Ćwiczenia z S7-1200

Komunikacja S7-1200 z przyciskowym panelem HMI KP300 PN

FAQ · Marzec 2012



Przykłady i Aplikacje

SIEMENS

www.siemens.pl/S7-1200

Spis treści

1	Opis za	gadnienia poruszanego w ćwiczeniu	3
	1.1	Wykaz urządzeń	3
2	Konfig	uracja S7-1200 PLC oraz HMI KP300 PN	4
	2.1	Nowy projekt	4
	2.2	Dodawanie CPU do projektu	4
	2.3	Dodawanie panelu HMI do projektu	5
	2.4	Ustawienie połączenia w PG/PC Interface	6
3	Progra	nowanie sterownika i panelu HMI	7
	3.1	Konfiguracja zmiennych	7
	3.2	Program sterownika	8
	3.3	Wizualizacja na panelu KP300 PN	10

1 Opis zagadnienia poruszanego w ćwiczeniu

W ćwiczeniu poruszony będzie temat stworzenia prostej aplikacji z wizualizacją na panelu przyciskowym KP300 PN. Zadaniem będzie napisanie aplikacji sterującej rozruchem silnika metodą gwiazda – trójkąt. W przykładzie będą przedstawione czynności programisty przy dodawaniu panelu HMI do projektu, skomunikowaniu go ze sterownikiem, a także wymianie danych pomiędzy stacją operatorską, a sterownikiem PLC za pośrednictwem sieci Profinet. Przedstawiona będzie też konfiguracja alarmów wywołanych stanami wysokimi na wejściach cyfrowych sterownika oraz wykorzystanie funkcjonalności zmiany podświetlenia panelu.

1.1 Wykaz urządzeń

Hardware

Lp.	Urządzenie	llość	Numer katalogowy
1.	Sterownik SIMATIC S7-1200, model CPU 1212C AC/DC/RLY	1	6ES7212-1BD30-0XB0
2.	Panel KP300 PN mono 3"	1	6AV6647-0AH11-3AX0
3.	Zasilacz PM 1207 (24V DC / 2,5A)	1	6EP1332-1SH71
4.	Switch Ethernet CSM 1277	1	6GK7277-1AA10-0AA0
5.	Kabel Ethernet 6m (komunikacja sterownika z panelem HMI oraz PG/PC)	3	6XV1870-3QH60

Software

Lp.	Nazwa	llość	Numer katalogowy
1.	Step 7 Basic v11	1	6ES7822-0AA01-0YA0

2 Konfiguracja S7-1200 PLC oraz HMI KP300 PN

2.1 Nowy projekt

Podczas tworzenia nowego projektu, należy nadać mu nazwę, ścieżkę jego lokalizacji na dysku twardym komputera, opcjonalnie autora i komentarz, następnie zatwierdzić przyciskiem **Create**.

Start			Create new project	
Devices & networks	A	Open existing project	Project name Komunikacja z KP300 Path D157-1200/Programy	
PLC programming		 Create new project Migrate project 	Author Siemens Comment	^
Visualization			×	Create
Online &		initial N	l l	create

2.2 Dodawanie CPU do projektu

Przy dodawaniu nowego urządzenia w widoku *Portal view* trzeba wybrać opcję **Configure a device**, następnie **Add new device**, potem rodzaj urządzenia (w tym przypadku sterownik PLC) i model urządzenia. Po tym należy zatwierdzić konfigurację, klikając przycisk **Add**.

Start		First steps
Devices &	Open existing project	Project: "Komunikacja_KP300" was opened successfully. Please select the next s
PLC programming	 Create new project Migrate project 	Start
Visualization	Close project	Devices & Configure a device
Singliosites -	 Welcome Tour First steps 	PLC programming Write PLC program
	Installed software	Visualization Configure an HMI screen
	Help	
	🚱 User interface language	Project view Open the project view
Project view	Opened project: C:\Documents and	Settinos\Administrator\Pulpit\Komunikacia KP300\Komunikacia KP300

🕻 Siemens - Komunikacja_KTP600	PN					_ # X
					Totally Integrated Autom F	ation ORTAL
Start 🦓		Add new device _				
Devices & state of the second	Show all devices	PLC_1	- Da no			^
PLC programming Visualization Online & Diagnostics	Configure networks	PLC PLC	() SMARC 57-1200 () SMARC 57-1200 () CU () CU	Order no : Viction: Description: Work memo with Dis x24 Al2 on board (expandable pipaboard Dis X24 for serial foo modules for instructions;	CPU 1212C ACDCRJy CPU 1212C ACDCRJy 6E57 212-18D30-0XB0 V21 25 KB: 120/240VAC power supply 304 SINK/SOURCE, D06 x relay and 4.4 Min Expeed counters set this due of signal board and 2 ts on board signal board expands by the 3 continuications modules minumication in bio 2 signal Sind Sink Sink Sink Sink Sink Sink Sink Sink	-
	• Help	Open device view		programmin communica	rg, Hull and PLC to PLC	dd >

2.3 Dodawanie panelu HMI do projektu

Dodając panel HMI w widoku *Project view*, należy kliknąć dwa razy lewym przyciskiem myszy w polu **Project tree** na **Add new device**, w nowym oknie dialogowym trzeba wybrać odpowiedni panel, następnie zaznaczyć opcję **Start device wizard** i potwierdzić klikając przycisk **OK**.



W oknie dialogowym *HMI Device Wizard* trzeba wybrać sterownik PLC, który będzie się z nim komunikował. Następnie kliknąć *Finish*, aby zakończyć konfigurację HMI lub *Next*, aby kontynuować parametryzację ustawień ekranu.

HMI Device Wizard: KP300) Basic mono PN		×	4
	PLC connections Configure	the PLC connection(s).		
PLC connections Screen layout	•			
Alarms	•			
Screens	• <u> </u>	Communication driver:		
System screens	<u>ا المسالم</u>	Interface:		
	HMI_1 KP300 Basic mono PN		Select PLC	
			Browse	
			\sim	
				k
				$\left \right\rangle$
Save settings		≪ <u>B</u> ack <u>H</u> ext ≫	<u>E</u> inish <u>C</u> ancel	
			Name	CPU type
			None PLC 1	CPU 1212C ACIDCIPIU
				CFO T2T2C AODORIY

2.4 Ustawienie połączenia w PG/PC Interface

Ustawienie odpowiedniego połączenia w **Set PG/PC Interface** zapewnia komunikację między sterownikiem PLC a symulacją Runtime (RT).

Możliwe jest przeprowadzenie symulacji pracy HMI z PLC, jeżeli komputer jest połączony ze sterownikiem S7-1200, bez konieczności fizycznego posiadania panelu.

Konfigurację tę wprowadza się w **Panelu sterowania.** W tym celu należy kliknąć na **Set PG/PC Interface.** W zakładce **Access path** jako punkt dostępu **Access Point of the Aplication** trzeba ustawić **S7ONLINE** (STEP 7) -> TCP/IP ->Karta sieciowa.

🕑 Panel sterowania						
Plik Edycja Widok Ulubione Narze	ędzia Pomoc				Set PG/PC Interface	X
🕲 Wstecz 🔹 🕥 🔹 🏂 🔎	Wyszukaj 🔀 Foldery [Syncl	hronizacja fol	derów	Access Path LLDP / DCP	
Adres 📴 Panel sterowania					Access Point of the Application:	
Panel sterowania	Aktualizacje Broadcom A automativenje Configurati	SF Centrum on zabezpieczeń	Czcionki	Data i godzin	S70NLINE (STEP 7)> TCP/IP -> Broadc (Standard 7, STEP 7) Interrace Parameter Assignment Used:	om NetXtreme 57x. 💌
Zobacz też 🛞	Kreator konfigur	ci Memory Card	💕 Mowa	hysz	TCP/IP → Broadcom Net/Vreme 57x <a WPLCSIM(PROFIBUS) WPLCSIM(TCP/IP) WR S7USB</a 	Properties Diagnostics Copy
Pomoc i obsługa techniczna	Pasek zadań i menu Start Poczta	Połączenia sieciowe	Set PG/PC Interface	SigmaTel Audio	CCP/IP > Broarcom NetXtreme 57 C	Delete Select
					ОК	Cancel Help

3 Programowanie sterownika i panelu HMI

3.1 Konfiguracja zmiennych

W **Project tree** rozwinąć gałąź **PLC tags**, następnie wybrać **Default tag table** i utworzyć zmienne PLC. Ważne, żeby pamiętać o odpowiednim do typu danych ich zaadresowaniu (zmienne typu Bool muszą mieć adresy bitowe, np. 10.0, zmienne Word muszą mieć adresy o wielkości słowa, np. MW10).

Project tree		Komu	nikacja_KP300 → PLC_1 [CPU 12120	CAC/DC/Riy] → F	PLC tags	 Default tag 	table [25]				
Devices							-	a Tags	🗉 User con			
B O O	1	*	🔮 🖻 😤 🕱									
		D	Default tag table									
🔻 🛅 Komunikacja_KP300	^		Name	Data type	Address	Retain	Visibl	Acces C	mment			
📑 Add new device		1	🐵 Przycisk Start	Bool 🛽	%10.0							
💑 Devices & networks		2	💷 Przycisk Stop	Bool	%10.1							
PLC_1 [CPU 1212C AC/DC/Rly]		3	💷 Przekaźnik termiczny	Bool	%10.2							
🕅 Device configuration		4	💷 Przekaźnik termistorowy	Bool	%10.3							
🛂 Online & diagnostics		5	💷 Stycznik sieciowy	Bool	%Q0.0							
🗢 😹 Program blocks		6	💷 Stycznik gwiazda	Bool	%Q0.1							
🗳 Add new block		7	💷 Stycznik trójkąt	Bool	%Q0.2							
📲 Main [OB1]		8	💷 Wykrycie zbocza opadającego	Bool	%M0.0							
Alarmy [OB123]		9	💷 Stany alarmowe	Word	%MW10							
🕨 🎆 System blocks		10	💷 Alarm od przekaźnika termicznego	Bool	%M11.0							
🕨 🎇 Technology objects		11	🚳 Alarm od przekaźnika termistorowego 🗌	Bool	%M11.1							
🕨 🔚 External source files		12	<add new=""></add>				V	V				
🕶 浸 PLC tags												
🍇 Show all tags							_		•			
Add new tag table												
🎬 Default tag table [25]												
▶ 🗊 PLC data tvpes												

3.2 Program sterownika

Program sterownika składa się z dwóch bloków organizacyjnych wywoływanych cyklicznie. Blok organizacyjny OB1 (Main) jest głównym blokiem programu i zawiera algorytm sterowania silnikiem z rozruchem typu gwiazda-trójkąt. Blok organizacyjny OB123 (Alarmy) odpowiada za wyświetlanie alarmów na panelu w przypadku przeciążenia silnika lub zbyt wysokiej jego temperatury.

W bloku OB1 należy napisać algorytm sterowania jak na poniższym przykładzie.





W programie wykorzystano instrukcje timerów TON (opóźnione załączanie) i TOF (opóźnione wyłączanie). Nastawy timerów dla uproszczenia programu wpisano bezpośrednio do instrukcji w postaci: *T#xxyy*, gdzie xx to wartość liczbowa nastawy timera, a yy to jednostka nastawy (ms, s, m lub h). Nastawy te można również przypisać wykorzystując zmienne timerów (np. *"Czas przełączenia".PT*) lub dowolne stworzone wcześniej zmienne typu Time.

Na potrzeby programu stworzono instrukcję wykrywania zbocza opadającego, dzięki której sterownik wykrywa moment wyłączenia wyjścia timera z czasem przełączenia pomiędzy stycznikiem gwiazdowym, a stycznikiem konfiguracji typu trójkąt. Podczas wykonywania algorytmu sterownik cyklicznie porównuje w tej funkcji stan bitu wyjściowego timera ze zmienną pomocniczą *Wykrycie zbocza opadającego*. Stan logiczny zmiennej pomocniczej jest też cyklicznie aktualizowany podczas wykonywania programu. Gdy bit wyjściowy timera będzie miał wartość 0 i jednocześnie bit pamięci pomocniczej będzie miał wartość 1, instrukcja poda sygnał logiczny powodujący uruchomienie wyjścia Q0.2 (*Stycznik trójkąt*). Podczas tworzenia aplikacji należy pamiętać, że do każdej instrukcji wykrywania zbocza należy przypisać oddzielny bit pamięci pomocniczej, ponadto nie może być on wykorzystywany w żadnym innym miejscu w programie, gdyż w przeciwnym wypadku zbocza mogą nie być wykrywane poprawnie.

Po stworzeniu algorytmu w bloku OB1, należy dodać nowy blok organizacyjny wywoływany cyklicznie, nazwać go *Alarmy*, następnie wewnątrz tego bloku napisać kolejną część programu tak jak na poniższym przykładzie.



3.3 Wizualizacja na panelu KP300 PN

Przykładowa aplikacja wizualizacyjna na panelu KP300 PN będzie zawierała ekran z informacją o aktualnym statusie sterowanego silnika. Ponadto skonfigurowane zostaną dwa stany alarmowe wywołane przez sygnały na wejściach binarnych sterownika. Alarmy będą powodowały zmianę kolory podświetlenia ekranu na kolor czerwony.

Należy rozwinąć w drzewie projektu gałąź *HMI_1 (KP300 Basic mono PN*), a następnie w gałęzi *Screens* otworzyć ekran *Root screen*. W tym ekranie można zmienić tekst powitalny na tekst : *Status silnika*.



Następnie należy stworzyć w polu **Texts and graphic lists** listę tekstową o nazwie **Stan pracy silnika** i edytować ją jak na poniższym przykładzie.

Project tree		Komun	ikacja_	_KP300 >	нмі_	1 [KP300 Basic mo	10 PN] → Text	and graph	hic lists	_∎≡×
Devices									🗄 Text lists	🚡 Graphic lists
B 0 0	1									
		Tex	t lists							
📑 Add new tag table	^		Name .	•			Selection	Co	mment	
🍯 Default tag table [27]		1-2-	Stan pra	acy silnika			Value/Range	-		
PLC data types			<add ne<="" td=""><td>:w></td><td></td><td></td><td></td><td></td><td></td><td></td></add>	:w>						
Watch and force tables										
🔤 Program info										
🛅 Text lists		/								
🕨 🧊 Local modules										
HMI_1 [KP300 Basic mono PN]		Tex	t list e	ntries						
时 Device configuration	/		Default	Value •		Text				
😼 Online & diagnostics		1.		0	-	Silnik wyłaczony				
🍸 Runtime settings		1	Õ	1		Uruchamianie silnika				
🕶 🛅 Screens		1.	ŏ	2		Silnik uruchomiony				
📑 Add new screen			0	<add news<="" td=""><td>-</td><td></td><td></td><td>_</td><td></td><td></td></add>	-			_		
Noot screen										
🕨 🗑 Screen management										
🕨 🔚 HMI tags										
🍡 Connections										
🖂 HMI alarms	_									
🗐 Recipes	=									
5 Scheduled tasks										
🔛 Text and graphic lists										
🙀 User administration										
🕨 🙀 Common data										
Documentation settings										
🕨 🐻 Languages & resources										

W oknie *Root screen* wstawić pole *Symolic I/O field* i sparametryzować je jak w prezentowanym przykładzie. W ten sposób na ekranie HMI będą wyświetlały się odpowiednie stany pracy silnika (wartość zmiennej *Status silnika* aktualizowana jest w networkach 5, 6 i 7 w obrębie bloku *OB1*).



Następnie w oknie *HMI alarms* należy dodać dwa alarmy dyskretne i powiązać je ze zmienną ze sterownika *Stany alarmowe*, tak jak w prezentowanym przykładzie. Zmienna *Stany alarmowe* obejmuje bity alarmowe od przekaźnika termicznego i przekaźnika termistorowego. Należy zwrócić uwagę, że zmienna ta jest typu Word, a bity alarmowe nie zaczynają się od bitu LSB tej zmiennej, tylko od bitu 8 (adres zmiennej to MW10, natomiast bity alarmowe mają adresy M11.0 i M11.1).

Project tree		Komunikacja_KP300 → HMI_1 [KP300 Basic mono PN] → HMI alarms _ 🗕 🖬	X Tasks	
Devices		🙀 Discrete alarms 🛛 🙀 Analog alarms 🛛 🖶 System events 🖓 Alarm classes 🔃 Alarm grou	ips Options	
B 0 0	B			Tas
		Discrete alarms	✓ Find and re	place 8
Alarmy [OB123]	^	ID Alarm text Alarm class Trigger tag Trigger, Trigger address HMI	acknow	
🕨 🔙 System blocks		🙀 1 💦 Alarm od przekaźnika termicznego 🛛 Errors Stany alarmo 0 Stany alarmo 🕫	tag> Find:	<u> </u>
Technology objects		🙀 2 🔹 Alarm od przekaźnika termistorowego 🛛 Errors 🔤 armowe 🗐 🚛 1 🔹 Stany alarmo 🗇 o	tag>	▼ ii
🕨 🔙 External source files				
🕶 浸 PLC tags			Y	
🍇 Show all tags			D	ata type Address
💣 Add new tag table		Reprogram blocks		
🍯 Default tag table [27]	1	🖌 🔤 Teap glogy objects 🛛 🚳 Stany	alarmowe W	Vord 🔳 %M 💌
🕨 💽 PLC data types				
Watch and force tables		🗧 Default tag table [27]		
📴 Program info	/			
Text lists	/	▼ 🔂 HMI_1 [KP300 Basic mono PN]		
Image:		👻 🔁 Hhill tags		
HMI_1 [KP300 Basic mono PN]		▼ 📽 Default tag table [2]		
🛐 Device configuration	=			
😧 Online & diagnostics				
🍟 Runtime settings		Show all	Bi Add obi	iect 🖌 🖌 🖌
🗢 📄 Screens		Discrete_alarm_2		
📑 Add new screen		Properties		ma
💽 Root screen		rioperues	Deale as with	
🕨 🕎 Screen management		General	 Replace with: 	
🕨 🛺 HMI tags		General		
2. Connections		Trigger	Replace	Replace all
MII alarms		Tooltip Alarm text Alarm od przekaźnika termistorowego		
- Kecipes		Acknowledgment	✓ Languages	& resources
5 Scheduled tasks			Editing language	ae:
🔛 Text and graphic lists	~	Alarm class: Errors III	Calling anguag	
> Details view		Alarm group: <10 alarm group> 🗏	✓ <	>

Po utworzeniu alarmów i ich skonfigurowaniu, należy kliknąć na wiersz z pierwszym alarmem i w jego właściwościach wybrać zakładkę *Events*. W polu *Incoming* dodać zdarzenie *SetBacklightColor* i wybrać jego wartość jako *Red*. W polu *Acknowledge* dodać takie samo zdarzenie, natomiast jego wartość określić jako *White*. Taką samą procedurę zastosować do drugiego alarmu. W ten sposób wystąpienie alarmu spowoduje podświetlenie ekranu na kolor czerwony, natomiast potwierdzenie alarmu podświetli ekran z powrotem na domyślny kolor biały.

Po wszystkich powyższych czynnościach można wgrać program do sterownika klikając prawym przyciskiem myszy na jego folderze w drzewie projektu i wybierając z menu kontekstowego **Download to device -> All**. Następnie wgrać program do panelu klikając prawym przyciskiem myszy na folderze panelu i wybierając z menu kontekstowego **Download to device -> Software (all).** Można też wgrać program zaznaczając urządzenie i klikając ikonę **Download to device**.

