

*dr inż. Bogdan Broel-Plater\**

*dr hab.inż., prof. PS Wieńczysław Daca\**

*mgr inż. Krzysztof Sozański\*\**

*mgr inż. Jan Szumski\*\**

*\* Instytut Automatyki Przemysłowej Politechniki Szczecińskiej*

*\*\* Ośrodek Badawczo-Rozwojowy Metrologii Elektrycznej w Zielonej Górze*

## **STEROWNIK PROGRAMOWALNY NA BAZIE MIKROKONTROLERA SAB80C166\*\*\***

*Streszczenie:*

*W pracy przedstawiono sterownik programowalny SA100. Został on opracowany we współpracy Instytutu Automatyki Przemysłowej Politechniki Szczecińskiej i Ośrodka Badawczo-Rozwojowego Metrologii Elektrycznej w Zielonej Górze. Cechą charakterystyczną sterownika jest ukierunkowanie na sterowanie procesami termicznymi.*

*Abstrakt:*

*In the paper the programmable logic controller SA100 is presented. It has been developed jointly by the Institute of Control Engineering and the Research and Development Center for Electrical Measurement. The SA100 controller is especially designed for control of thermal processes.*

### **1. WPROWADZENIE**

Celem niniejszej pracy jest przedstawienie sterownika programowalnego SA100. Sterownik ten opracowany został wspólnie przez Instytut Automatyki Przemysłowej Politechniki Szczecińskiej (IAP PS) oraz Ośrodek Badawczo-Rozwojowy Metrologii Elektrycznej (OBR ME) w Zielonej Górze. W bieżącym roku sterownik zostanie przez OBR ME wprowadzony do produkcji.

Rynek sterowników programowalnych (ang. PLC, niem. SPS) rozwija się w ostatnich latach bardzo dynamicznie. Sterowniki programowalne należą obecnie do podstawowych urządzeń automatyki przemysłowej. Pierwotnym zastosowaniem sterowników programowalnych była realizacja układów przełączających operujących wyłącznie na sygnałach binarnych. Obecne sterowniki są w stanie przetwarzać zarówno sygnały binarne jak i ciągłe. W obszarze dużych sterowników programowalnych następuje ich integracja z cyfrowymi programowalnymi systemami automatyzacji wielkości procesów ciągłych. Sterowniki małe przejmują często funkcje kompaktowych regulatorów procesowych.

Jeszcze przed 5 laty sterowniki programowalne były urządzeniami mało znanymi w polskiej automatyce przemysłowej. Obecnie sytuacja się zmieniła. Na rynku polskim oferowane są sterowniki programowalne wszystkich czołowych producentów światowych. Przykłady najbardziej rozpowszechnionych w Polsce rodzin sterowników oraz ich producenci przedstawieni są w tab. 1.

---

\*\*\* prace wykonywane w ramach projektu celowego nr 8 T11 A 002 95 C/2475  
dofinansowywanego przez Komitet Badań Naukowych w latach 1996/97

L.P.	Nazwa rodziny	Producent
1	Melsec	Mitsubishi
2	Modicon	AEG/Schneider
3	Simatic	Siemens
4	Sysmak	Omron
5	SUCO	Klöckner-Mueller
6	TSX	Telemachanique

Tabela 1. Wybrane rodziny sterowników swobodnie programowalnych oferowane na rynku polskim

Na etapie opracowywania założeń sterownika SA100 pojawiło się istotne pytanie: jakimi cechami funkcjonalnymi charakteryzować się powinien projektowany sterownik, aby przy istniejącej konkurencji miał on szansę znaleźć swoje miejsce na rynku?

Podstawowy warunek to wymóg takiego poziomu technologicznego projektowanego sterownika (w zakresie sprzętu i oprogramowania) aby był on porównywalny z produktami oferowanymi na rynku polskim przez czołowe firmy zachodnie. Jest to warunek niewystarczający, bowiem za produktami dużych firm światowych przemawia przede wszystkim ich renoma.

Przy projektowaniu sterownika SA100 przyjęto zatem szereg dodatkowych założeń dotyczących funkcji projektowanego sterownika. Najważniejsze z nich to:

- sterownik SA100 ukierunkowany jest na sterowanie procesami termicznymi, w szczególności elektrogrzejnymi,
- sterownik posiada strukturę modułową (moduł podstawowy, moduły peryferyjne, konsola operatorska) komunikujące się ze sobą łączem szeregowym RS485,
- sterownik SA100 posiada proste i przyjazne dla użytkownika możliwości programowania.

Założenie pierwsze oznacza, że bazując wyłącznie na module podstawowym oraz konsoli sterownika istnieje możliwość budowy systemu sterowania zawierającego m.in. 12-kanalowy samonastrajający regulator temperatury. Czujniki temperatury typu termopara lub termoopornik przyłączane są przy tym bezpośrednio do odpowiednich zacisków modułu podstawowego sterownika. Moduł podstawowy wyprowadza także sygnały sterujące z regulatorów w różnej formie w zależności od typu urządzenia wykonawczego z którym współpracuje dany kanał regulatora (sygnał ciągły, sygnał binarny typu PWM, sygnał sterowania elementu wykonawczego z silnikiem nawrotnym). W standartowych sterownikach programowalnych realizacja takiego układu sterowania wymaga z reguły zastosowania wielu modułów sprzętowych.

Założenie drugie oznacza możliwość dopasowania struktury sprzętowej sterownika do sterowanego procesu oraz rozproszenia przestrzennego poszczególnych modułów.

Założenie trzecie oznacza minimalizację wymogów sprzętowych dla programowania sterownika. Odbyna się ono przy zastosowaniu standardowego komputera osobistego (PC). Ponadto w przeciwieństwie do wielu standartowych sterowników programowalnych program użytkownika przetwarzany jest w sterowniku SA100 na zasadzie interpretacji a nie kompilacji. Dzięki temu zbiór narzędzi programowych dostarczanych użytkownikowi wraz ze sterownikiem SA100 w celu jego programowania jest bardzo prosty (jeden program wykonawczy typu "exe" oraz jeden zbiór typu "help").

## 2. FUNKCJE STEROWNIKA

Schemat funkcjonalny sterownika SA100 przedstawiony został na rys. 1. Użytkownik pisząc swój program może w sposób dowolny operować na zasobach sterownika. Są one dostępne użytkownikowi poprzez przypisane im nazwy symboliczne. Tabela 2. zawiera przegląd tych zasobów. Operowanie na zasobach sterownika odbywa się przy pomocy zestawu 101 rozkazów. Przegląd tych rozkazów zawiera tabela 3.

Oprogramowanie sterownika PLC wykorzystuje technikę interpretacyjną tzn. że system operacyjny sterownika PLC jest wyposażony w interpreter rozkazów programu użytkownika zapisanych w specjalnym kodzie pośrednim. Program edycyjny - uruchomiony na komputerze osobistym - po sprawdzeniu poprawności formalnej programu użytkownika dokona translacji tego programu do postaci pośredniej zastępując wszystkie rozkazy odpowiadającymi im kodami pośrednimi i uzupełniając je o adresy względne zmiennych i etykiet. Lista rozkazów języka zaimplementowanego w sterowniku umożliwia użytkownikowi napisanie:

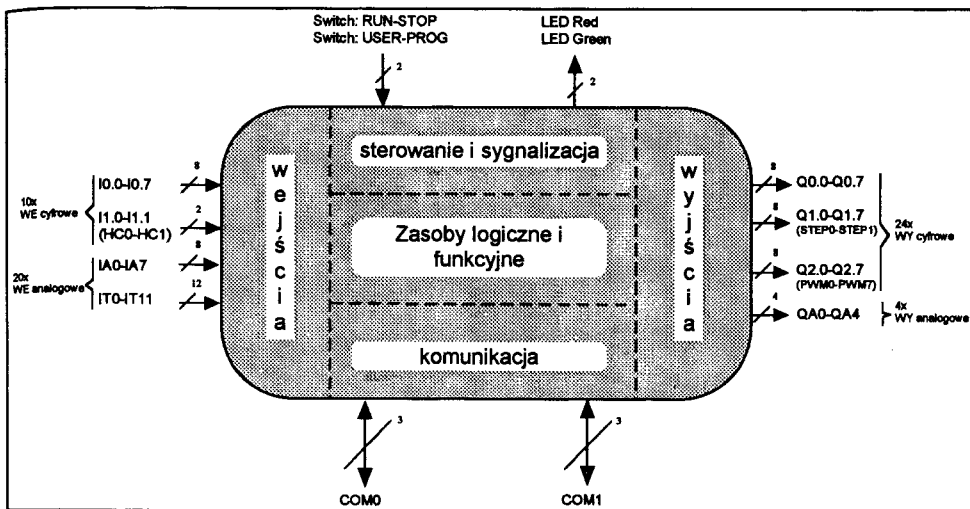
- programu głównego wykonywanego cyklicznie,
- programu uaktywnianego przerwaniem licznikowym INT\_HC,
- programu uaktywnianego przerwaniem czasowym INT\_TIM,
- programu reakcji zatrzymanie wykonywania programu użytkownik lub na wystąpienie błędu systemowego w pracy sterownika .

Każdy z tych programów składa się z sekwencji rozkazów wybranych przez użytkownika z pełnej listy rozkazów sterownika. Wymienione cztery programy składają się na kompletny program użytkownika. Struktura programu użytkownika przedstawiona jest na rys. 2.

Elementami narzuconymi w programie są:

- |                |   |  |
|----------------|---|--|
| <b>MAIN</b>    | - | etykieta oznaczająca początek programu głównego, musi opisywać pierwszą linię programu, program edycyjny dopisuje ją samoczynnie po otwarciu nowego pliku, |
| <b>INT_HC</b>  | - | etykieta oznaczająca początek programu obsługi przerwania od szybkiego licznika sprzętowego HC0,   |
| <b>INT_TIM</b> | - | etykieta oznaczająca początek programu obsługi przerwania od tajmera programowalnego TIM0,   |
| <b>STOP</b>    | - | etykieta oznaczająca początek programu wywołanego zatrzymaniem pracy sterownika lub wystąpieniem błędu systemowego w pracy sterownika                      |
| <b>END</b>     | - | rozkaz sygnalizujący koniec treści programu, musi być w ostatniej linii programu.  |

Poprawność struktury programu kontrolowana jest samoczynnie każdorazowo przed jego wysłaniem do sterownika lub na żądanie piszącego program poprzez użycie odpowiedniego polecenia w menu programu edycyjnego.



Rys. 1. Schemat funkcjonalny sterownika SA100

• Zasoby fizyczne

Nr	oznaczenie	nazwa	ilość
1	I	WE binarne	10
2	Q	WY binarne	24
3	IA	WE analogowe	8
4	IT	WE temperaturowe	12
5	QA	WY analogowe	4

• Zasoby logiczne

Nr	oznaczenie	nazwa	ilość
1	F	znaczniki bitowe	2048 x 1 bit
2	D	rejstry pomocnicze danych	256 x 8 bit
3	RO	zmienne systemowe informacyjne	32 x 8 bit
4	RW	zmienne systemowe sterujące	32 x 8 bit
5	K	stałe liczbowe	0-255, 0-65535

• Zasoby funkcyjne

Nr	oznaczenie	nazwa	ilość
1	COM	łącze komunikacyjne szeregowe	2
2	HC	szybkie liczniki sprzętowe	2
3	TIM	tajmery programowalne	32
4	STEP	sterownik silnika nawrotnego	2
5	PWM	sterownik wypełnienia impulsów	8
6	PID	samonastrajający regulator PID	12

Tabela 2. Przegląd zasobów sterownika SA100

Grupa	Nazwa grupy	Charakterystyka grupy	Przykładowe mnemoniki	Liczba rozkazów grupy
1	Rozkazy transferu danych i ładowania stałych	Transfer danych typu byte lub word w całym obszarze zasobów. Ładowanie stałych do danych typu bit, bajt lub słowo.	MOVW LKB ZERW BS	8
2	Rozkazy bitowe	Operacje bitowe w całym obszarze zasobów wykorzystujące stos logiczny	OP NEG	19
3	Rozkazy logiczne	Operacje typu AND, OR i XOR na danych typu bajt lub słowo	ORW NAB	14
4	Rozkazy arytmetyczne	Operacje typu ADD, SUB, MUL i DIV na danych typu bajt lub słowo	ADDW DECB	12
5	Rozkazy porównań	Porównania arytmetyczne danych typu byte lub word z ustawianiem stosu logicznego	CPEW CPGEB	12
6	Rozkazy przesunięć i obrotów	Operacje przesuwania i obrotu na pojedynczych danych typu bajt i słowo	SLW RRB SWAPB	10
7	Rozkazy konwersji kodów	Operacje konwersji kodów w zakresie: kod binarny - ASCII	WCA5 A3CB	4
8	Rozkazy organizacyjne	Rozkazy sterujące przebiegiem programu użytkownika	JU PEZ	22

razem

101

Tabela 3. Przegląd listy rozkazów sterownika SA100

<b>MAIN:</b>	...
	...
	...
	...
	PEU/PEZ/PEN/...
<b>INT_HC:</b>	...
	...
	...
	...
	RETU/RETZ/RETN/PEU/PEZ/PEN/...
<b>INT_TIM:</b>	...
	...
	...
	...
	RETU/RETZ/RETN/PEU/PEZ/PEN/...
<b>STOP:</b>	...
	...
	...
	...
	RETU/RETZ/RETN/PEU/PEZ/PEN/...
	<b>END</b>

Rys. 2. Struktura programu użytkownika w sterowniku SA100

### 3. OPIS BUDOWY SPRZĘTOWEJ

W części sprzętowej sterownika SA100 zastosowano nowoczesne rozwiązania sprzętowe. Centralne funkcje w sterowniku pełni 16-bitowy mikrokontroler SAB80C166.

Mikrokontroler SAB80C166 wprowadzony został na rynek w roku 1991. Główne zalety tego mikrokontrolera to:

- bardzo wydajna 16-bitowa CPU o strukturze RISC pracująca z użyciem techniki PIPELINE,
- najszybszy z istniejących obecnie (400 ns. typowo) system obsługi przerw (m.in. w trybie zbliżonym do DMA) zawierający 16 priorytetów z 4 grupami każdy,
- rozbudowany zestaw wbudowanych urządzeń peryferyjnych takich jak:
  - 10 kanałów przetwornika AC o rozdzielczości 10 bit,
  - 7 tajmerów 16 bitowych,
  - 16 kanałów jednostki porównująco-przechwytyjącej,
  - 2 kanały łącza szeregowego.

SAB80C166 jest obecnie najbardziej wydajnym mikrokontrolerem w zakresie zastosowań czasu rzeczywistego.

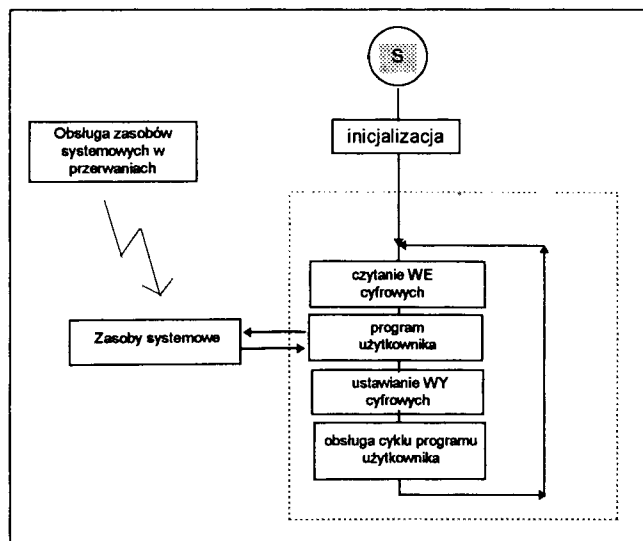
Wejścia i wyjścia sterownika realizowane są w zasadzie na bazie wewnętrznych układów peryferyjnych SAB80C166. Jedynie wejścia temperaturowe dostosowane do bezpośredniego przyłączenia różnego typu czujników temperatury wykorzystują zewnętrzny 16-bitowy układ akwizycji danych analogowych AD7715 firmy Analog Devices. Wejścia (z wyjątkiem wejść analogowych wykorzystujących wewnętrzny przetwornik AC układu 80C166), wyjścia a także łącza komunikacyjne posiadają pełną izolację galwaniczną.

Oprogramowanie systemowe sterownika, dane nieulotne a także wprowadzany łączem RS232C z komputera osobistego program użytkownika przechowywane są w pamięci typu FlashEPROM.

Sterownik wykonany jest w technologii montażu powierzchniowego (SMD).

### 4. OPIS OPROGRAMOWANIA SYSTEMOWEGO

Oprogramowanie systemowe sterownika napisane zostało w języku C przy zastosowaniu pakietu rozwojowego firmy Keil C166, wersja 3.0. Rysunek 2 przedstawia strukturę tego programu.



Rys.3. Struktura oprogramowania systemowego sterownika SA100

## 5. PRZYKŁADOWY PROGRAM UŻYTKOWNIKA

W tabeli 4. przedstawiony został przykładowy program użytkownika. Realizuje on regulację temperatury w pojedynczej pętli regulacji. Program ten ilustruje zasadę obsługi zasobów systemowych sterownika - do których należy także regulator PID - w tle programu użytkownika. Konfigurowanie całej pętli regulacji następuje tylko jednokrotnie w pierwszym cyklu programu użytkownika. W następnych cyklach program użytkownika nie wykonuje już żadnych operacji związanych z regulacją. Raz zainicjalizowany regulator PID pracuje już samoczynnie w tle programu użytkownika.

Regulator samoczynnie co okres czasu TA pobiera wielkość regulowana, tj. temperaturę z kanału IT0 i po wyliczeniu sygnału sterującego odsyła go samoczynnie do modulatora szerokości impulsów PWM0. Modulator ten pracuje także w tle programu użytkownika.

MAIN:	Lp	f255.0	;czy pierwszy cykl programu ?
	Jn	koniec	;nie - skocz na koniec programu
	Bs	f255.0	;ustawienie semafora pierwszego cyklu programu
	LkB	pidb3,kb20	;jednorazowe konfigurowanie regulatora PID
	LkB	pidb15,kb0	;adres wartości zadanej - rejestrów DW20
	LkB	pidb27,kb0	;adres wartości regulowanej - temperatura IT0
	LkW	dw20,kw2048	;adres sygnału sterującego - sterownik PWM0
	LkB	pidb39,kb100	;wartość zadana 50% = 200 °C
	LkB	pidb51,kb10	;KP=10
	LkB	pidb63,kb6	;Td=10 sek
	LkB	pidb75,kb2	;Ti=60 sek
	LkB	pidb87,kb0	;Tp=2 sek
	LkB	pidb111,kb0	;N=0
	LkB	pidb123,kb0	;staus SI - praca normalna
	LkB	pidb99,kb225	;status ID - bez samonastrajania
	LkB	pwmb16,kb1	;status SE - regulator aktywny
	LkW	pwmb24,kw1	;jednorazowe konfigurowanie sterownika PWM
	Peu		;TA=1 sek
koniec:	Nop		;status - sterownik PWM aktywny
INT_HC:	Nop		
INT_TIM:	Nop		
STOP:	Br	f255.0	;zeruj semafor pierwszego cyklu programu
	Peu		
	End		

Tabela 4. Przykładowy program użytkownika - jednopętlowa regulacja temperatury