



Politechnika Wrocławska

Wydział Informatyki i  
Telekomunikacji



## Urządzenia wejścia i wyjścia

### *Sprawozdanie*

Przekaźniki, styczniaki, elementy sterownicze, sterowniki mocy, falowniki  
i siłowniki

Bartosz Dróżdż 280041

Igor Janiak 280125

Mikołaj Nowak 280082

Wydział informatyki i telekomunikacji,

# Informatyczne systemy automatyki

31 stycznia 2024

## Spis treści

1 Wstęp	1
2 Budowa układu	1
3 Sterownik Logo	3
4 Podsumowanie	5

## Spis ilustracji

<u>Rysunek 1 Schemat układu</u> .....	2
<u>Rysunek 2 Podłączony układ</u> .....	3
<u>Rysunek 3 układ ze sterownikiem LOGO</u> .....	4
<u>Rysunek 4: Schemat blokowy</u> .....	5

## 1 Wstęp

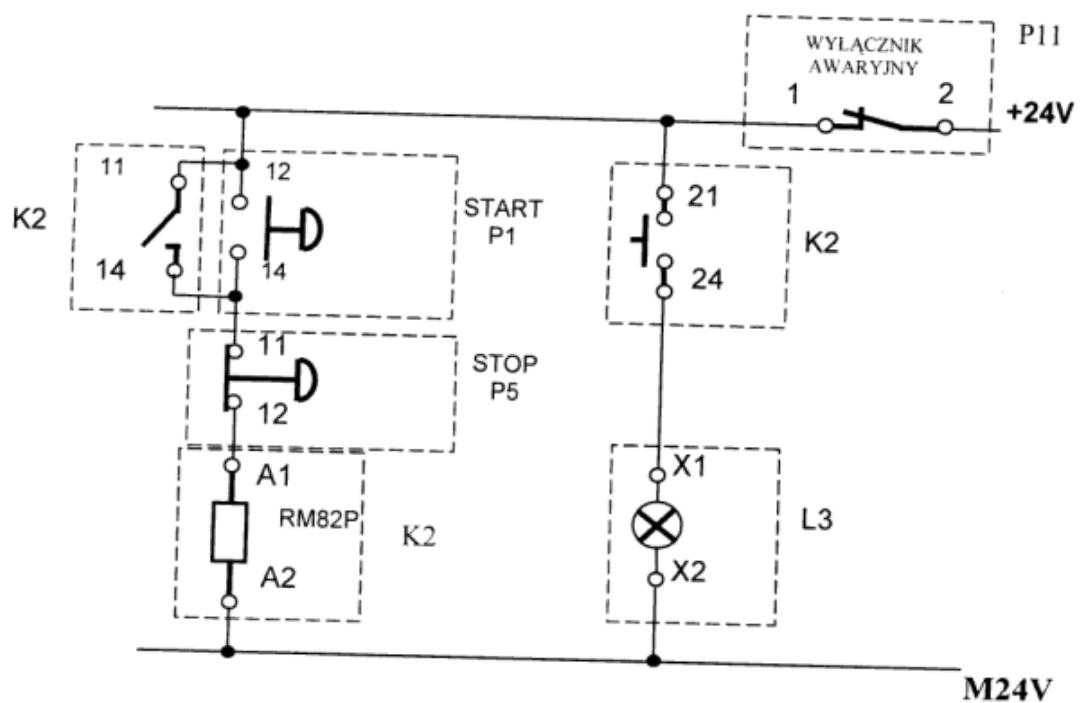
Celem ćwiczenia jest stworzenie układu elektrycznego zapalającego i gaszącego lampkę za pomocą przycisków z mechanizmem bezpieczeństwa, a następnie zbudowanie układu zachowującego się w ten sam sposób wykorzystując sterownik LOGO. Układ spełnia następujące wymagania:

- Przycisk START uruchamiający układ
- Przycisk STOP zatrzymujący układ
- Przycisk bezpieczeństwa odcinający zasilanie od układu

## 2 Budowa układu

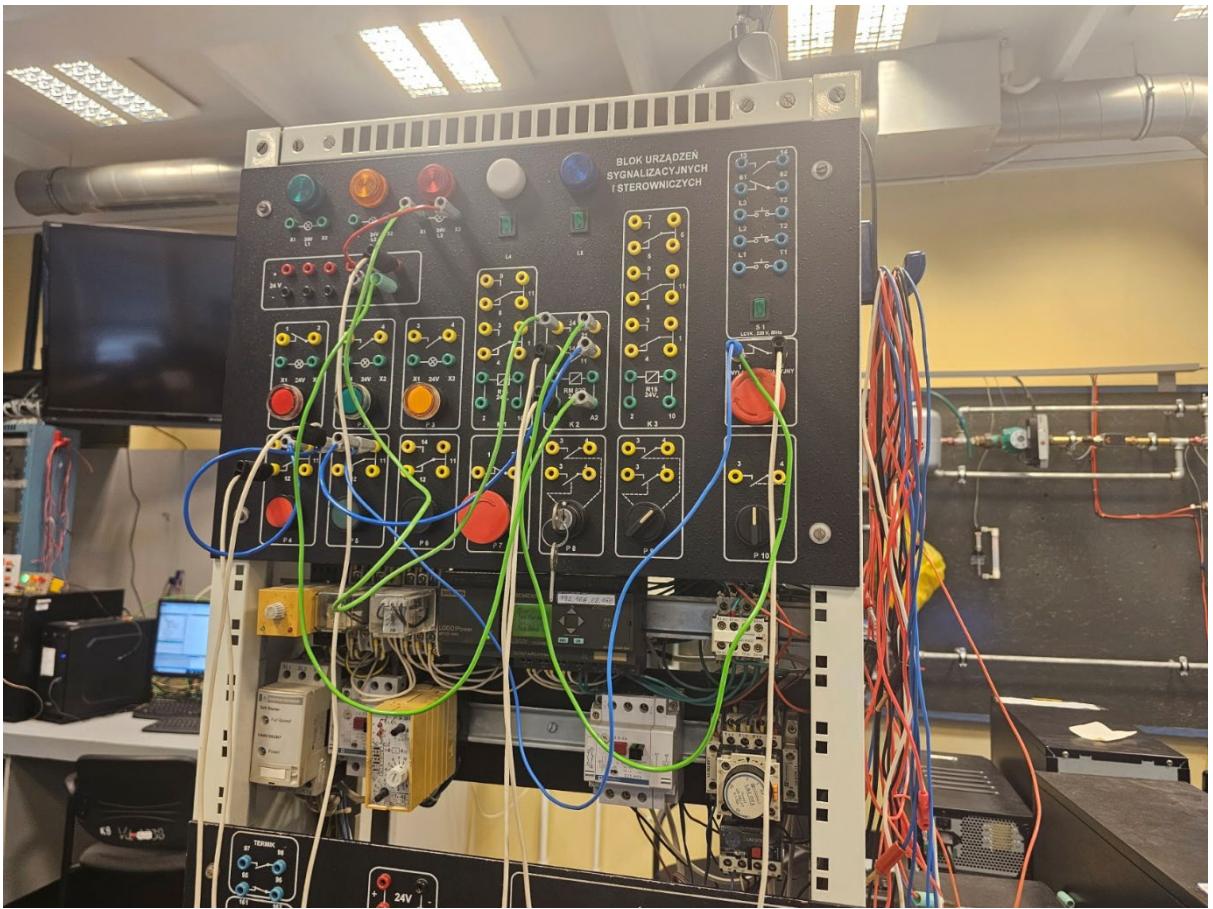
Pierwszą częścią ćwiczenia jest podłączenie układu elektrycznego bez wykorzystania sterownika.

Poniżej zaprezentowany jest schemat układu budowanego w tym etapie ćwiczenia.



Rysunek 1 Schemat układu

Podłączony układ wygląda następująco



Rysunek 2 Podłączony układ

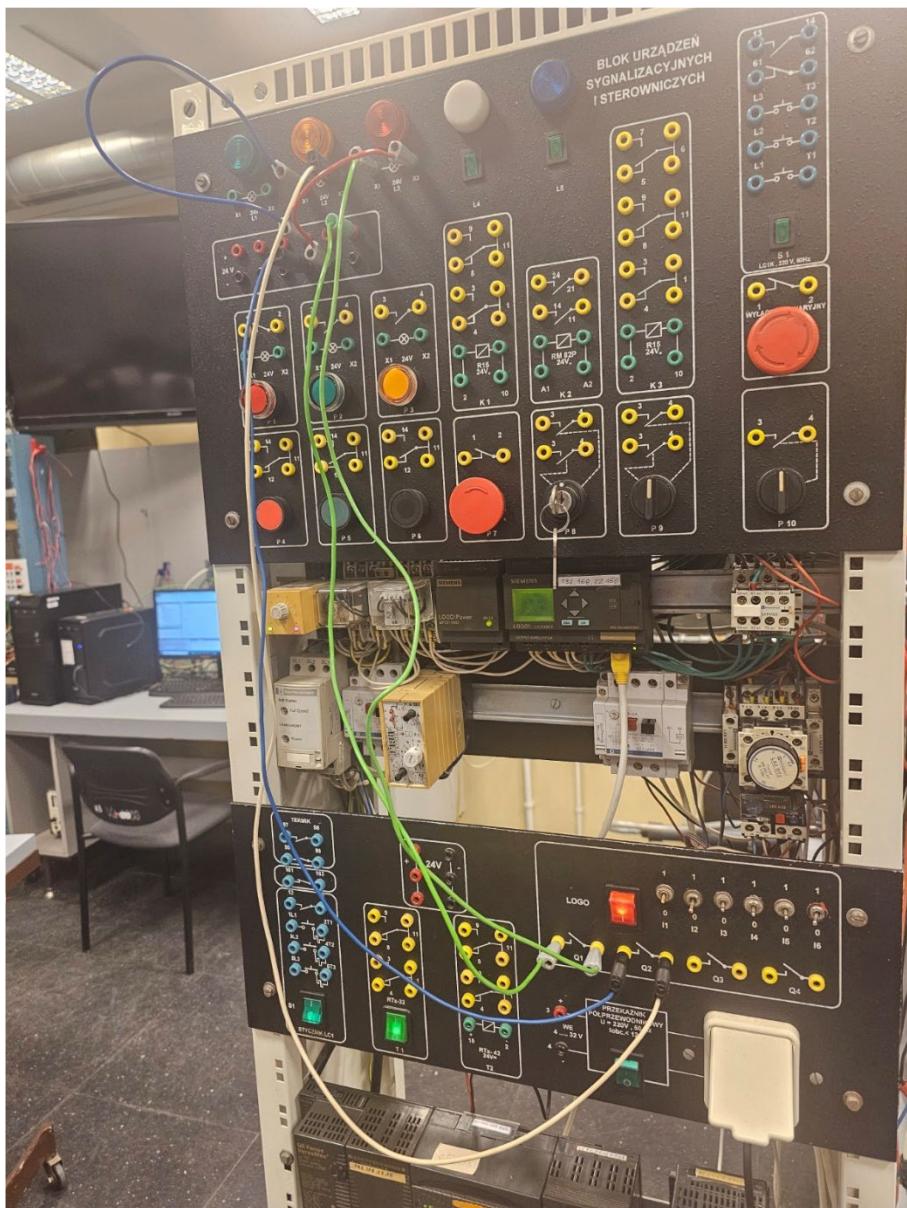
Przycisk **START** posiada domyślny stan niski, natomiast przycisk **STOP** oraz przycisk **awaryjny** (tzw. "grzybek") mają stan wysoki. Po naciśnięciu przycisku awaryjnego jego stan zmienia się na niski i pozostaje taki do momentu ręcznego resetu poprzez przekręcenie.

Zastosowanie tych elementów umożliwia użytkownikowi włączenie diody LED przez naciśnięcie przycisku **START**. Dioda pozostaje w stanie świecenia do czasu naciśnięcia przycisku **STOP** lub przerwania zasilania przez aktywację przycisku awaryjnego. Po jego zadziałaniu układ nie zostanie ponownie uruchomiony, dopóki nie zostanie zresetowany.

Zaletą użycia przycisków zamiast przełączników typu **on-off** jest konieczność każdorazowego, świadomego uruchomienia układu po przerwie w zasilaniu lub użyciu przycisku bezpieczeństwa. Rozwiążanie to zwiększa bezpieczeństwo i zmniejsza ryzyko przypadkowego włączenia obwodu.

### 3 Sterownik Logo

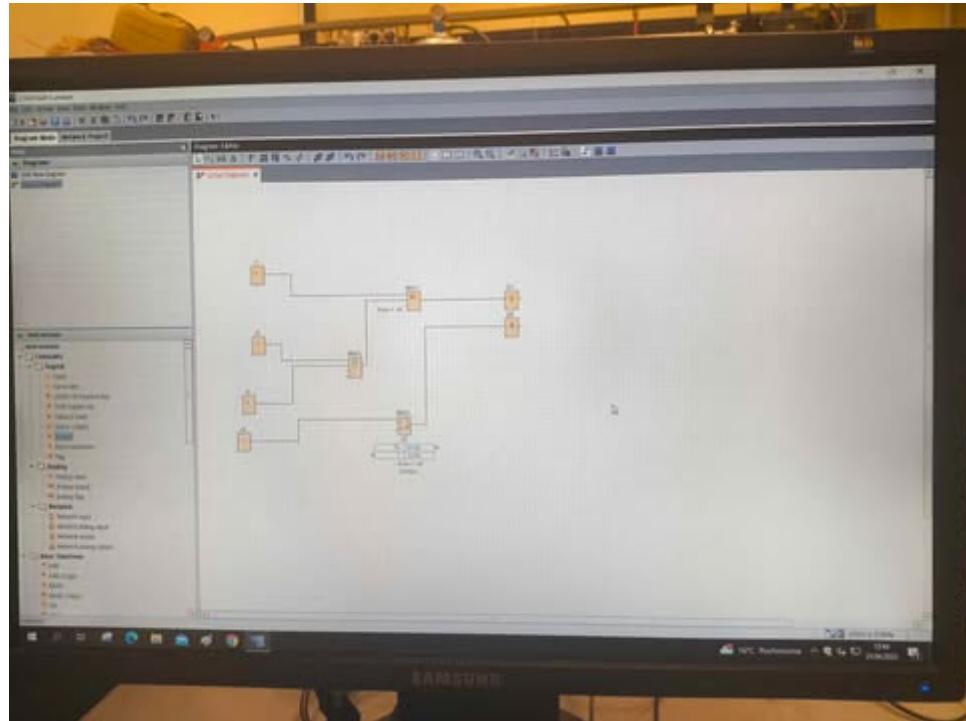
W drugiej części naszym zadaniem było zaprogramowanie sterownika do obsługi stworzonego wcześniej układu elektrycznego.



Rysunek 3 układ ze sterownikiem LOGO

W celu odwzorowania działania układu przy wykorzystaniu sterownika, zamiast fizycznych przycisków otwierających i zamkijających obwód zastosowano przełączniki skonfigurowane w programie sterującym. Funkcjonalność układu pozostała niezmieniona — sterownik przejął rolę realizacji logiki działania, sterując lampką w zależności od stanów wejściowych przełączników. Programowanie odbywało się w języku blokowym, z wykorzystaniem schematu połączeń odzwierciedlającego fizyczną strukturę układu.

Zaprojektowany schemat blokowy odzwierciedla fizyczne połączenia w układzie. I1, I2, I3, I4 to odpowiednio przyciski start, stop, awaryjny oraz przycisk timera. Program działał i poprawnie sterował wszystkimi akcjami.



Rysunek 4: Schemat blokowy

## 4 Podsumowanie

- Ćwiczenie pozwoliło na praktyczne zapoznanie się z budową prostego układu elektrycznego sterowanego za pomocą przycisków typu START, STOP oraz przycisku awaryjnego (grzybka).
- Zastosowanie przycisków zamiast przełączników on-off znaczco zwiększa bezpieczeństwo obsługi układu poprzez wymuszenie świadomego jego uruchomienia po każdorazowym odcięciu zasilania.
- Druga część zadania umożliwiła poznanie sposobu programowania sterownika LOGO i odwzorowania logiki działania fizycznego układu w formie schematu blokowego.
- Zastosowanie sterownika umożliwia większą elastyczność w projektowaniu i modyfikowaniu logiki działania układów automatyki, bez konieczności zmian w fizycznym okablowaniu.
- Całość projektu przebiegła zgodnie z założeniami — układ działał poprawnie zarówno w wersji konwencjonalnej, jak i sterowanej przez sterownik programowalny.