**Treść zadania projektowego:**

Zaprojektować układ sterowania silnikiem windy. Winda porusza się między trzema następującymi kondygnacjami: 0, 2, 3. Zadany jest stan początkowy (zachowanie windy w momencie uruchomienia programu na sterowniku PLC i na mikrokontrolerze).

Stan początkowy – winda ustawia się na 2 kondygnacji.

**Założenia ustalone przed prowadzącą laboratoria:**

I1 P1 - przycisk wysyłania windy na 0 kondygnację

I2 P2 - przycisk wysyłania windy na 1 kondygnację

I3 P3 - przycisk wysyłania windy na 2 kondygnację

I4 P4 - przycisk wysyłania windy na 3 kondygnację

I5 P5 - czujnik (sensor) obecności windy na 0 kondygnacji

I6 P6 - czujnik (sensor) obecności windy na 1 kondygnacji

I7 P7 - czujnik (sensor) obecności windy na 2 kondygnacji

I8 P8 - czujnik (sensor) obecności windy na 3 kondygnacji

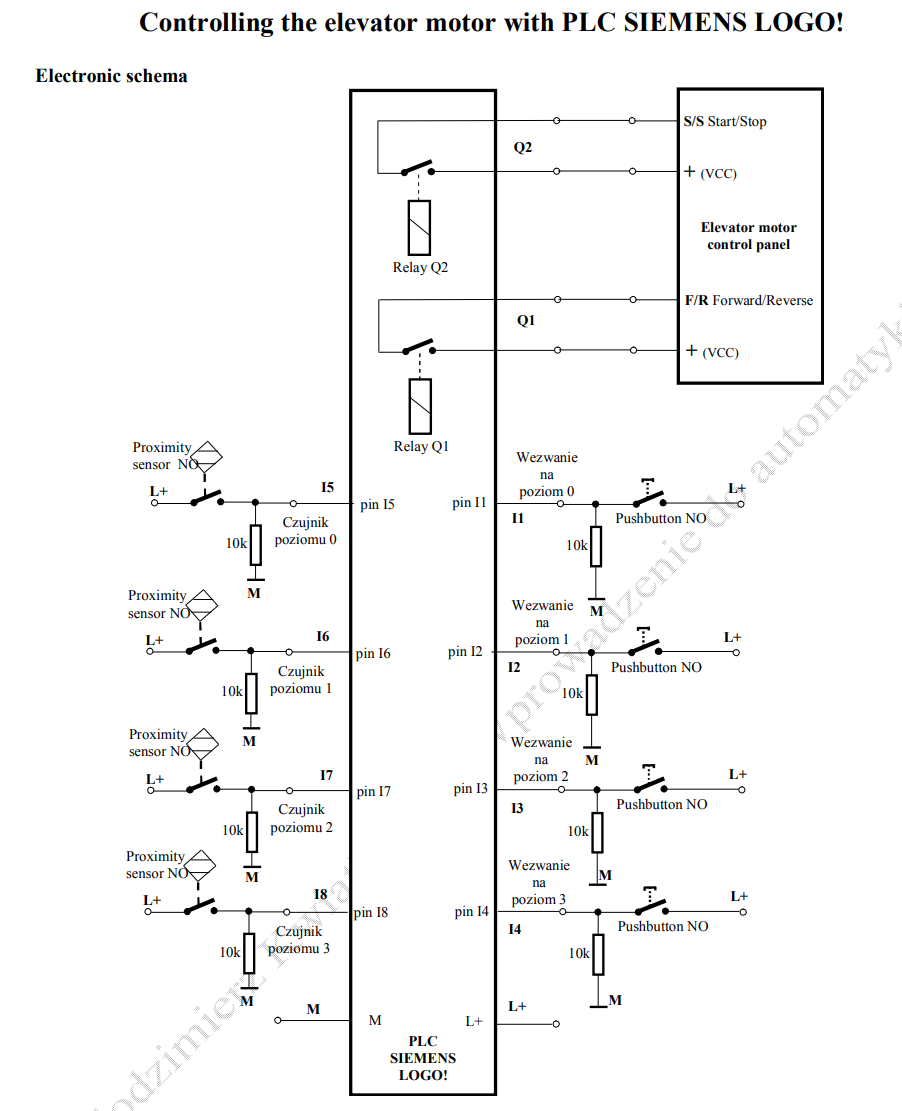
Sterowanie silnikiem windy

F/R, S/S – wejścia sterujące silnikiem połączone z wyjściami PLC

F/R z wyjściem Q1

S/S z wyjściem Q2

|  |  |  |
| --- | --- | --- |
| F/R  (bit kierunku) | S/S  (bit stopu) | reakcja |
| 0 | 1 | Stop |
| 0 | 0 | ↑ (w górę) |
| 1 | 0 | ↓ (w dół) |
| 1 | 1 | Kombinacja niewykorzystana (niedozwolona) |



**Sterowanie silnikiem windy (ARDUINO)**

Pin 5 - F/R (bit kierunku) - Q1

Pin 4 - S/S (bit stopu) - Q2

Uwaga: Wiszące wejścia S/S i F/R powodują ruch windy do góry (wtedy sygnały S/S i F/R mają wartości zerowe).

The #define directives in Arduino sketch

#define Button1Pin A0 //nazwa pinu dla "Przycisk żądanie 0-go poziomu"

#define Button2Pin A1 //nazwa pinu dla "Przycisk żądanie 1-go poziomu"

#define Button3Pin A2 //nazwa pinu dla "Przycisk żądanie 2-go poziomu"

#define Button4Pin A3 //nazwa pinu dla "Przycisk żądanie 3-go poziomu"

#define Sensor1Pin A4 //nazwa pinu dla "Sensor obecności kabiny na 0-wym poziomie"

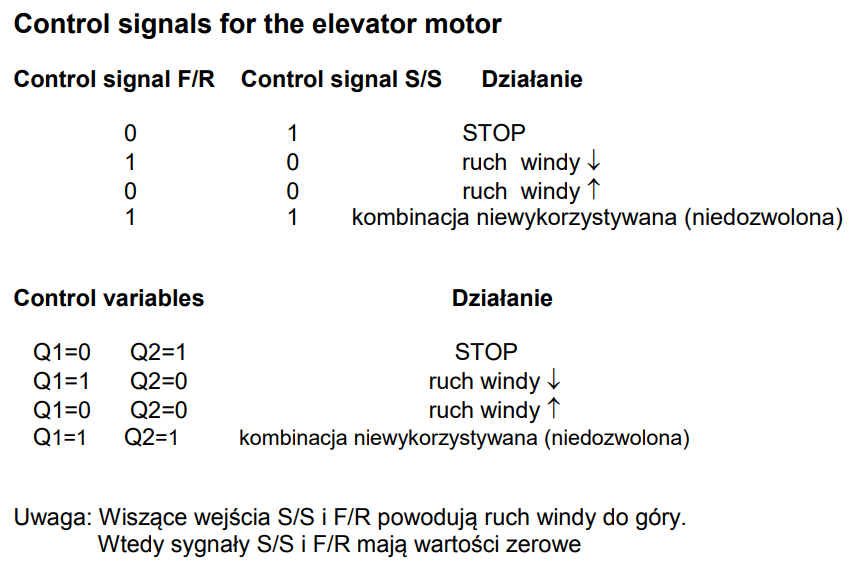
#define Sensor2Pin A5 //nazwa pinu dla "Sensor obecności kabiny na 1-ym poziomie"

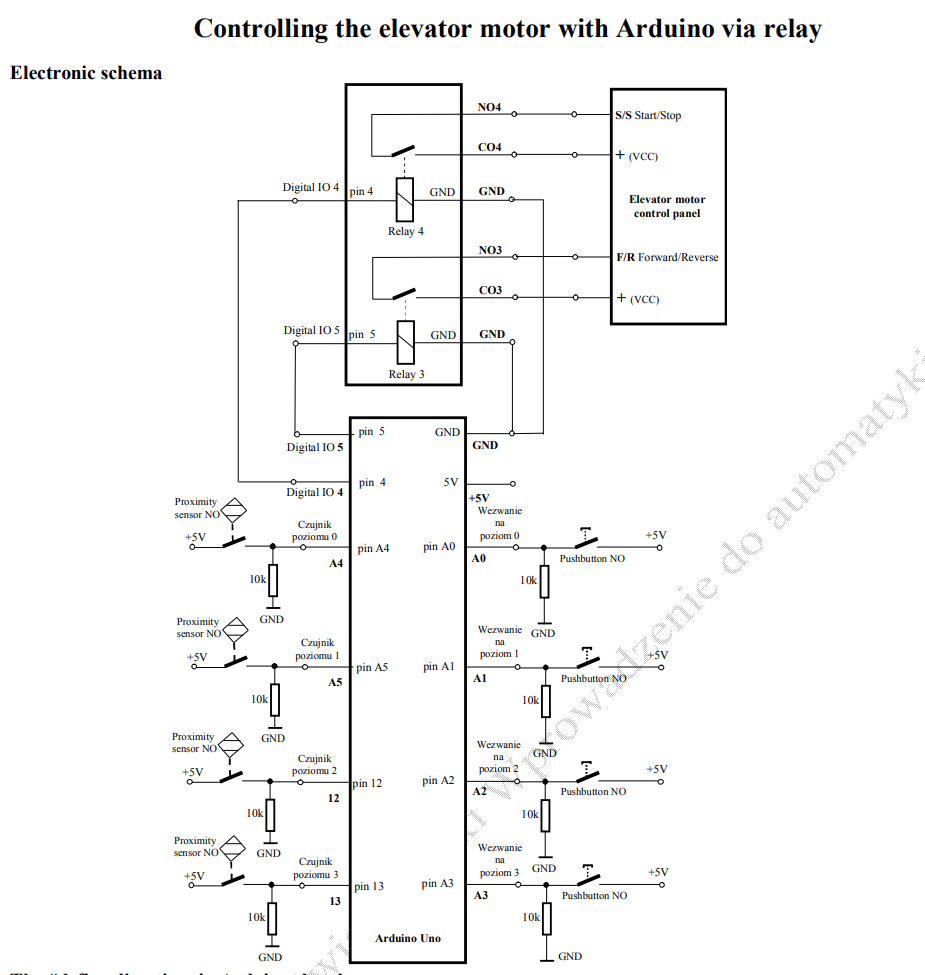
#define Sensor3Pin 12 //nazwa pinu dla "Sensor obecności kabiny na 2-im poziomie"

#define Sensor4Pin 13 //nazwa pinu dla "Sensor obecności kabiny na 3-im poziomie"

#define OutputQ1Pin 5 //nazwa pinu dla Forward/Reverse Signal (bit kierunku)

#define OutputQ2Pin 4 //nazwa pinu dla Start/Stop Signal (bit stopu)





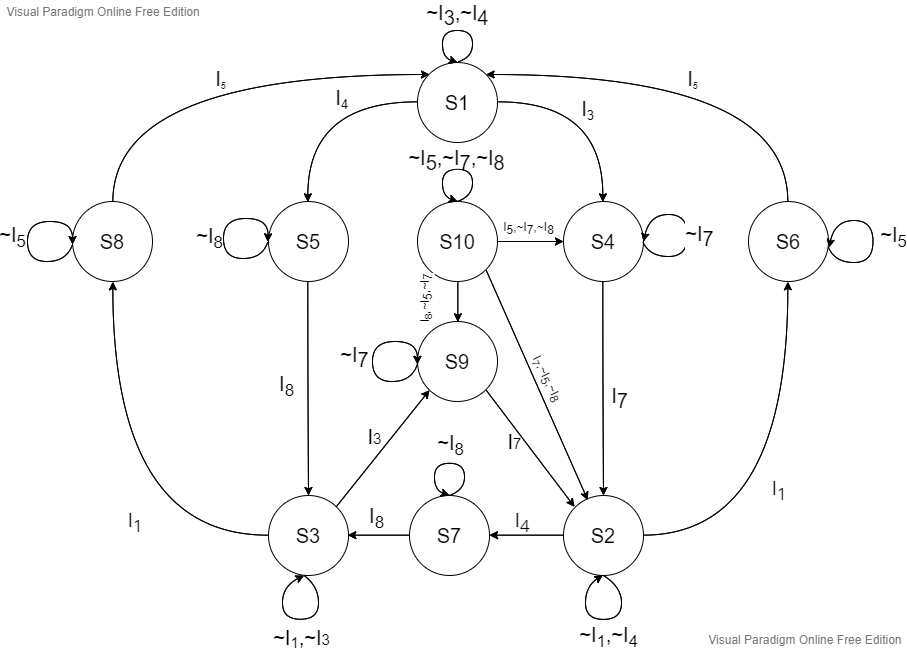
1. **Założenia funkcjonalne**
2. **Definicja stanów maszyny stanowej:**

*, oczekuje na ustalenie obecności windy na danej kondygnacji*

1. **Definicja stanów maszyny stanowej:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kod  Stan | M1 | M2 | M3 | M4 |
| S1 | 0 | 0 | 0 | 1 |
| S2 | 0 | 0 | 1 | 0 |
| S3 | 0 | 0 | 1 | 1 |
| S4 | 0 | 1 | 0 | 0 |
| S5 | 0 | 1 | 0 | 1 |
| S6 | 0 | 1 | 1 | 1 |
| S7 | 1 | 0 | 0 | 0 |
| S8 | 1 | 0 | 0 | 1 |
| S9 | 1 | 0 | 1 | 0 |
| S10 | 0 | 0 | 0 | 0 |

1. **Diagram przejść maszyny stanowej**



1. **Wyznaczenie funkcji przejścia**
2. **Tabela przejść stanów**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I1 | I2 | I3 | I4 | I5 | I6 | I7 | I8 | M1 | M2 | M3 | M4 |  |  |  |  |
| \* | \* | 0 | 0 | \* | \* | \* | \* | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| \* | \* | 1 | \* | \* | \* | \* | \* | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| \* | \* | \* | 1 | \* | \* | \* | \* | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | \* | \* | 0 | \* | \* | \* | \* | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | \* | \* | \* | \* | \* | \* | \* | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| \* | \* | \* | 1 | \* | \* | \* | \* | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | \* | 0 | \* | \* | \* | \* | \* | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | \* | \* | \* | \* | \* | \* | \* | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| \* | \* | 1 | \* | \* | \* | \* | \* | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| \* | \* | \* | \* | \* | \* | 0 | \* | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| \* | \* | \* | \* | \* | \* | 1 | \* | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| \* | \* | \* | \* | \* | \* | \* | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| \* | \* | \* | \* | \* | \* | \* | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| \* | \* | \* | \* | 0 | \* | \* | \* | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| \* | \* | \* | \* | 1 | \* | \* | \* | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| \* | \* | \* | \* | \* | \* | \* | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| \* | \* | \* | \* | \* | \* | \* | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| \* | \* | \* | \* | 0 | \* | \* | \* | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| \* | \* | \* | \* | 1 | \* | \* | \* | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| \* | \* | \* | \* | \* | \* | 0 | \* | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| \* | \* | \* | \* | \* | \* | 1 | \* | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| Definicja stanu początkowego (winda ustawia się na 2 kondygnacji): | | | | | | | | | | | | | | | |
| \* | \* | \* | \* | 1 | \* | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| \* | \* | \* | \* | 0 | \* | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| \* | \* | \* | \* | 0 | \* | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

**M1, M2, M3, M4 –** wyróżniony bit stanu układu, w którym obecnie znajduje się układ

– wyróżniony bit stanu układu, w którym układ będzie się znajdować

w następnej chwili czasowej

1. **Wzory (wyrażenia algebraiczne) do obliczania wartości funkcji dla zadanych wartości argumentów**

**Przed minimalizacją:**

**Minimalizacja dla :**

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

Ostatecznie:

**Minimalizacja dla :**

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

Ostatecznie:

**Minimalizacja dla :**

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

cd. usuwanie nadmiarowości:

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

cd. usuwanie nadmiarowości:

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

cd. pochłanianie:

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

Ostatecznie:

**Minimalizacja dla :**

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

cd. usuwanie nadmiarowości:

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

cd. pochłanianie:

cd. usuwanie nadmiarowości:

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

cd. pochłanianie:

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

cd. pochłanianie:

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy dwóch powyższych wyrażeń:

cd. pochłanianie:

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

cd. pochłanianie:

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

cd. usuwanie nadmiarowości:

Ostatecznie:

**Po minimalizacji:**

1. **Wyznaczenie funkcji wyjścia**
2. **Tabela funkcji wyjścia**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Kod  Stan | M1 | M2 | M3 | M4 | Q1  (F/R) | Q2  (S/S) |
| S1 | 0 | 0 | 0 | 1 | 0 | 1 |
| S2 | 0 | 0 | 1 | 0 | 0 | 1 |
| S3 | 0 | 0 | 1 | 1 | 0 | 1 |
| S4 | 0 | 1 | 0 | 0 | 0 | 0 |
| S5 | 0 | 1 | 0 | 1 | 0 | 0 |
| S6 | 0 | 1 | 1 | 1 | 1 | 0 |
| S7 | 1 | 0 | 0 | 0 | 0 | 0 |
| S8 | 1 | 0 | 0 | 1 | 1 | 0 |
| S9 | 1 | 0 | 1 | 0 | 1 | 0 |
| S10 | 0 | 0 | 0 | 0 | 0 | 0 |

1. **Wzory dla funkcji wyjścia**

**Przed minimalizacją:**

**Minimalizacja dla**

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

Ostatecznie:

**Minimalizacja dla**

Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

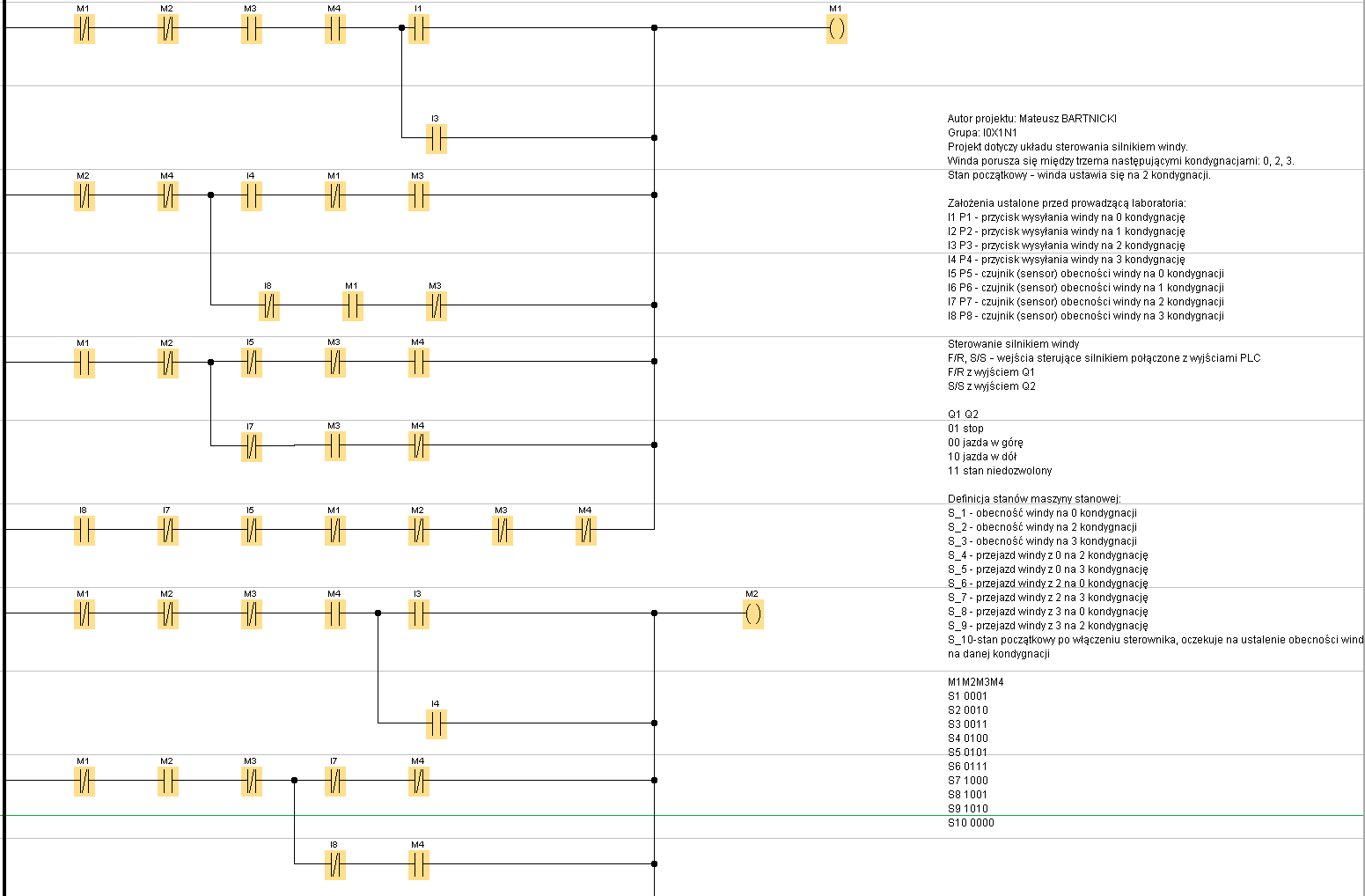
cd. pochłanianie:

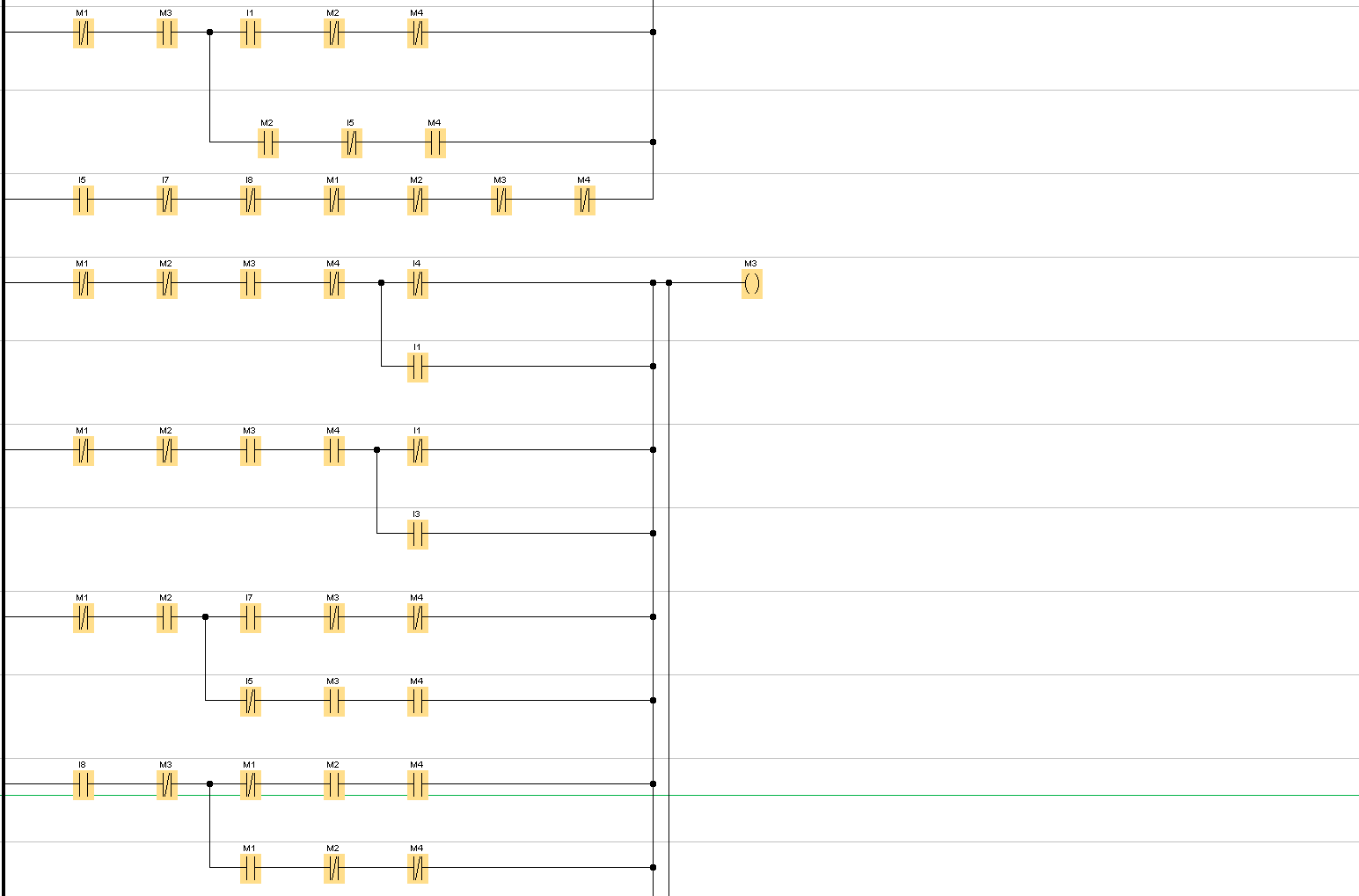
Wykorzystanie rozdzielności operacji iloczynu względem operacji sumy:

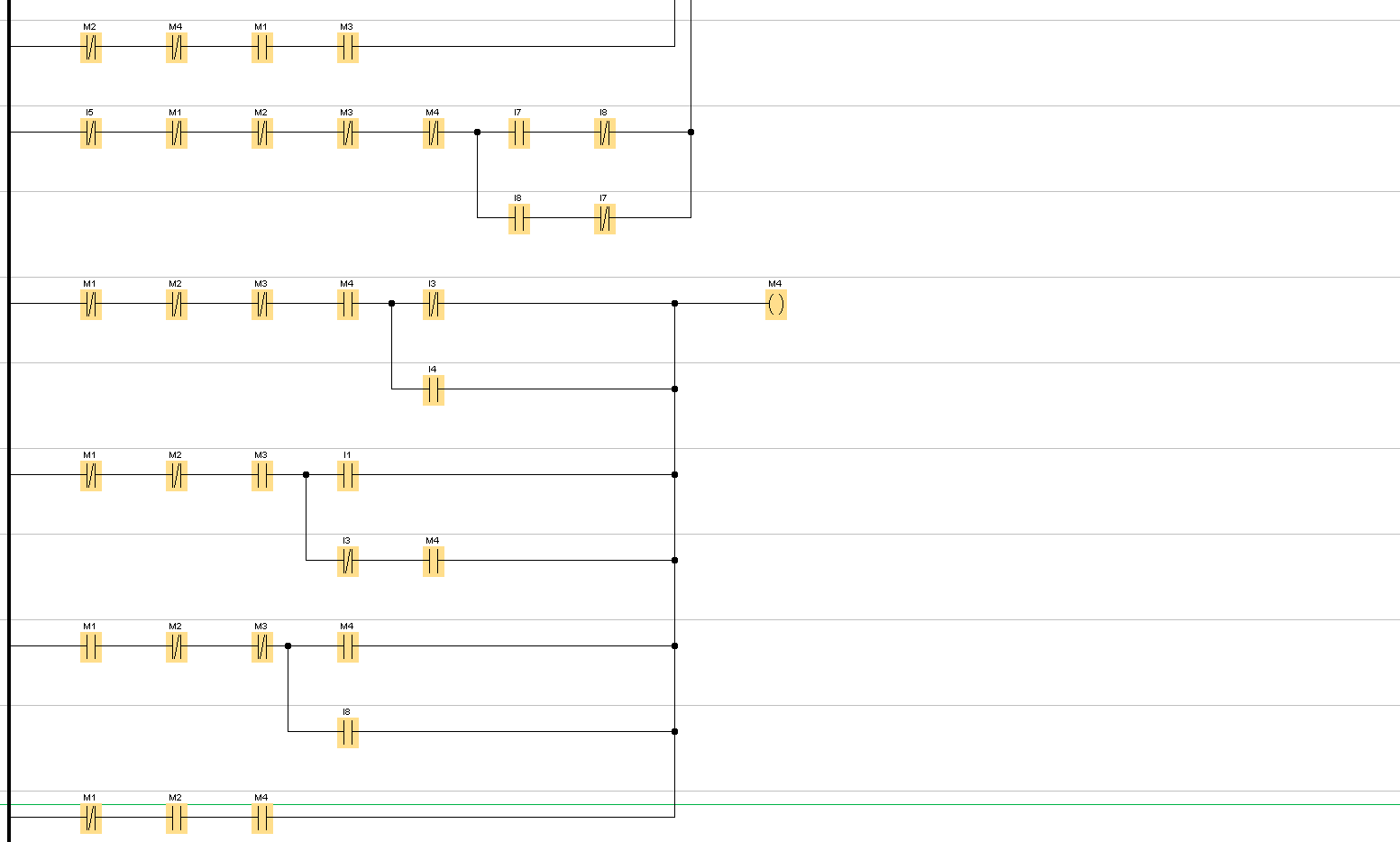
cd. usuwanie nadmiarowości:

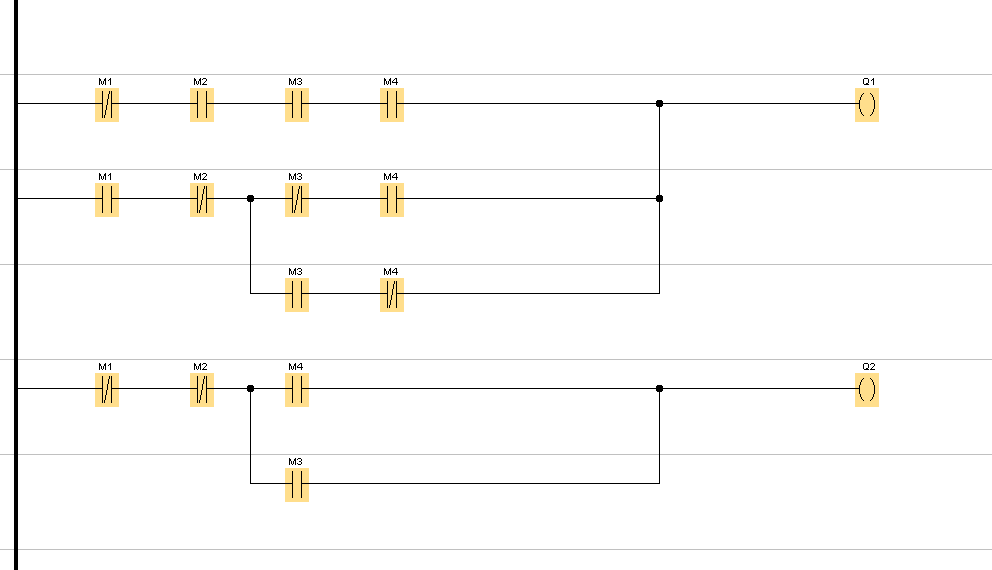
**Po minimalizacji:**

**Program w języku LD:**

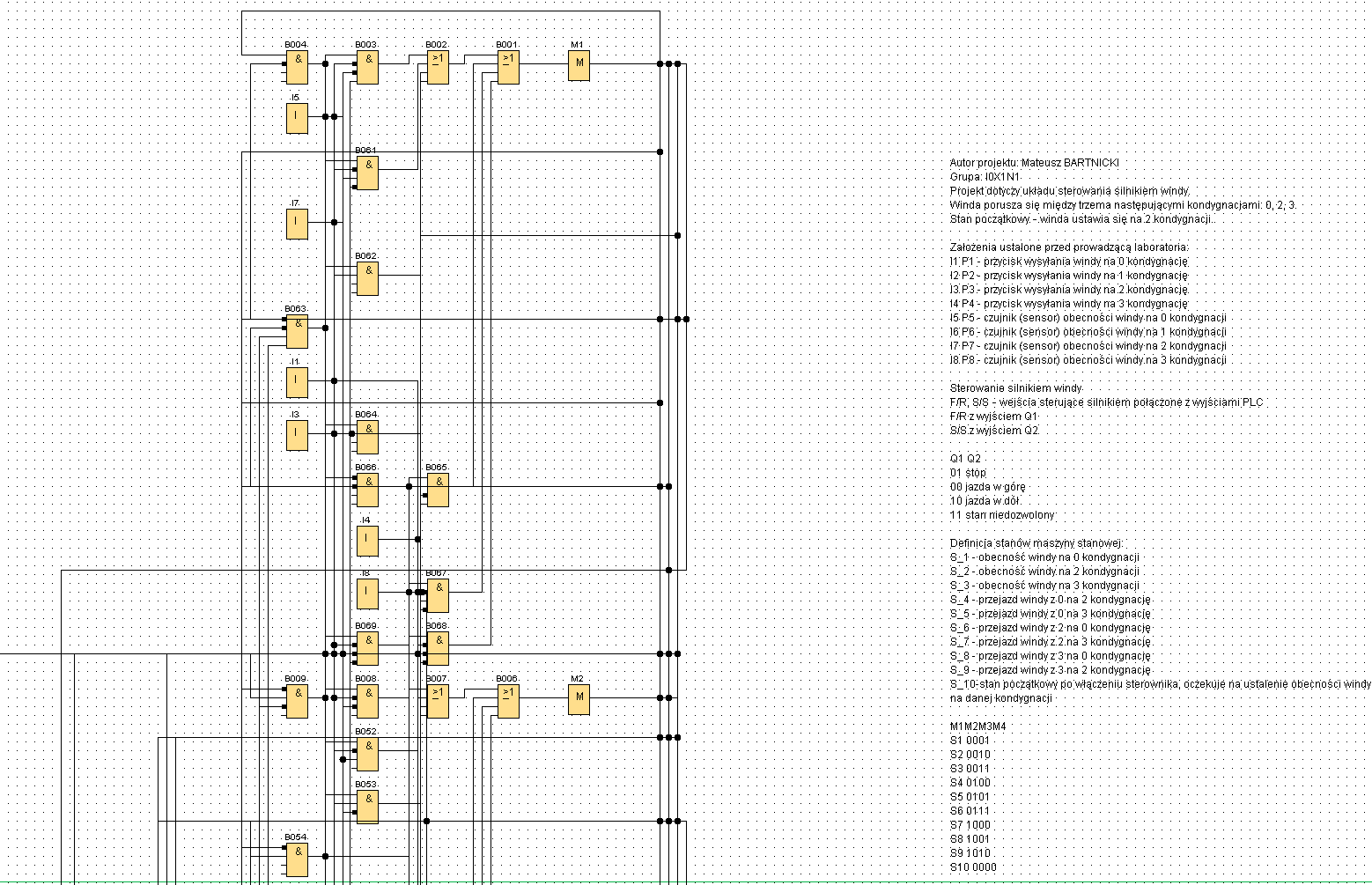
****

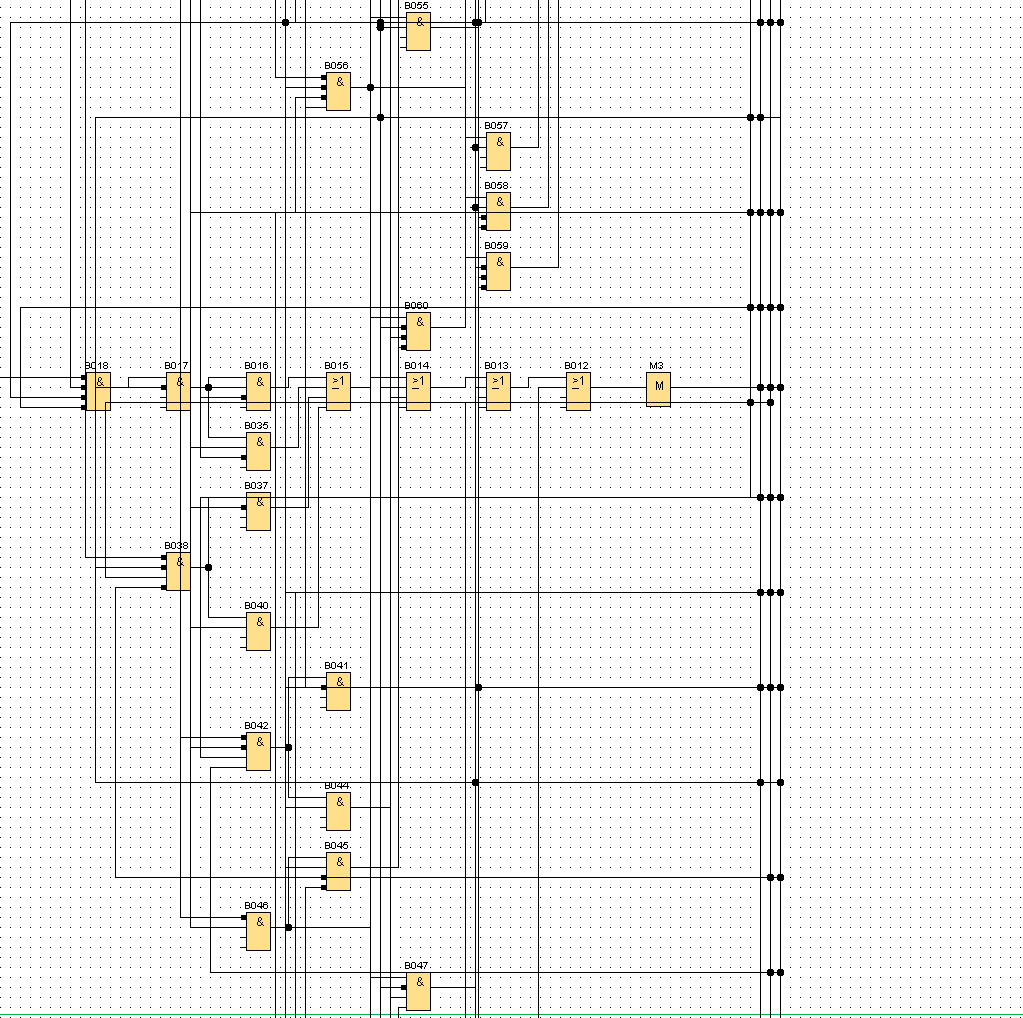
****

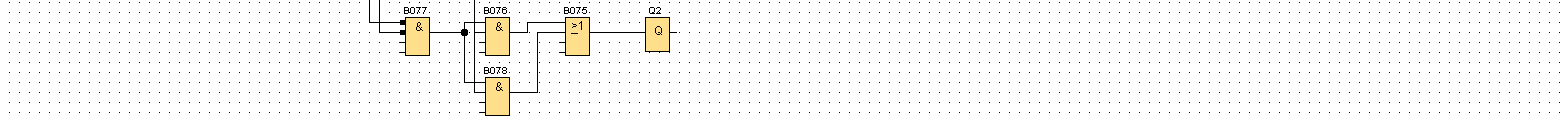
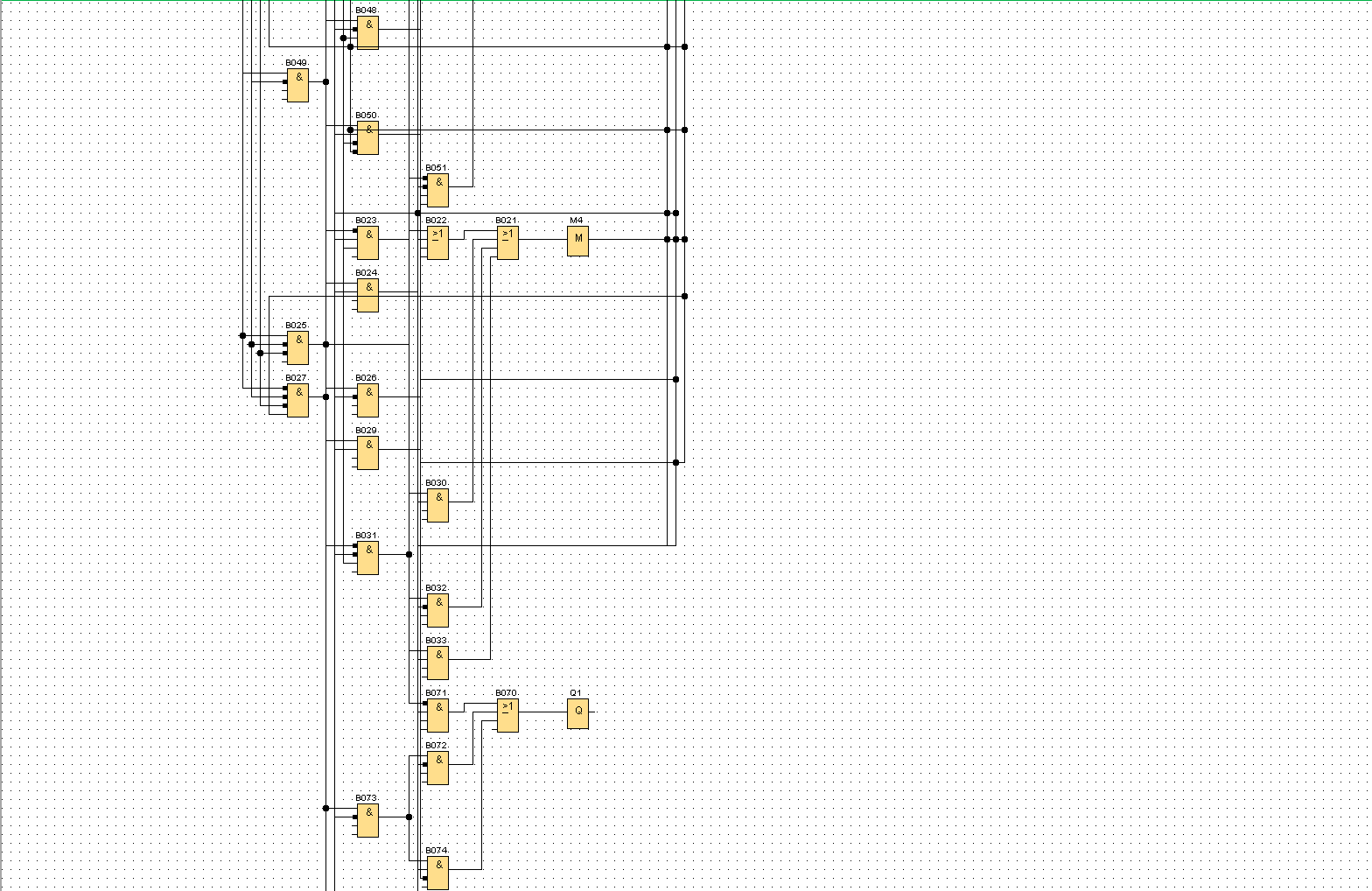
****

****

**Program w języku FBD:**

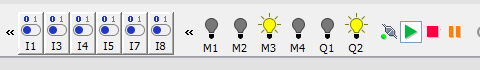
****

****

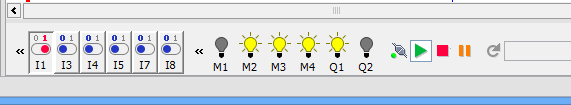
****

**WNIOSKI Z IMPLEMENTACJI W PROGRAMIE LOGO SOFT COMFORT:**

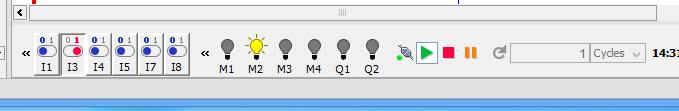
Po uruchomieniu programu następuje oczekiwanie na informację, na którym piętrze znajduje się winda. W zależności od tego, który czujnik obecności windy się uruchomił to winda porusza się w górę, w dół lub pozostaje w miejscu – tak aby na samym początku znajdować się na 2 kondygnacji. Poniżej poprawne ustawienie windy na 2 piętrze (Stan   
S2 – 0010, sterowanie silnikiem windy „stop” – 01):

****

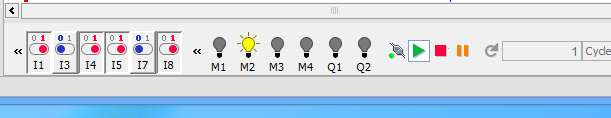
Skierowanie windy na niższą kondygnację powoduje poprawną zmianę stanu układu (na stan S6 – 0111, sterowanie silnikiem windy „w dół” – 10):

****

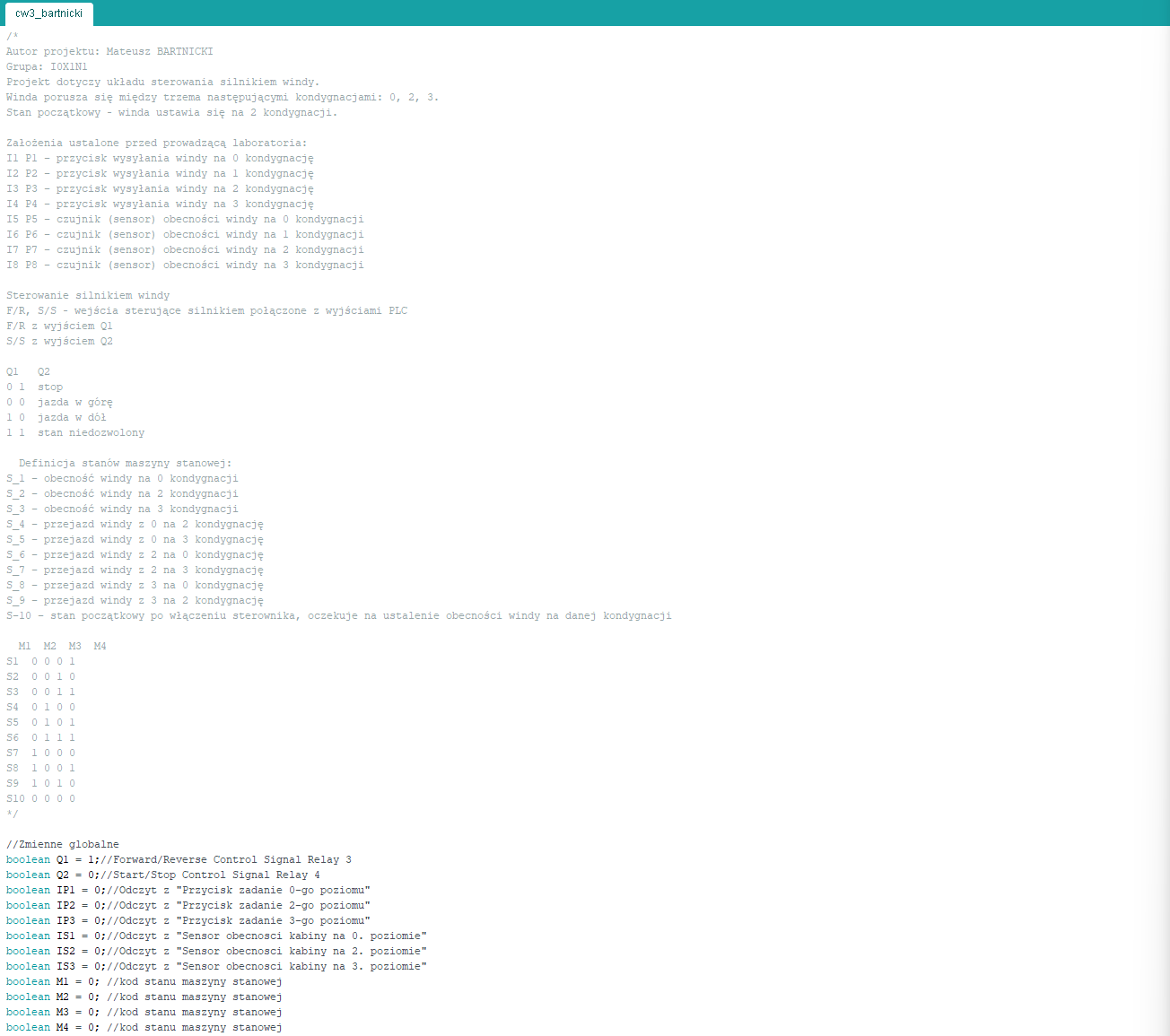
Następnie winda po dotarciu na 0 kondygnację została skierowane na 2 kondygnację. Nastąpiła poprawna zmiana stanu układu (z S1 na S4 – 0100, sterowanie silnikiem windy „w górę” – 00):

****

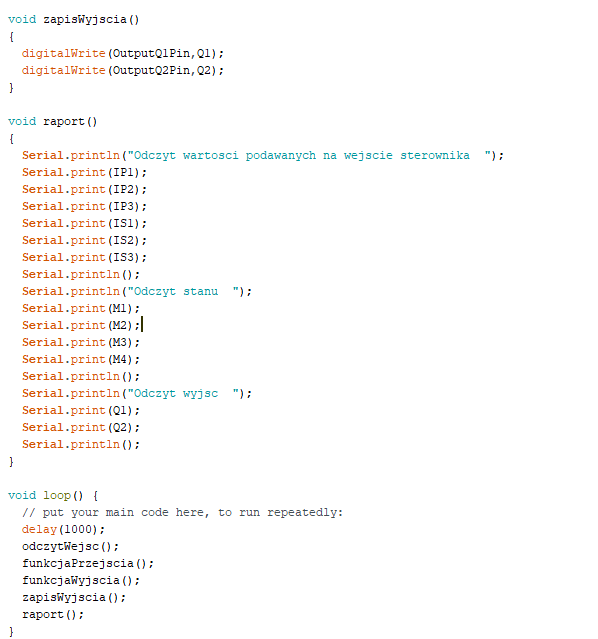
Ustawienie błędnych wartości sygnałów wejściowych nie spowodowało zmiany stanu,   
w którym znajdował się układ (pozostał w stanie S4).

****

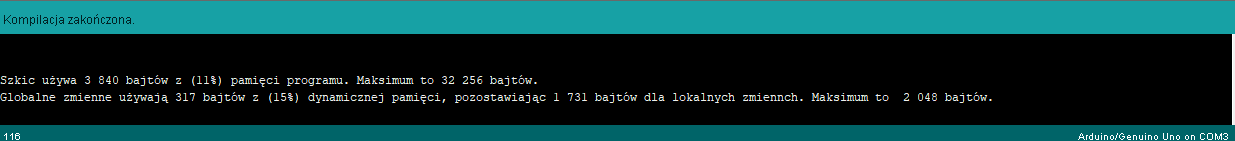
**Szkic na zestaw Arduino:**





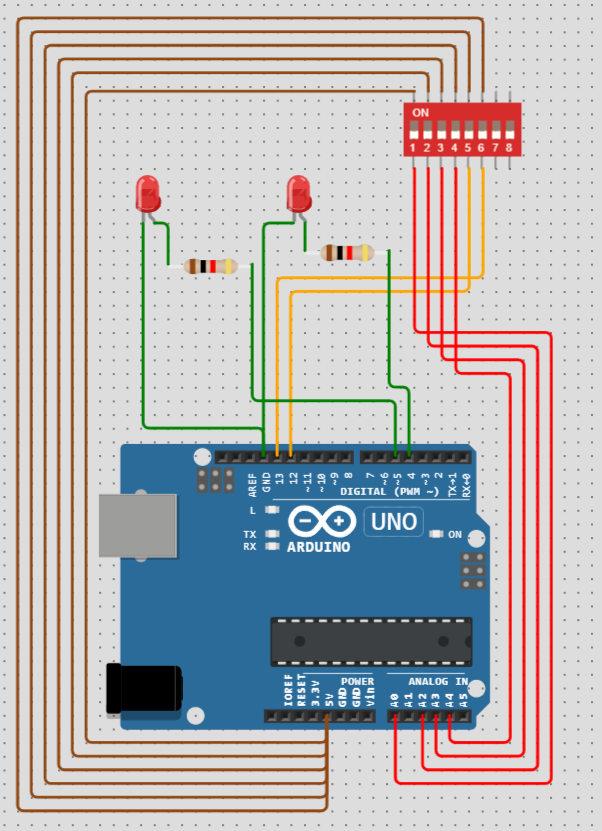


Potwierdzenie bezbłędnej kompilacji programu:



**WNIOSKI Z IMPLEMENTACJI NA ARDUINO:**

W celu sprawdzenia poprawności opracowanego szkicu na Arduino skorzystano   
z internetowego symulatora płytki Arduino znajdującego się pod adresem <https://wokwi.com/>.

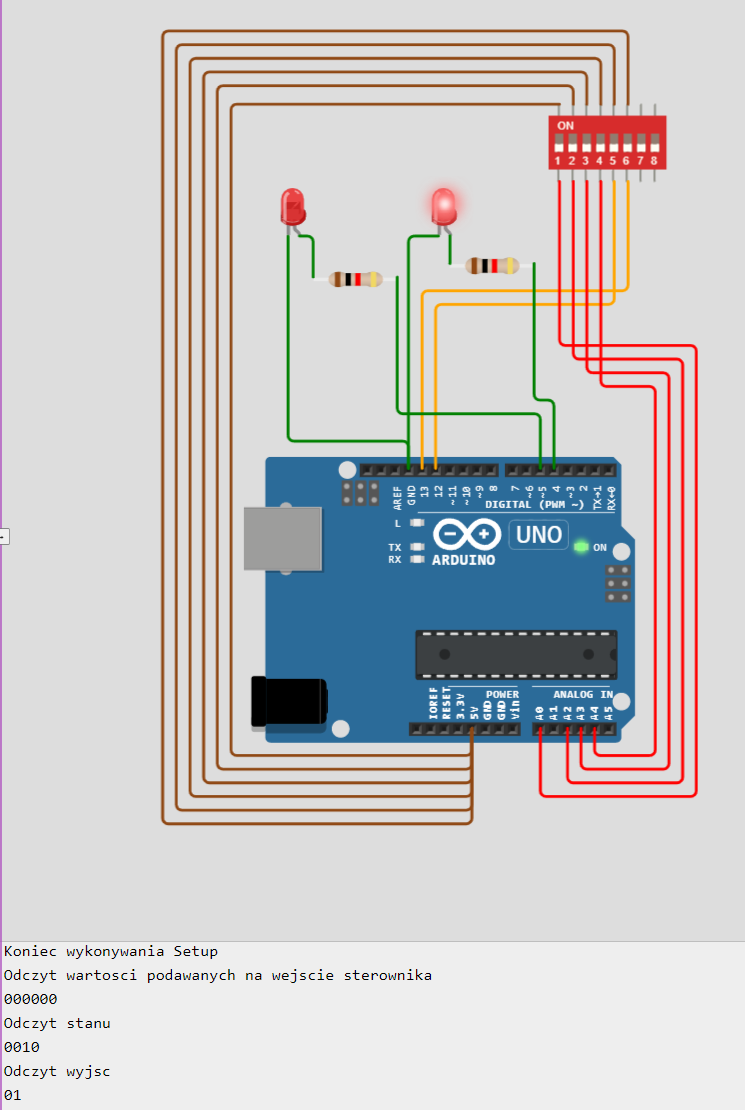


Poprawność działania szkicu może zostać zweryfikowana za pomocą komunikatów wyświetlanych na terminu oraz na podstawie diod podłączonych do wyjść układu. Dioda z lewej strony podłączona jest do wyjścia o nr 5 (F/R – bit kierunku), a dioda z prawej strony podłączona jest do wyjścia o nr 4 (S/S – bit stopu). Do sterowania wartościami sygnałów wejściowych wykorzystano 8-elementowy przełącznik, za pomocą którego możemy swobodnie zmieniać sygnały wejściowe bezpośrednio wpływając na stan,   
w którym znajduje się układ.

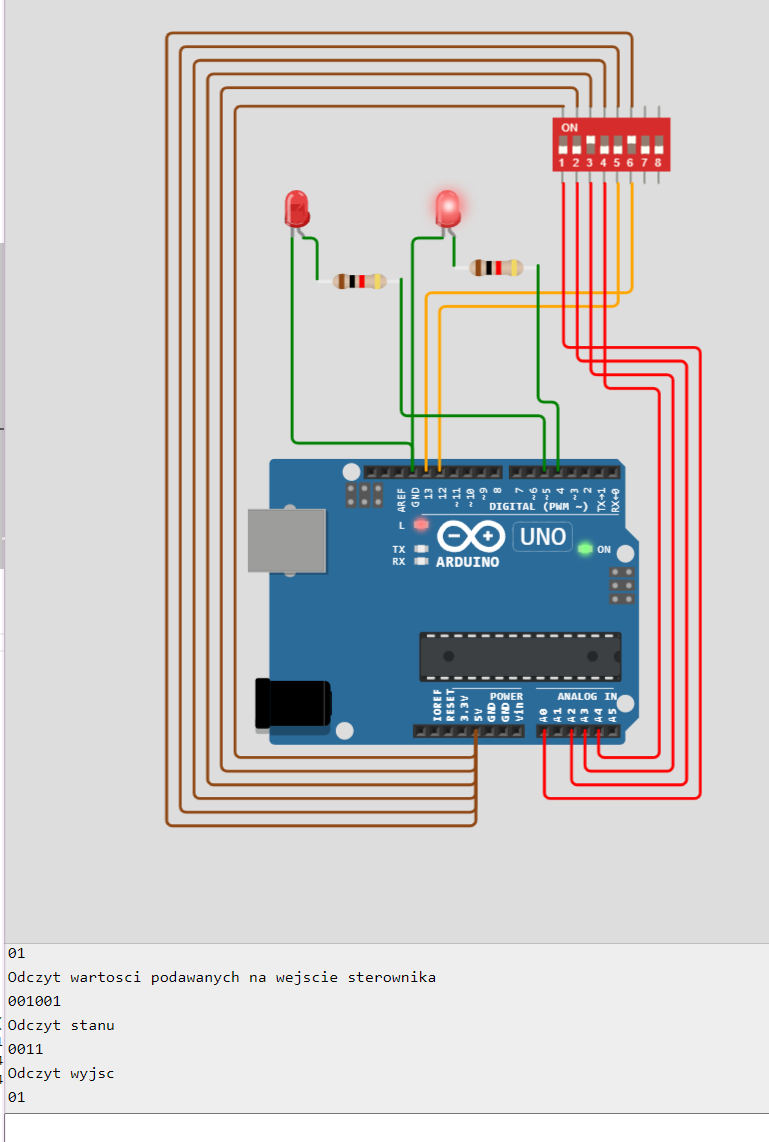
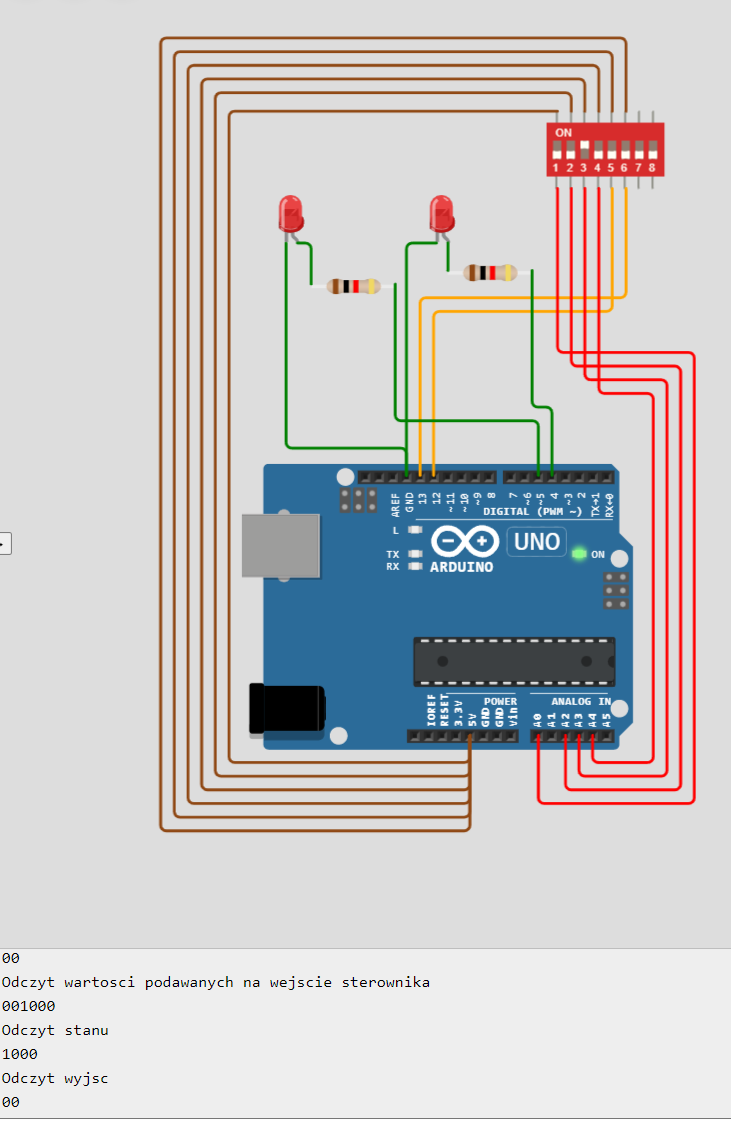
W celu weryfikacyjnym działania szkicu w funkcji głównej szkicu (void loop()) wprowadzono opóźnienie w działaniu układu, które wynosi 1s. Pozwala to na wygodną obserwację zmieniających się wyników.

**Sprawdzenie poprawności działania programu:**

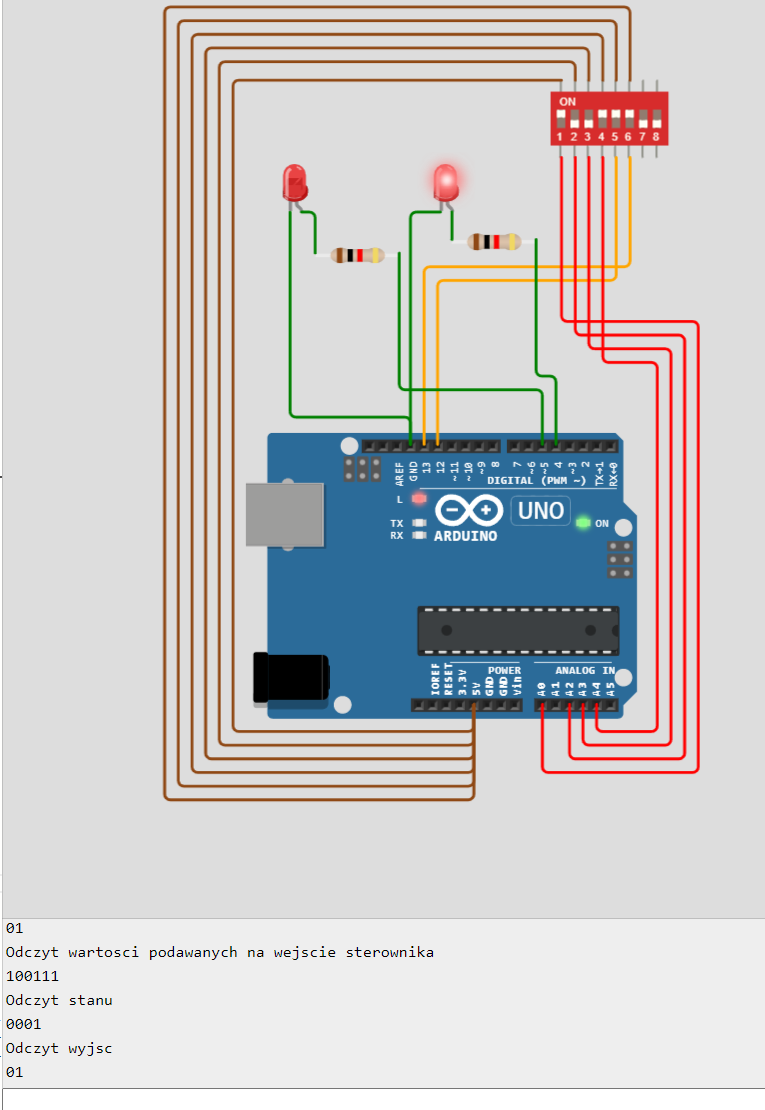
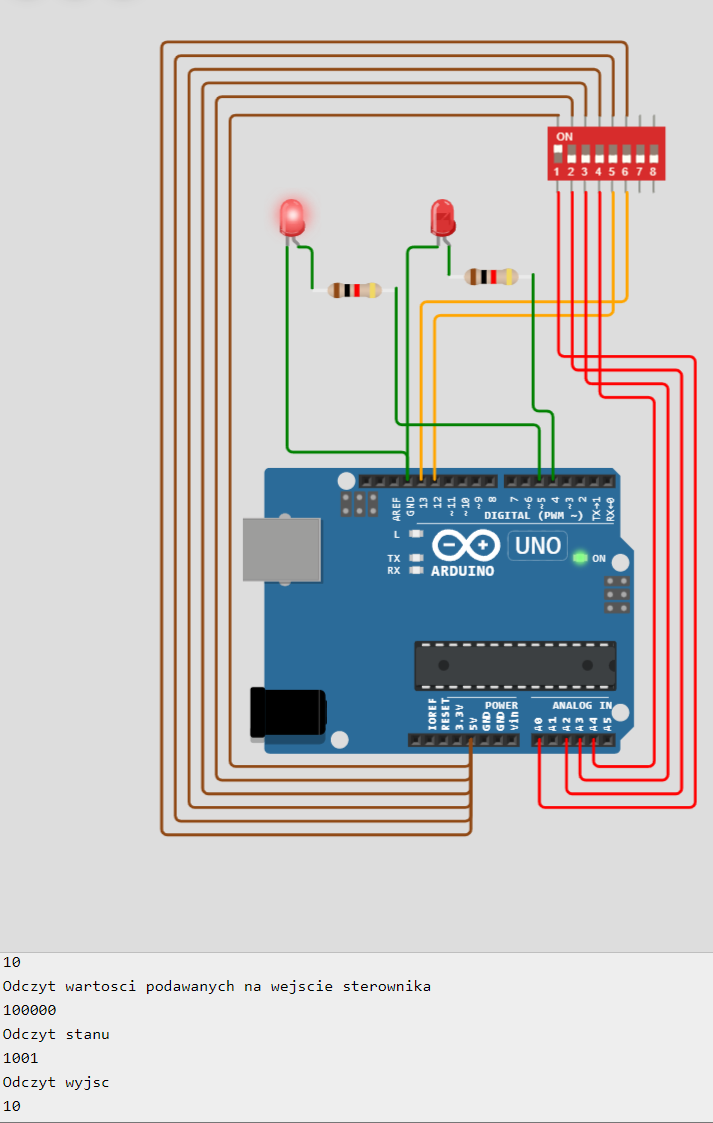
Zgodnie z założeniami do projektu, po uruchomieniu programu winda ustawić się ma na 2 kondygnacji (Stan S2 – 0010, sterowanie silnikiem windy „stop” – 01).



W przypadku skierowania windy na wyższą kondygnację następuje poprawna zmiana stanu (ze stanu S2 na stan S7 – 1000, sterowanie silnikiem „w górę” – 00). Następnie podany zostanie sygnał obecności windy na 3 piętrze.



W przypadku skierowania windy na niższą kondygnację następuje poprawna zmiana stanu (ze stanu S3 na stan S8 – 1001, sterowanie silnikiem „w dół” – 10). Układ nie zmienia swojego stanu po podaniu błędnych sygnałów na wejście – rys. z prawej.



**WNIOSKI OGÓLNE:**

Biorąc pod uwagę wszystkie założenia do projektu ustalone przez prowadzącą laboratoria oraz sposób implementacji i funkcjonowania układu uważam, że projekt został zrealizowany w stopniu co najmniej dobrym. Funkcję przejść oraz funkcje wyjść zostały poprawnie zminimalizowane (uproszczone), co pozwoliło na znaczne zmniejszenie wykorzystanych elementów do realizacji projektu. Na podstawie przeprowadzonych badań związanych z poprawnością realizacji układu, można zauważyć, że następuje poprawna zmiana stanów układu zgodnie z tabelą stanów układu oraz na wyjściu podawane są odpowiednie sygnały w zależności od stanu układu. Zaimplementowane rozwiązanie odporne jest na podawanie błędnych sygnałów wejściowych, czyli nie następuje niepożądana zmiana stanu układu w przypadku podania wartości niezgodnych z tabelą przejść układu.