## WOJSKOWA AKADEMIA TECHNICZNA

# *LABORATORIUM*

# *WPROWADZENIE DO AUTOMATYKI*

Grupa szkoleniowa

Stopień, imię i nazwisko prowadzącego

Stopień, imię i nazwisko słuchacza

***I7X3S1***

***mgr inż. Małgorzata Rudnicka - Schmidt***

***Grzegorz Pol***

Data wykonania ćwiczenia

***27.11.2008 r.***

***SPRAWOZDANIE***

***Z***

***PRACY LABORATORYJNEJ***

***NR 2***

|  |  |
| --- | --- |
| **Temat:** | Programowanie sterowników logicznych PLC |

* 1. **Zadania:**

Podczas zajęć laboratoryjnych należało wykonać zadania o poniższej treści:

1. Zapisać swoje inicjały w formie 4 ostatnich (najmniej znaczących) bitów kodu ASCII dla każdego znaku, a następnie zaprojektować schemat drabinkowy układu, w którym po naciśnięciu pierwszego przycisku spowoduje „zapalenie się” wyjść, które będą odpowiadały ostatnim 4 bitom pierwszej litery imienia, a po naciśnięciu drugiego przycisku – na wyjściu pojawią się sygnały, które będą symbolizować ostatnie 4 bity pierwszej litery nazwiska w kodzie ASCII
2. Drugie zadanie polegało również na ostatnich 4 bitach pierwszej litery imienia i nazwiska zapisanej w kodzie ASCII oraz zaprojektowaniu układu drabinkowego, który po uaktywnieniu wejść w sekwencji odpowiadającej wartością 4 ostatnich bitów pierwszej litery imienia powinien „zapalić” wyjście nr 1 i analogicznie po uaktywnieniu wejść jako sekwencję 4 ostatnich bitów pierwszej litery nazwiska „zapali” wyjście numer 2.
3. Trzecie zadanie natomiast polegało na zaprojektowaniu schematu układu w środowisku FBD sterującą pracą silnika. Układ posiada 6 wejść i 2 wyjścia. Oznaczenia:

- włączanie silnika (zapalanie )

- wyłączanie silnika (gaszenie )

, lub włączanie silnika i włączanie alarmu (jeżeli jest włączone to zostaje wyłączone)

- wyłączanie alarmu (gaszenie )

* 1. **Aplikacja**

Do realizacji powyższych zadań używałem aplikacji LOGO! Soft Comfort v4.0 Pliki programów wysłałem prowadzącemu drogą mailową.

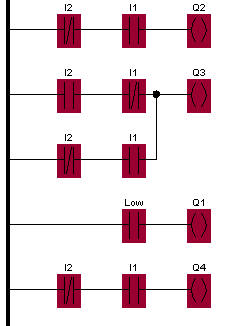
* 1. **Kodowanie inicjałów**

W moim przypadku operowałem połowicznie ze swoich znaków, gdyż 4 najmniej znaczące bity z litery **P** to **0000**. Literkę **P** zastąpiłem literą dźwiękowo ją naśladującą - **B.**

1. Kod ASCII dla litery **G** wynosi odpowiednio: (71)10  (47)16 (0100 0111)2, a więc 4 najmniej znaczące bity w tym przypadku to sekwencja: **0111**
2. Kod ASCII dla litery **B** wynosi odpowiednio: (66)10  (42)16 (0100 0010)2, a więc 4 najmniej znaczące bity w tym przypadku to sekwencja: **0010**
   1. **Tabela prawdy dla zadania a)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I1 | I2 | Q1 | Q2 | Q3 | Q4 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |

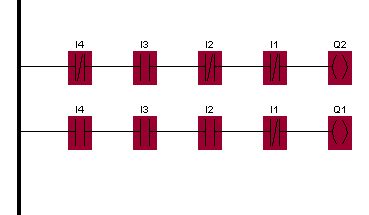
* 1. **Zapis funkcji dla zadania a)**
  2. **Schemat LAD dla zadania a)**

****

* 1. **Tabela prawdy dla zadania b)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I1 | I2 | I3 | I4 | Q1 | Q2 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| Pozostałe 14 kombinacji | | | | 0 | 0 |

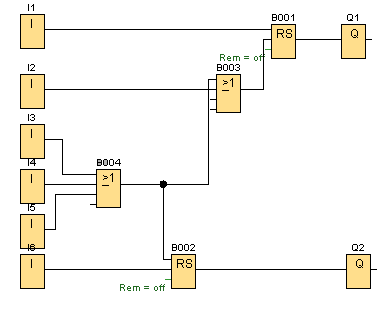
* 1. **Zapis funkcji dla zadania b)**
  2. **Schemat LAD dla zadania b)**

****

* 1. **Tabela prawdy dla zadania c)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Q1 | Q2 | I1 | I2 | I3 | I4 | I5 | I6 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

* 1. **Zapis funkcji dla zadania c)**
  2. **Schemat blokowy dla zadania c)**

****

W tym schemacie po raz pierwszy użyliśmy blok Latching Relay, który to odgrywa tutaj rolę prostej pamięci. Wynik na wyjściu oprócz wartości na wejściu zależny jest od poprzedniego stanu wyjścia zapisanego przez RS. Tabela prawdy dla bloku wygląda następująco:

|  |  |  |  |
| --- | --- | --- | --- |
| R | S | Q | Wynik |
| 0 | 0 | X | Brak zmian |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |
| 1 | 1 | 0 | Reset |

**7. Wnioski**

Podsumowując: wszystkie zadania z pkt. 1.1 zostały przeze mnie pomyślnie zrealizowane. Aplikacja LOGO!Soft w wersji 4.0 jest o wiele bardziej rozbudowana od swojej poprzedniczki (v.2.0). Umożliwia nam między innymi korzystanie z nowych funkcji programu: np. bloku RS, który to daje nam możliwości na stworzenie jeszcze bardziej dostosowanego do naszych potrzeb schematu przekaźników. Ponadto zastosowanie tutaj metody *Drag & Drop* stanowczo ułatwia użytkownikom korzystanie z nowszej wersji aplikacji Siemensa.