UKŁADY KOMBINACYJNE

Wprowadzenie. Zadanie przykładowe I. Metoda Karnaugha. Schemat sprzętowy. Program w C. Program w ST. Program w LD. Program ST w środowisku TwinCAT PLC Control. Program LD – PLC Control. Niepoprawne pomiary. Zadanie przykładowe II.

WPROWADZENIE

1. Układy przełączające

Podział

Układy przełączające Układy sekwencyjne (z pamięcią)

- Realizacje sprzętowe
 Układy kombinacyjne bramki bez sprzężeń zwrotnych
 Układy sekwencyjne przerzutniki, bramki ze sprzężeniami zwrotnymi
- Opis matematyczny



Układy kombinacyjne – $Y_t = \lambda(X_t)$ – aktualny stan wyjść zależy wyłącznie od aktualnego stanu wejść.

Układy sekwencyjne – $\begin{cases} Q_{t+1} = \delta(Q_t, X_t) \\ Y_t = \mu(Q_t, X_t) \end{cases} - Q_t \text{ stan wewnętrzny} \end{cases}$

Stan wyjść zależy od wejść i stanu wewnętrznego (zależnego od poprzednich wejść \rightarrow pamięć: $Y_t = \mu(\delta(Q_{t-1}, X_{t-1}), X_t)).$

2. Metodologia projektowania układów kombinacyjnych

- Sformułowanie tablicy wejść/wyjść
- Utworzenie funkcji przełączającej metodą Karnaugha
- Realizacja sprzętowa bramki
- Realizacja programowa (języki C, ST, ew. inne)
 - funkcja przełączająca \rightarrow wzór (podstawowa realizacja)
 - schemat bramkowy \rightarrow funkcje AND, OR, NOT
 - tablica wejść/wyjść \rightarrow indeksem do tablicy jest kombinacja we/wy
 - zestaw instrukcji if ... then ... else odpowiadający tablicy we/wy
- Niepoprawne pomiary

3. Struktura programu w prostym sterowniku



Cykl wykonywania programu – np. 10 ms, 0.1 s lub ponowne rozpoczęcie zaraz po poprzednim wykonaniu.

Wizualizacja – LEDy, bargrafy, wyświetlacz LCD.

ZADANIE PRZYKŁADOWE I

G1

G1

-

1. Sterowanie nagrzewaniem



a≤

 $c \leq$

t < a

t < b

 $b \leq t < c$

t

a, b, c: 0 – temperatura poniżej poziomu 1 – temperatura powyżej poziomu lub mu równa

G1, G2 – grzejniki

_

G2 – obydwa grzejniki włączone G2

2. Tablica wejść/wyjść

Zadanie:

c	b	a	G1	G2
0	0	0	1	1
0	0	1	1	0
0	1	1	0	1
1	1	1	0	0

Inaczej – tablica zero–jedynkowa, tablica prawdy *Uwaga*. Stany nie ujęte w tablicy reprezentują awarie

Cwaga. Stany nie ujęte w tablicy reprezentują awarie czujników pomiarowych (typowa reakcja na uszkodzenie → wyłączyć zasilanie).

METODA KARNAUGHA

1. Reguly tworzenia tablic Karnaugha

Chodzi o sformułowanie możliwie prostego wzoru, aby ułatwić realizację sprzętową (w realizacji programowej znaczenie minimalizacji jest mniejsze).

- Wyjścia rozpatruje się oddzielnie.
- Współrzędne pól elementarnych opisane są refleksyjnym kodem Graya.
- Puste pola uzupełnia się znakami nieokreśloności (-).

2. Tablice zadania



3. Reguły upraszczania – łączenie pól elementarnych

- Liczba pól łączonych ze sobą musi być potęgą 2 (1, 2, 4, 8, ...).
- Połączone pola muszą tworzyć prostokąt lub kwadrat, im większy tym lepiej.
- Pola "-" dołącza się do pól "1".
- Pola mogą zachodzić na siebie.
- Górny i dolny wiersz uważa się za sąsiednie, jak również lewą i prawą kolumnę (ze względu na kod Graya).



4. Tworzenie wynikowych wzorów – kolumny

- Wypisać kolumny wejść odpowiadające zakreślonym obszarom.
- Utworzyć iloczyn z tych wejść, które w wypisanych kolumnach mają niezmienione wartości (stale 0 lub 1), przy czym 1 odpowiada sygnałowi prostemu, a 0 – zanegowanemu.
- Wynikowy wzór jest sumą wzorów powstałych z zaznaczonych obszarów.

G	l				G2				
c b	a	c	b	а		c	b	a	_
0 0	0	 0	0	0		0	1	1	
0 0	1	1	0	0		0	1	0	
1 0	0	0	1	0		\downarrow	\downarrow		
1 0	1	1	1	0		- c	• b		– iloczyn
\downarrow				\downarrow					
\overline{b}				a					
G1=	= b			G2	$=\overline{a}$ +	-bc			– suma

Uwagi. Zaznaczenie zbyt małych obszarów w tablicy Karnaugha nie jest błędem, ale wynikowy wzór się rozrasta (nie jest minimalny), bo iloczynów w końcowej sumie przybywa. Zjawisko hazardu obecne w realizacjach sprzętowych nie występuje w realizacjach programowych, ponieważ obliczenia są wykonywane w tym samym cyklu (nie ma różnicy czasów propagacji bramek).

SCHEMAT SPRZĘTOWY



Programowym odpowiednikiem schematów sprzętowych jest graficzny język FBD.

PROGRAM W C

1. Operatory logiczne

char – typ zmiennych logicznych ! negacja, && iloczyn logiczny, || suma logiczna

```
char a,b,c,G1,G2;
...
G1=!b;
G2=!a||b&&!c;
```

2. if ... else – tablica wejść/wyjść

```
if(!c&&!b&&!a) {G1=1; G2=1;}
else
    if(!c&&!b&&a) {G1=1; G2=0;}
    else
        if(!c&&b&&a) {G1=0; G2=1;}
        else
        {G1=0; G2=0;}
```

Wersja powyższa ujmuje także przypadek z niepoprawnymi pomiarami (G1=0, G2=0 – zob. dalej).

PROGRAM W ST

Norma PN–EN 61131–3: 2004(4). *Sterowniki programowalne. Część 3: Języki programowania* definiuje pięć języków:

- ST tekst strukturalny (*Structured Text*)
- IL lista rozkazów (Instruction List)
- LD schemat drabinkowy (Ladder Diagram)
- FBD funkcjonalny schemat blokowy (Function Block Diagram)
- SFC sekwencyjny schemat funkcjonalny (Sequential Function Chart)

Będziemy stosować pakiet TwinCAT sterowników PLC/PAC firmy Beckhoff.

1. Operatory logiczne

2. IF...THEN...ELSE





PROGRAM W LD

1. Elementy



zmienna wejściowa prosta – argument

–//⊢

zmienna zanegowana



2. Operacje logiczne



3. Program – zadanie I



PROGRAM ST W ŚRODOWISKU TWINCAT PLC CONTROL

1. Uruchomienie systemu

• *System > Start* – jeśli system jest już uruchomiony, to *Start* wyszarzone.



Odpowiedź Cancel na pytanie o rejestrację.

PLC Control



Pojawia się okno TwinCAT PLC Control z ostatnio uruchamianym projektem.

🛱 TwinCAT PLC Control - Zbiornik Z1 Z3.pro				
File Edit Project Insert Ex	tras Online Window Help			
POUs •	MAIN (PRG-ST)			

2. Nowy projekt

• File > New



Typ systemu docelowego
 Symulacja i prace domowe - PC or CX (x86)
 Laboratorium - CX(ARM)



• Język programowania – *ST*

New POU		
Name of the new POU:	MAIN	ОК
Type of POU	Language of the POU	Cancel
Program	ΩL	
Function Block	CLD	
C Function	C FBD	
Return Type:	C SFC	
BOOL	ST	
	C CFC	

Pojawia się puste okno edytora programu z *Untitled* jako nazwą projektu. Górna część jest przeznaczona na deklaracje zmiennych i bloków funkcjonalnych, a dolna na właściwy kod.



- 3. Kodowanie, kompilacja
 - Deklaracje i kod



• *Project > Build* lub *Rebuild all*



Informacja o wyniku kompilacji w dolnej części okna

```
Size of used data: 43 of 1048576 bytes (0.00%)
Size of used retain data: 0 of 32768 bytes (0.00%)
0 Error(s), 1 Warning(s).
```

Na warningi nie należy zwracać uwagi.

• Zapis pliku

File > *Save*



Należy najpierw utworzyć katalog na pliki projektu (których może być nawet kilkanaście), tutaj katalog *Sterowanie nagrzewaniem*, i w nim zapisać plik z kodem źródłowym – tutaj *Nagrzewanie.pro* (rozszerzenie dodawane automatycznie).

Zapisywanie	i jako 🔹 💽 🔀
Zapisz w: 🛛 🍋) Sterowanie nagrzewaniem 💽 🗲 🗈 💣 📰 🗸
Nazwa pliku:	Nagrzewanie Zapisz
Zapisz jako	TwinCAT PLC Control Project (*.pro)
typ:	

4. Symulacja

• Online > Choose Run–Time System...



• *Local > RunTime 1 (Port 801)* – lokalny komputer PC

Choose Run-Time System	×
□ - 2 Local (192.168.255.100.1.1) □ - 5 Run-Time 1 (Port 801) □ - 5 CX_077915 (5.7.121.21.1.1) □ - 5 <default> (255.255.255.255.255.255)</default>	OK Cancel

• Ładowanie programu

Online > Login

Online	Window	Help		
Login		F11		
Logout			F12	

Odpowiedź *Tak* na pytanie o załadowanie programu. Początkowe wartości zmiennych.

🎉 MAIN (PRG-ST)	
0001 a = FALSE	~
0002 b = FALSE	
0003 c = FALSE	
0004 G1 = FALSE	
0005 G2 = FALSE	~
0001 G1 := NOT b;	G1 = FALSE
0002 G2 := NOT a OR b AND NO	G2 = FALSE
0003	
0004	
	<

• Online > Run

Online	Window	Help
Logir	1	F11
Logo	ut	F12
Download		
Run		F5
Stop		Shift+F8
Rese	et 🛛	



5. Zapisywanie nowej wartości

• 2 kl. zmienna

Pojawia się proponowana nowa wartość.

🎉 M	AIN (PRG-ST)	
0001	a = FALSE < := TRUE	>
0002	b = <mark>FALSE</mark>	

• Online > Write Values lub Ctrl+F7

Write Values		Ctrl+F7	
🎉 МА	N (PRG-ST)		
0001 0002 0003 0004 0005	a = TRUE b = FALSE c = FALSE G1 = TRUE G2 = FALSE		
0001 G 0002 G 0003 0004	1 := NOT b; 2 := NOT a OR b ())))	G1 = TRU G2 = AL	

6. Zakończenie

• Online > Logout

	ricip
Login	F11
Logout	F12
Download	

PROGRAM LD – PLC CONTROL

- 1. Nowy projekt, typ POU, język LD
 - *File > New*
 - *PC or CX* (*x*86)
 - Language LD

New POU		
Name of the new POU:	MAIN	ОК
Type of POU	Language of the POU	Cancel
Program	СЦ	
Function Block	● LD	
C Function	C FBD	
Return Type:	SFC	
BOOL	C ST	
	C CFC	

Okno edytora LD – deklaracje w górnej części (j.p.)

 dolna część przeznaczona na schemat; widoczny jeden szczebel drabinki.

🗱 MAIN	I (PRG-LD)	<
0001 PF 0002 VA 0003 EN	IOGRAM MAIN R ID_VAR	
<		 >
0001		
< -		>

2. Tworzenie schematu LD

• Wybór miejsca, menu kontekstowe (prawy klawisz myszy) > wybór elementu schematu, np. *Contact (negated)*.



Pojawia się styk zanegowany, bez nazwy zmiennej – pytajniki ???

• Nazwa zmiennej

Zaznaczyć ??? > przycisk F2 > okno *Input assistant* > wybór zmiennej, np. b.





• Pełny schemat

黲 MAIN (PRG-LD)	
0001 PROGRAM MAIN	^
0004 a, b, c, G1, G2; BOOL; 0005 END_VAR	24
	>
0001.	
ь	G1
0002	
a	G2
	()
b c	
	>

- Kompilacja *Project > Build*
- Zapis pliku *File > Save* Najpierw utworzono katalog *Nagrzewanie LD* na pliki projektu.

Zapisywanie	jako				? 🔀
Zapisz w: 📔) Nagrzewanie LD	•	• 🔁	ci *	
Nazwa pliku:	Nagrzewanie LD				Zapisz
Zapisz jako typ:	TwinCAT PLC Control Project (*.pro)		•		Anuluj

3. Symulacja

- Online > Choose Run–Time System > Local > RunTime 1 (Port 801)
- Ładowanie *Online > Login* (początkowe wartości zmiennych)
- Uruchomienie *Online* > *Run*



4. Nowa wartość zmiennej

- 2 kl. styk, np. *a*
- Połowa styku zmienia kolor; proponowana nowa wartość widoczna także w górnej części okna (deklaracje).

😹 MAI	N (PRG-LD)
0001	a = FALSE < := TRUE>
0002	b = <u>FALSE</u>
0003	c = FALSE
0004	G1 = TRUE
0005	G2 = TRUE
0006	
0001	
	b
0002	
	a
	b BOOL C
0003	L
00001	
رتساريخا	

• Ctrl + F7 lub *Online* > *Write Values*



NIEPOPRAWNE POMIARY

1. Tablica poprawności pomiarów

c	b	a	Р
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

- P = 1 pomiary poprawne
- P = 0 pomiary niepoprawne

2. Tablica Karnaugha



Uwaga. Teraz w tablicy Karnaugha dotyczącej poprawności pomiarów nie ma pól "-" (nie ma nieokreśloności).

3. Wzór



meokresionoser).

4. Wymaganie technologiczne

W przypadku niepoprawnych pomiarów obydwie grzałki należy wyłączyć - G1=G2=0.

5. Programowanie

```
C
char P; // poprawność pomiarów
...
P=a&&b||!b&&!c;
if(P)
        (G1=!b; G2=!a||b&&!c;)
else
        (G1=G2=0;)
```

<u>ST</u>

<u>LD</u>





 $P = a \cdot b$ $P = P + \overline{b} \cdot \overline{c}$

ZADANIE PRZYKŁADOWE II

1. Sterowanie nagrzewaniem

Włączanie grzejników:

- równolegle G1, G2, gdy $t \le t_a$
- tylko G1, $gdy t_a < t \le t_b$
- tylko G2, $gdy t_b < t \le t_c$
- szeregowo G1, G2, gdy $t_c < t \le t_d$
- wyłączone G1, G2, gdy $t_d \le t$



Połączenie powyższe pozwala uzyskać cztery stopnie mocy grzejnej przy dwóch grzejnikach.

2. Tablica wejść/wyjść

d	c	b	а	\mathbf{w}_1	W_2	W ₃
0	0	0	0	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	1	0
0	1	1	1	0	1	1
1	1	1	1	0	0	0

3. Tablice Karnaugha i kolumny



4. Programowanie – stan normalny

<u>C</u>

char d,w1,w2,w3; - dodatkowe deklaracje
...
w1=!b; w2=!a||b&&!d; w3=c&&!d;



RG-LD)	
a, b.,c, d, w1, w2, w3: BOOL;	
_VAR	
b	w1
/	()
8	w2
	<u> </u>
1/1	0
b d	
c u	W-0
	1
	a - - b - <t< td=""></t<>

5. Niepoprawne pomiary





Zmienna P



 $P = \overline{dcb} + cba + \overline{dca}$

Programowanie

<u>C</u>

```
char P;
...
P=!d&&!c&&!b || c&&b&&a || !d&&!c&&a;
if(P)
        {w1=!b; w2=!a||b&&!d; w3=c&&!d;}
else
        w1=w2=w3=0;
```

<u>ST</u>

🎉 MAIN (PRG-ST)	
0001 PROGRAM MAIN 0002 VAR	
0004 P:BOOL;	
	>
0001 P:= NOT d AND NOT c AND NOT b OR	~
0002 cANDbANDa OR	
0003 NOT bAND NOT cAND a;	
0004	
0005 IF P THEN	
0006 w1 := NOT b;	
0007 w2 := NOT a OR b AND NOT d;	
0008 w3 := c AND NOT d;	
0009ELSE	
0010 w1 := w2 := w3 := FALSE;	
0011END_IF	
0012	~
	> .;;