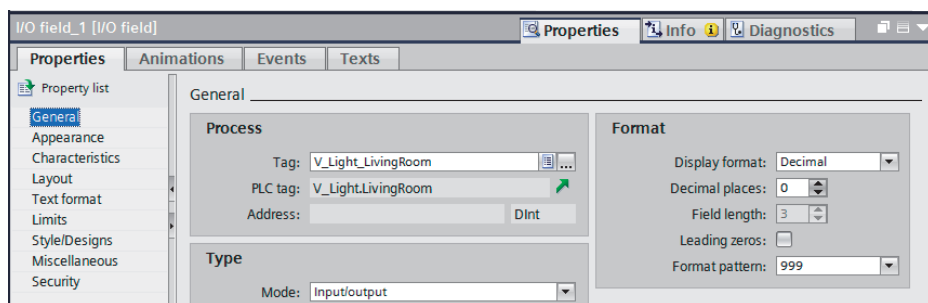
Stany sygnałów wejściowych zostaną wyświetlone w postaci tekstu. Z tego powodu dodajemy obiekty *Symbolic I/O*. Przykładowa konfiguracja dla czujnika ruchu (*Motion Sensor*) została przedstawiona na rysunku 5.18.



RYСУNEK 5.18. Przykładowa konfiguracja dla czujnika ruchu (*Motion Sensor*)

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, dlatego został wskazany w polu *Tag* zerowy element. 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*.

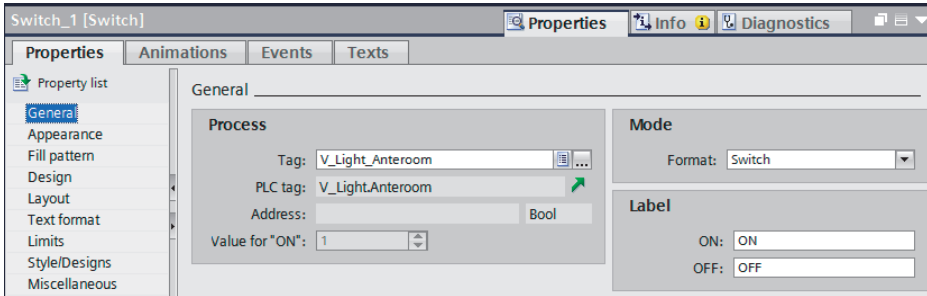
W przypadku sygnałów wyjściowych dla *Living room* dodajemy obiekt *Field IO*. Konfiguracja została przedstawiona na rysunku 5.19.



RYСУNEK 5.19. Konfiguracja obiektu *Field IO*

Jako tag podpinamy *V_Light_LivingRoom*. W polu *Mode* został użyty typ *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 obiekty *Switch* oraz *Button*. Za pomocą tych dwóch obiektów użytkownik będzie miał możliwość zmiany (włączenia lub wyłączenia) oświetlenia w przedpokoju. Konfigurację obiektu *Switch* przedstawia rysunek 5.20.



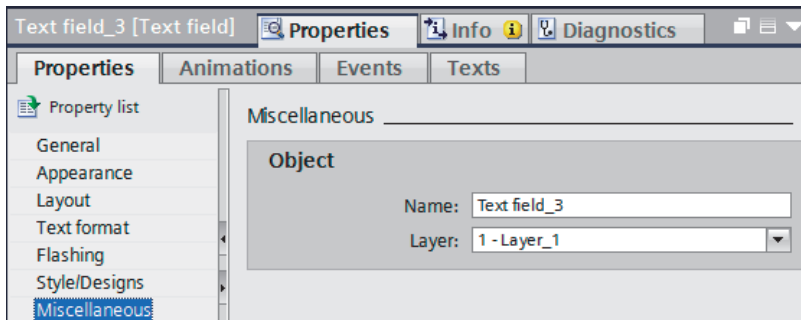
RYСУNEK 5.20. Konfiguracja obiektu *Switch*

Podobnie jak w przypadku innych obiektów, również obiekt *Switch* wymaga powiązania z tagiem. W tym wypadku będzie to *V_Light_Anteroom*. Dla każdego z dostępnych stanów (stan niski, stan wysoki) można w polu *Label* ustawić etykietę.

W przypadku obiekt *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 je z tagami dla *Bathrom* 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*, co przedstawia rysunek 5.21.



RYСУNEK 5.21. Widok zakładki *Miscellaneous*

Tutaj ustawiamy w polu *Layer* warstwę *1-Layer_1*. Ogólna zasada jest taka, że im wyższy numer warstwy, tym bliżej użytkownika znajdują się obiekty. Zatem obiekty znajdujące się na niższych warstwach są przysyłane.

Do zgrupowania obiektów wykorzystamy te 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*). Dla każdego obiektu *Rectangle* należy przygotować nagłówek z opisem, co znajduje się na danej „zakładce”. Nagłówek przygotowujemy analogicznie jak nagłówki dla ekranu. Nagłówek również powinien znajdować się na warstwie *1-Layer_1*.

Pozostało tylko zrobić kompilację całego projektu i wgrać ją 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.

Skorowidz

A

ADC, 91
aktywacja sygnału
 dźwiękowego, 25
alarm analogowy, 138
Alarm
 indicator, 140
 view, 141
 window, 139
Alarms, 41, 135, 140, 142
Appearance, 128
Attributes, 135
automatyczne
 aktualizacje, 35
 wylogowanie, 147
autoryzacja, 146
autostart, 25

B

Basic objects, 45
blok
 EventDirectionHSC3, 104
 EventEqualHSC3, 104
 LC_ManegmentRecipe, 108
 LC_ParameterDBL, 108
 LC_ParametersRecipe, 108
 LC_SceneData, 114
 OB100, 76, 81, 92, 104
 OB80, 84
Buttons, 43

C

Check Error, 87
Connections, 45
Controls, 46
czas, 22
czujnik ruchu, 132

D

data, 22
Data/time field, 125
DataLog, 61, 92
Device configuration, 44
Diagnostic Buffer, 87
Digital IO, 66, 71, 99
dodawanie
 ekranów, 123
 języków, 149
 panelu HMI, 38
 zdarzeń, 142
dostęp do panelu, 26

E

edytowanie właściwości
 obiektu, 124
ekran, 29, 39, 123, 138
 Alarms, 140
 Light, 131
 Root screen, 44
 Service, 143

ekrany
 dodawanie, 123
 przełączanie, 151
 wizualizacji, 44
Elements, 46
Error, 72
Events, 124

F

Field IO, 132
funkcja
 ADC_Class(), 92
 DIO_ClassDigitalIO(), 99
 DIO_ReadIO(), 99
 DL_AddressState(), 95
 DL_CheckChanges(), 97
 DL_ClassDataLog(), 98
 DL_ControlFunction(), 93
 DL_CreateLogs(), 93
 DL_WriteRecord(), 96
 ERR_ClassCheckError(), 88
 ERR_ClassGetError(), 77, 80, 81
 ERR_ClassReset(), 90
 ERR_CompareGetError
 ↪Component(), 79
 ERR_CompareOB80OBnr(), 83
 ERR_CompareSoftware
 ↪Error(), 85
 ERR_GetErrorFrom
 ↪Structure(), 78

ERR_GetErrorID(), 78
 ERR_GetFaultID(), 82
 ERR_GetSoftwareError(),
 86
 ERR_OB80(), 83
 ERR_OverflowBuffer
 →Control(), 79
 ERR_ResetGetError(), 89
 ERR_ResetOB80(), 89
 ERR_ResetSoftwareError(),
 89
 ERR_SaveErrorToBuffer(),
 80
 ERR_SaveSoftwareError
 →ToBuffer(), 86
 HD_CTRL_HSC(), 67
 HD_CTRL_PWM(), 68
 HD_DataLogClose(), 65
 HD_DataLogCreate(), 63
 HD_DataLogOpen(), 64
 HD_DataLogWrite(), 65
 HD_RD_SYS_T(), 67
 HD_RecipeExport(), 69
 HD_RecipeImport(), 70
 HD_WR_SYS_T(), 68
 LC_AnteroomLight(), 101
 LC_ClassLight(), 102
 LC_ClassRecipe(), 114
 LC_ClassScene(), 115
 LC_ControlLoadMemory(),
 109
 LC_ControlRecipe
 →Function(), 110
 LC_CopyLoadMemory(),
 112
 LC_ExportRecipe(), 110
 LC_ImportRecipe(), 113
 LC_InitializationScene(),
 115
 LC_Light(), 100
 LC_LightLevel(), 105
 LC_LivingRoom(), 106
 LC_SceneActivate(), 117
 LC_WeekdayCarculate(),
 116
 PT_ClassPLCTime(), 121

PT_ReadTimeOfDay(),
 119
 PT_SetTimeOfDay(), 120

G

General, 128
 GetError, 75
 Global screen, 139
 Graphics, 47
 Groups, 145

H

Hardware, 61
 hasło, 26
 Historical data, 45
 HMI, Human Machine
 Interface, 7
 alarms, 45
 tags, 45, 125
 HSC, 66

I

I/O field, 130
 Info, 127

J

języki, 149

K

katalog HMI tags, 126
 kompilowanie projektu, 49
 konfiguracja, 17, 121
 alarmów, 141
 alarmu analogowego, 138
 autostartu, 25
 czasu, 23, 24
 dla czujnika ruchu, 132
 ekranu, 29
 HMI, 47
 obiektu
 Alarm indicator, 140
 Alarm window, 139

Data/time field, 125
 Field IO, 132
 Switch, 133
 projektu, 47, 48
 serwera czasu, 22
 zakładki Toolbar, 141
 zarządzania
 użytkownikami, 144
 zmiennych do alarmów,
 136
 zdarzenia, 144
 zdarzenie SetTag, 154
 kreator ekranów, 39
 książki, 158
 kursy wideo, 159

L

Layout, 129
 licencja, 35
 Light, 73, 131
 Light Control, 100
 Login, 147

M

menedżer licencji, 35
 Miscellaneous, 133
 moduł
 ADC, 91
 Check Error, 87
 DataLog, 61, 92
 Diagnostic Buffer, 87
 Digital IO, 66, 71, 99
 GetError, 75
 HSC, 66
 Light, 73
 Light Control, 100
 OB, 81
 PLC Time, 67, 74, 119
 PWM, 68
 Recipe, 69
 Reset, 88
 SoftwareError, 84
 Temperature, 75
 Motion Sensor, 132
 multipanel, 9

O

OB, 81

obiekt

Alarm indicator, 140

Alarm window, 139

Data/time field, 125

I/O field, 130, 132

Switch, 133

Symbolic I/O, 131

okno

Create a new project, 37

Global screen, 139

kreatora

Alarms, 41

Buttons, 43

PLC connections, 39

Screen layout, 40

Screen navigation, 42

System screens, 42

Login, 147

nadzoru, 50

Project texts, 150

Start Center, 21

symulatora, 54

Online & diagnostics, 44

P

panel

Basic, 12, 13

Comfort, 12, 13

HMI KTP400 Basic, 17

panele

I generacji, 10

II generacji, 11

mobilne, 8

przyciskowe, 8

pierwszy projekt, 37

PLC, Programming Logic

Controller, 7

connections, 39

Time, 67, 74, 119

podłączenie zasilania, 19

pomoc, 34

Portal view, 31

program

Start Center, 20

TIA Portal, 31

WinCC Basic, 53

Project

texts, 150

stt_ERR_Enum_SoftwareError, 84

stt_ERR_SoftwareError, 85

view, 33

projekt

dodawanie języków, 149

kompilowanie, 49

konfiguracja, 47, 48, 121

okno nadzoru, 50

uruchomienie, 121

w pamięci urządzenia, 51

w TIA Portal, 37

przełączanie ekranów, 151

przycisk

LogOff, 148

przełączania ekranów, 151

Reset, 142

PWM, 68

R

receptury, 106

Recipe, 69

Recipes, 45

Reset, 88

Root screen, 44, 48

Runtime setting, 44

SSCADA, Supervisory Control
and Data Acquisition, 7

sceny, 114

Scheduled tasks, 45

schemat elektryczny, 58

Screen

layout, 40

management, 44

navigation, 42

Screens, 44

Service, 143

serwer

czasu, 22

NTP, 23

SetTag, 154

SoftwareError, 84

Start Center, 20

sterowanie, 55

struktura

stt_DL_Parameters, 62

stt_ERR_Enum_SoftwareError, 84

stt_ERR_GetError, 77

stt_ERR_OB80, 82

stt_ERR_SoftwareError, 85

stt_PWM, 68

V_ErrorOB_Int0, 137

V_InsertScene, 73

V_Light, 73

V_SoftwareError_Int, 136

V_SystemTime, 74, 130

V_Temperature, 75

Switch, 133

sygnał dźwiękowy, 25

Symbolic I/O, 131

symulacja, 53

system SCADA, 7

System screens, 42

T

tablica

tagów Language, 152

V_SystemTime, 130

tag ActualLanguage, 153

Temperature, 75

Text

and graphic lists, 45

format, 129

list, 126

TIA Portal, 31

drzewo narzędzi, 33

karta zadań, 34

kreator ekranów, 39

obszar roboczy, 33

okno nadzoru, 34

pasek edycji, 34

pasek narzędzi, 33

pierwszy projekt, 37
zmiana widoku
 środowiska, 34
Toolbar, 141
transfer projektu, 28
tworzenie ekranu, 48

U

uruchamianie, 17, 121
 symulatora, 53
User
 administration, 45, 144
 groups, 145
ustawienia domyślne, 52
użytkownicy, 143

V

Visibility, 153

W

warstwa
 błędów, 75
 Error, 72
 Hardware, 61
 logiczna, 90
 sprzętowa, 61
 wizualizacji, 71
widok Portal view, 31, 33
wizualizacja, 44, 71
 panelu, 155
właściwości obiektu, 124
właściwość Visibility, 153
wylogowanie, 147

Z

zakładka
 Appearance, 128
 Attributes, 135

Events, 124
General, 128
Groups, 145
Layout, 129
Miscellaneous, 133
Text format, 129
Toolbar, 141
User administration, 144
User groups, 145
zapisywanie domyślnych
 ustawień, 52
zarządzanie użytkownikami,
 144
zasilanie, 19
zdarzenie, 142, 144
 SetLanguage, 152
 SetTag, 154
zestaw startowy, 56
złącza, 18
zmiana języka, 152

PROGRAM PARTNERSKI

— GRUPY HELION —

- 
1. ZAREJESTRUJ SIĘ
 2. PREZENTUJ KSIĄŻKI
 3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion

W kontakcie z inteligentnym domem

Tak jak do komunikacji między człowiekiem i komputerem służą klawiatura i ekran, tak do porozumiewania się człowieka z linią technologiczną czy skomplikowanym systemem elektronicznym służy panel sterowniczy. W literaturze przedmiotu oraz w praktyce przemysłowej oznacza się go angielskim akronimem HMI (Human-Machine Interface). Panel, mający postać płaskiej powierzchni — dawniej z przyciskami, pokrętkami i licznikami, dziś z wyświetlaczem — pozwala odczytywać komunikaty systemu i wydawać mu polecenia, podłączać urządzenia zewnętrzne oraz wizualizować sam system w formie uproszczonego schematu graficznego.

Ta książka stanowi kompleksowy kurs tworzenia wizualizacji nowoczesnych paneli dotykowych HMI firmy Siemens. Rozdział po rozdziale autor pokazuje funkcje urządzenia, które trzeba poznać, by móc efektywnie wdrożyć projekt do zarządzania inteligentnym domem. Nie oznacza to, że z podręcznika skorzystają jedynie wykonawcy tego typu systemów — aktualnie panele sterownicze znajdują się w praktycznie każdej maszynie realizującej procesy technologiczne lub produkcyjne. Bardzo wiele z nich pracuje pod panelami SIMATIC firmy Siemens.

- Konfiguracja i programowanie paneli HMI
- Praca z ekranami
- Tworzenie scenariuszy (receptury)
- Odczytywanie komunikatów (alarmy, raporty)
- Zarządzanie użytkownikami
- Łączenie większej liczby paneli
- Bezpieczeństwo systemu i diagnozowanie problemów

Partner projektu:

SIEMENS
Ingenuity for life

KOD KORZYŚCI
Sięgnij po więcej! ►



ISBN 978-83-283-3776-3



9 788328 337763

Helion



helion.pl



HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

Sprawdź nasze szkolenia!



AKADEMIA IT & BUSINESS

HELIONSZKOLENIA.PL

INFORMATYKA W NAJLEPSZYM WYDANIU

Cena: 44,90 zł