

WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego

WYDZIAŁ CYBERNETYKI



Projekt zaliczeniowy Wprowadzenie do automatyki

Temat pracy:

**PROJEKT UKŁADU STEROWANIA Z WYKORZYSTANIEM
PLC I MIKROKONTROLERÓW**

Student:
Marcin KOZŁOWSKI

Warszawa 2023

1.Spis treści

1.Spis treści	2
2.Treść zadania.....	3
3.Założenia	4
4. Schemat podłączenia (PLC Siemens LOGO!)	5
5. Schemat podłączenia (Arduino Uno)	6
6. Definicja stanów maszyny stanowej	7
7. Diagram przejścia stanów	8
8. Sposób kodowania stanów	9
9. Tabela przejścia stanów	9
10. Równania funkcji przejścia	10
11. Tabela funkcji wyjścia.....	10
12. Równania funkcji wyjścia	10
13. Program w języku LD	11
14. Program w języku FBD.....	13
15. Tabulogram Arduino	16
16. Wnioski	21

2. Treść zadania

Marcin Kozłowski

data wydania tematu: 28.04.2023

Projekt układu sterowania z wykorzystaniem PLC i mikrokontrolerów

Zaprojektować układ sterowania silnikiem windy. Winda porusza się między trzema następującymi kondygnacjami: 1, 2, 3. Zadany jest stan początkowy (zachowanie windy w momencie uruchomienia programu na sterowniku PLC i na mikrokontrolerze). Stan początkowy – winda ustawia się na 1 kondygnacji

W rozwiązaniu należy podać:

- definicję stanów maszyny stanowej,
- sposób kodowania stanów,
- diagram przejść stanów,
- tabelę przejść stanów,
- wyrażenia algebraiczne do obliczania wartości funkcji przejścia,
- tabelę funkcji wyjścia,
- wyrażenia algebraiczne do obliczania wartości funkcji wyjścia,
- schemat podłączenia sterownika PLC do windy (przycisków, sensorów),

3. Założenia

Przyciski i czujniki:

- 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)

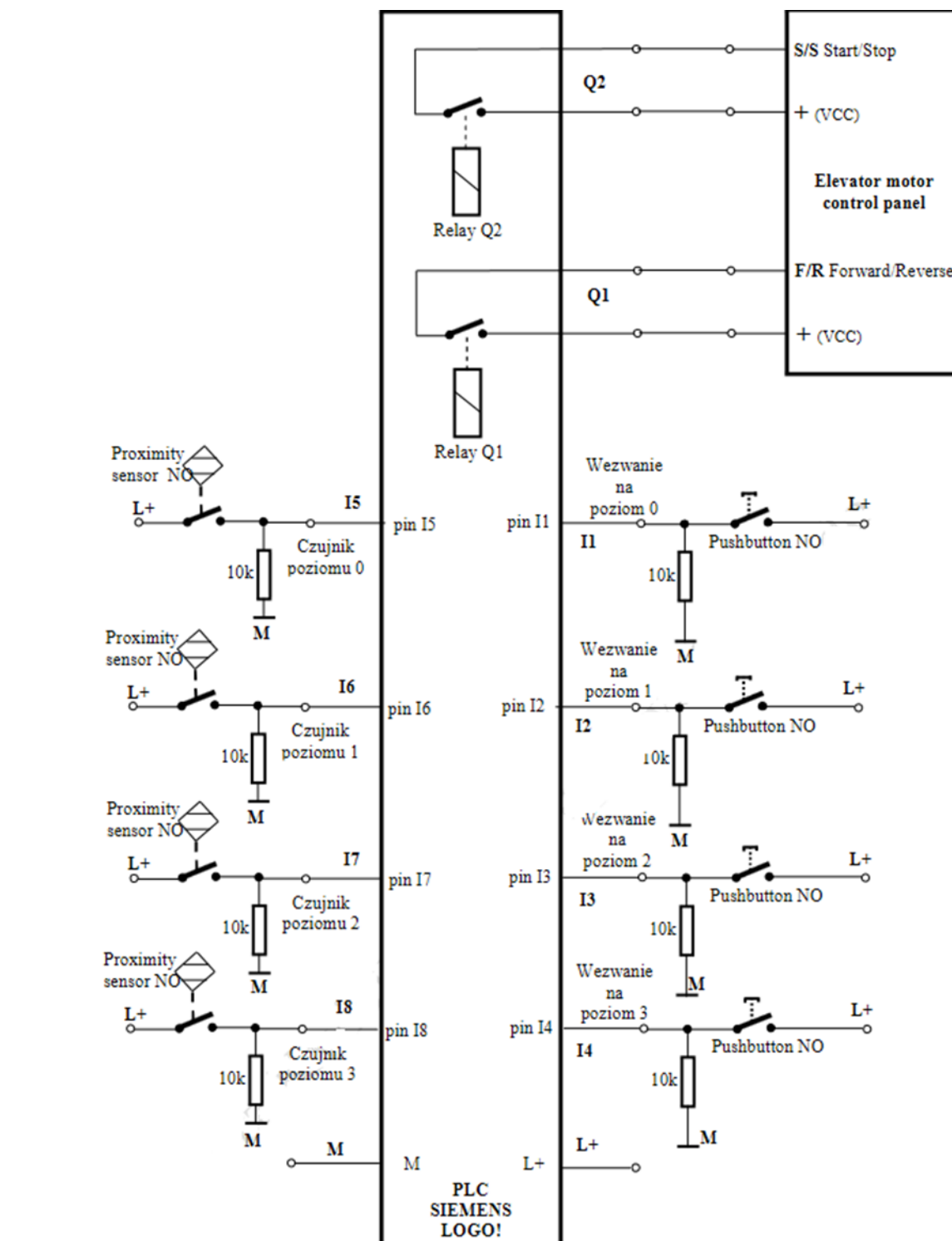
F/R (bit kierunku)	S/S (bit stopu)	reakcja
0	1	stop
0	0	↑(w górę)
1	0	↓(w dół)
1	1	nieokreślona

Sterowanie silnikiem windy (ARDUINO):

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

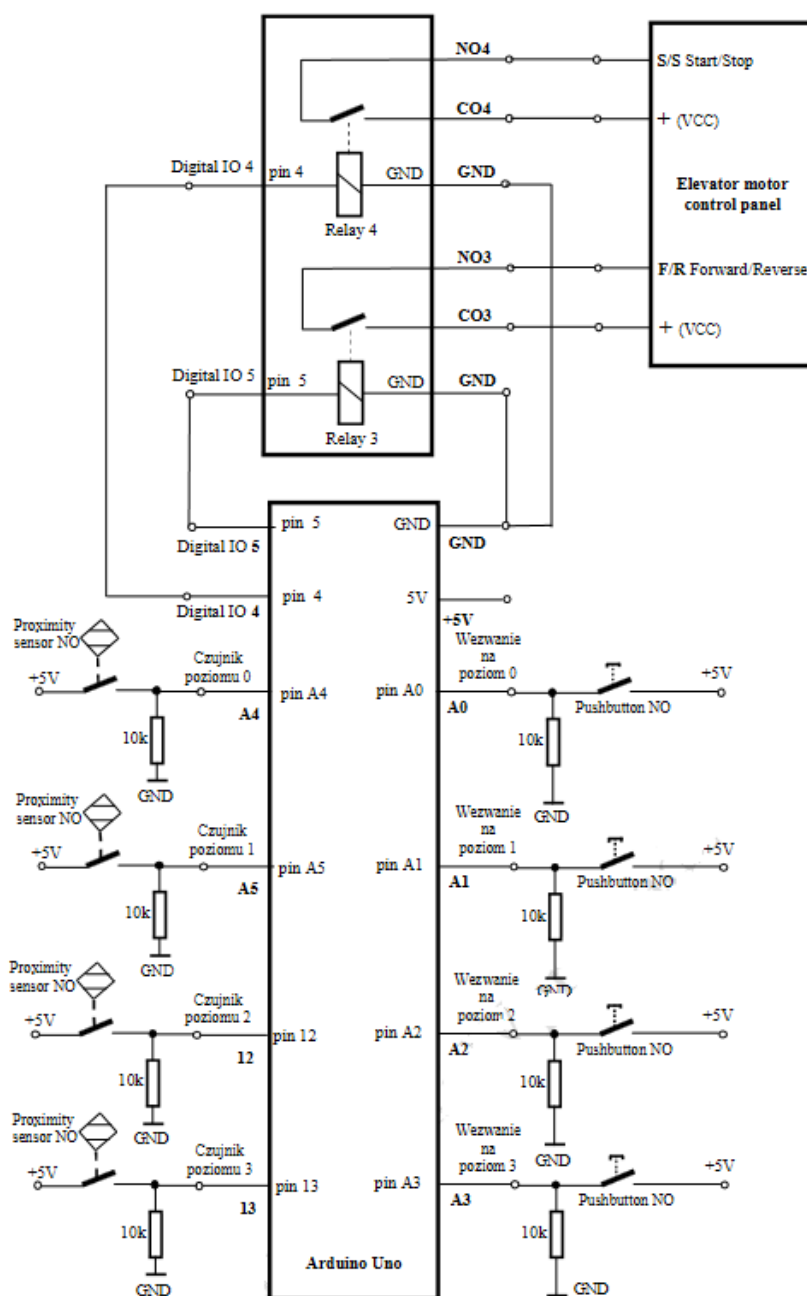
Pin 4 - S/S (bit stopu) - Q2

4. Schemat podłączenia (PLC Siemens LOGO!)



Rys 1. Sterowanie silnikiem windy – schemat PLC.

5. Schemat podłączenia (Arduino Uno)



Rys 2. Sterowanie silnikiem – schemat Arduino Uno.

6. Definicja stanów maszyny stanowej

S1 – winda uruchomiona

S2 – winda znajduje się na kondygnacji 1

S3 – winda jedzie z 1 na 2 kondygnację

S4 – winda jedzie na 3 kondygnację

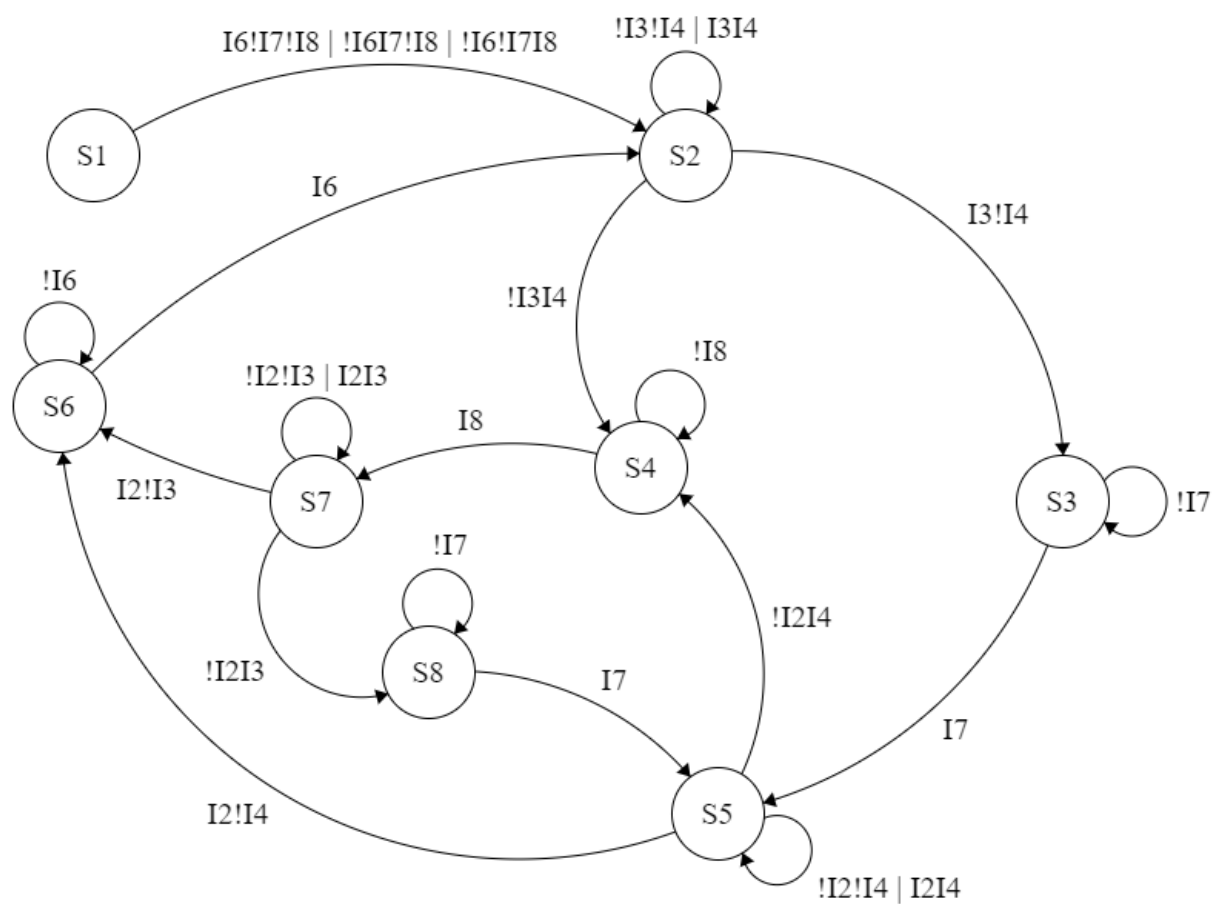
S5 – winda znajduje się na kondygnacji 2

S6 – winda jedzie na 1 kondygnację

S7 – winda znajduje się na kondygnacji 3

S8 – winda jedzie z kondygnacji 3 na 2

7. Diagram przejścia stanów



Rys 3. Diagram FSM.

8. Sposób kodowania stanów

	M1	M2	M3	M4	M5	M6	M7
S1/Start	0	0	0	0	0	0	0
S2/1	1	0	0	0	0	0	0
S3/1->2	0	1	0	0	0	0	0
S4/1,2->3	0	0	1	0	0	0	0
S5/2	0	0	0	1	0	0	0
S6/3,2->1	0	0	0	0	1	0	0
S7/3	0	0	0	0	0	1	0
S8/3->2	0	0	0	0	0	0	1

Tabela 1. Kodowanie stanów.

9. Tabela przejścia stanów

	x(t)							P2	P3	P4	P6	P7	P8	x(t+1)						
	M1	M2	M3	M4	M5	M6	M7	I2	I3	I4	I6	I7	I8	M1'	M2'	M3'	M4'	M5'	M6'	M7'
S1/Start	0	0	0	0	0	0	0	*	*	*	1	0	0	1	0	0	0	0	0	0
S1/Start	0	0	0	0	0	0	0	*	*	*	0	1	0	0	0	0	0	1	0	0
S1/Start	0	0	0	0	0	0	0	*	*	*	0	0	1	0	0	0	0	1	0	0
S2/1	1	0	0	0	0	0	0	*	0	0	*	*	*	1	0	0	0	0	0	0
S2/1	1	0	0	0	0	0	0	*	1	0	*	*	*	0	1	0	0	0	0	0
S2/1	1	0	0	0	0	0	0	*	0	1	*	*	*	0	0	1	0	0	0	0
S2/1	1	0	0	0	0	0	0	*	1	1	*	*	*	1	0	0	0	0	0	0
S3/1->2	0	1	0	0	0	0	0	*	*	*	*	1	*	0	0	0	1	0	0	0
S3/1->2	0	1	0	0	0	0	0	*	*	*	*	0	*	0	1	0	0	0	0	0
S4/1,2->3	0	0	1	0	0	0	0	*	*	*	*	*	1	0	0	0	0	0	1	0
S4/1,2->3	0	0	1	0	0	0	0	*	*	*	*	*	0	0	0	1	0	0	0	0
S5/2	0	0	0	1	0	0	0	0	*	0	*	*	*	0	0	0	1	0	0	0
S5/2	0	0	0	1	0	0	0	1	*	0	*	*	*	0	0	0	0	1	0	0
S5/2	0	0	0	1	0	0	0	0	*	1	*	*	*	0	0	1	0	0	0	0
S5/2	0	0	0	1	0	0	0	1	*	1	*	*	*	0	0	0	1	0	0	0
S6/3,2->1	0	0	0	0	1	0	0	*	*	*	1	*	*	1	0	0	0	0	0	0
S6/3,2->1	0	0	0	0	1	0	0	*	*	*	0	*	*	0	0	0	0	1	0	0
S7/3	0	0	0	0	0	1	0	0	0	*	*	*	*	0	0	0	0	0	1	0
S7/3	0	0	0	0	0	1	0	1	0	*	*	*	*	0	0	0	0	1	0	0
S7/3	0	0	0	0	0	1	0	0	1	*	*	*	*	0	0	0	0	0	0	1
S7/3	0	0	0	0	0	1	0	1	1	*	*	*	*	0	0	0	0	0	1	0
S8/3->2	0	0	0	0	0	0	1	*	*	*	*	1	*	0	0	0	1	0	0	0
S8/3->2	0	0	0	0	0	0	1	*	*	*	*	0	*	0	0	0	0	0	0	1

Tabela 2. Przejścia stanów.

10. Równania funkcji przejścia

$$M1' = \overline{M1} \overline{M2} \overline{M3} \overline{M4} \overline{M5} \overline{M6} \overline{M7} I6 \overline{I7} \overline{I8} + M1 \overline{I3} \overline{I4} + M5 I6 + M1 I3 I4$$

$$M2' = M1 I3 \overline{I4} + M2 \overline{I7}$$

$$M3' = M1 \overline{I3} I4 + M3 \overline{I8} + M4 \overline{I2} I4$$

$$M4' = M2 I7 + M4 \overline{I2} \overline{I4} + M7 I7 + M4 I2 I4$$

$$M5' = \overline{M1} \overline{M2} \overline{M3} \overline{M4} \overline{M5} \overline{M6} \overline{M7} \overline{I6} (I7 \overline{I8} + \overline{I7} I8) + M4 I2 \overline{I4} + M5 \overline{I6} + M6 I2 \overline{I3} \overline{I4}$$

$$M6' = M3 I8 + M6 \overline{I2} \overline{I3} + M6 I2 I3$$

$$M7' = M6 \overline{I2} I3 + M7 \overline{I7}$$

11. Tabela funkcji wyjścia

	M1	M2	M3	M4	M5	M6	M7
Q1	0	0	0	0	1	0	1
Q2	1	0	0	1	0	1	0

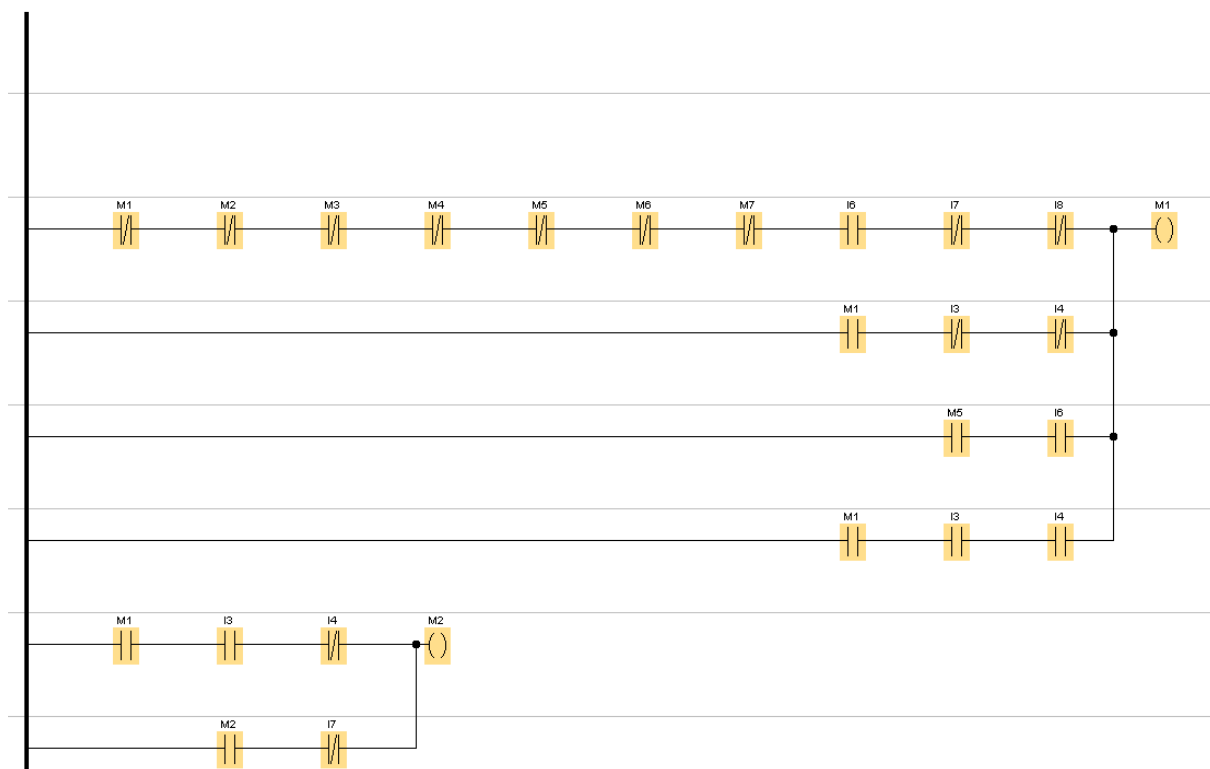
Tabela 3. Tabela funkcji wyjścia.

12. Równania funkcji wyjścia

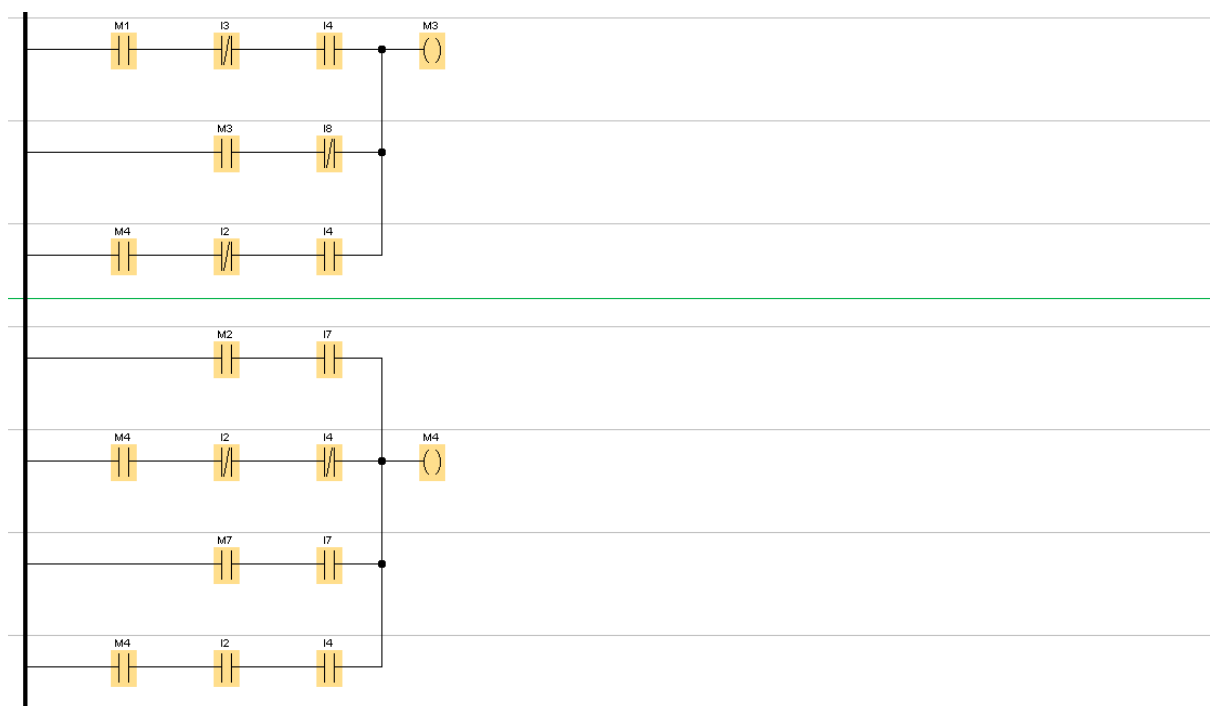
$$Q1 = M5 + M7$$

$$Q2 = M1 + M4 + M6$$

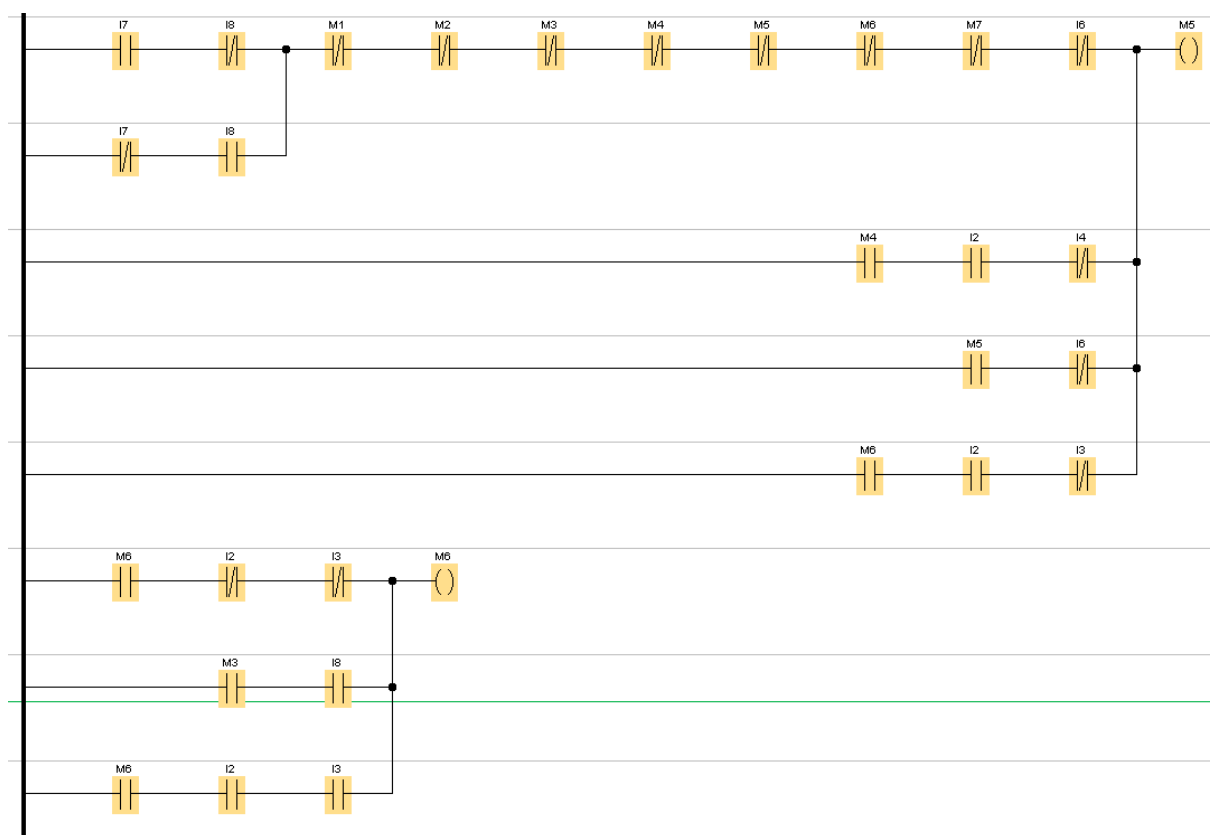
13. Program w języku LD



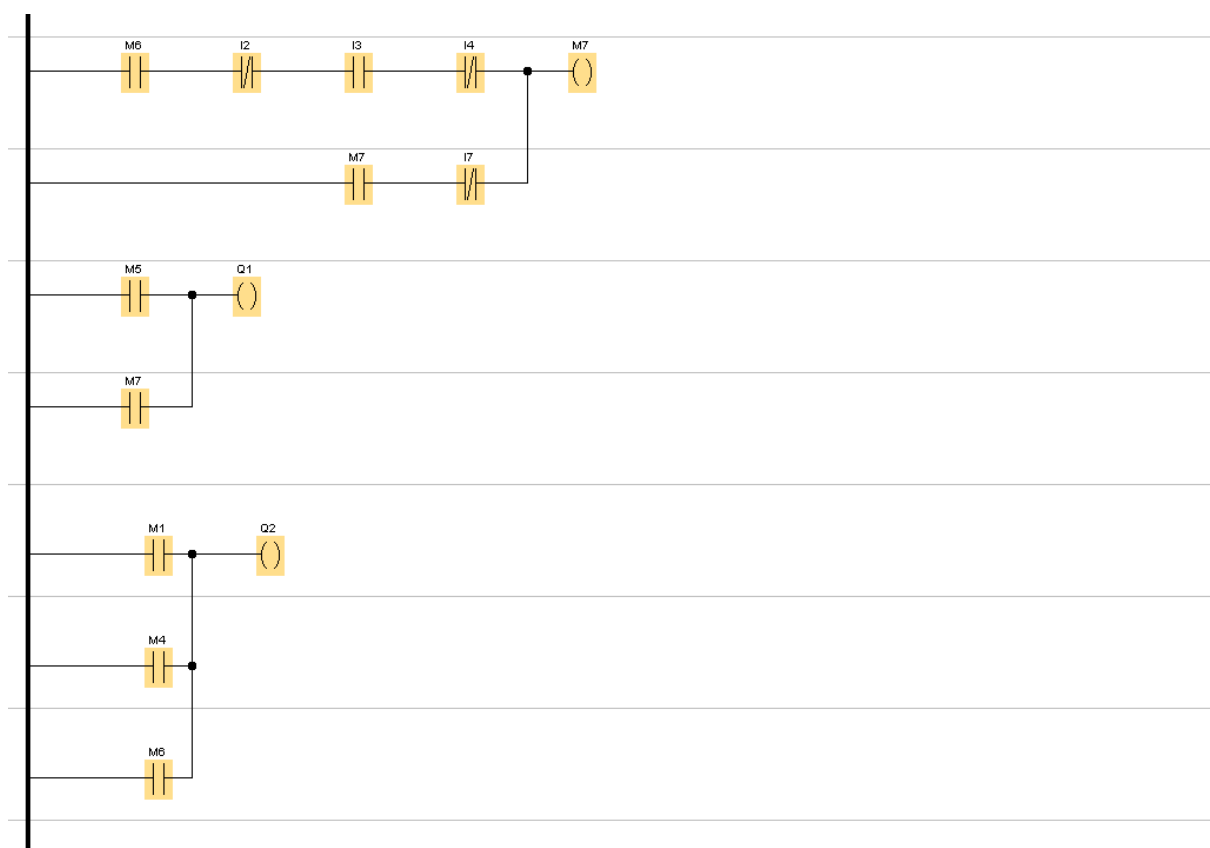
Rys 4. Diagram LD.



Rys 5. Diagram LD – c.d.

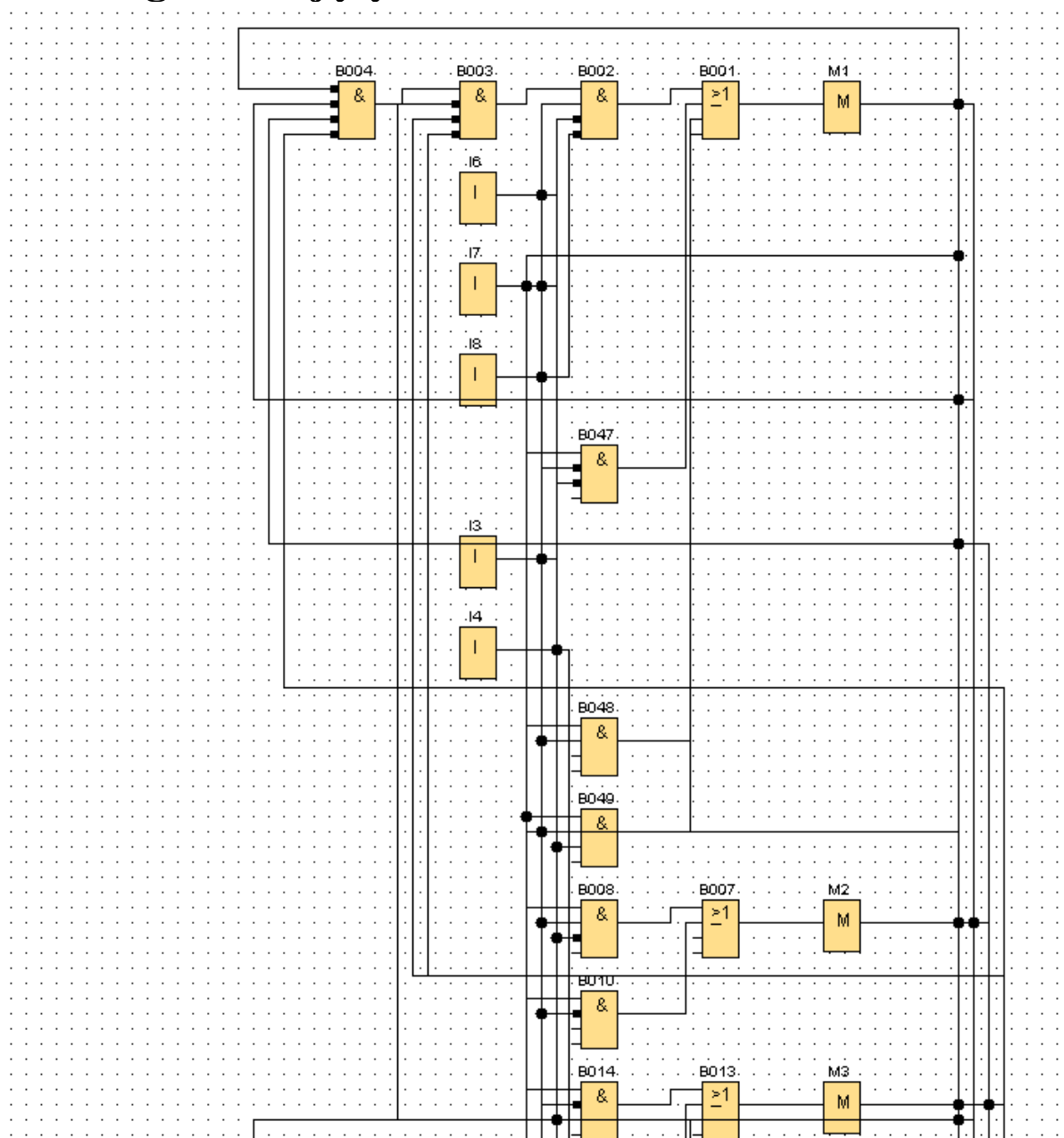


Rys 6. Diagram LD – c.d.

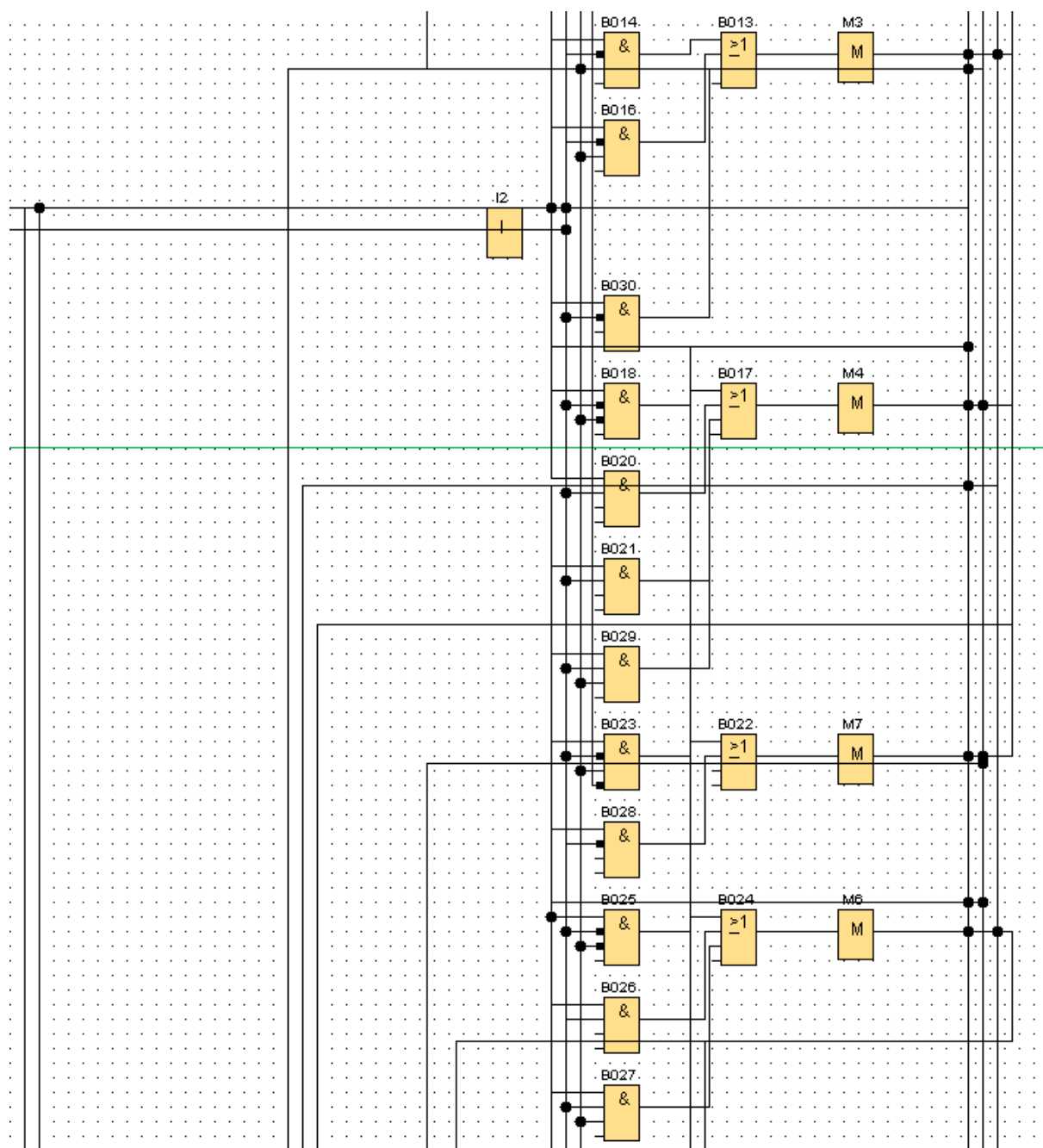


Rys 7. Diagram LD – c.d.

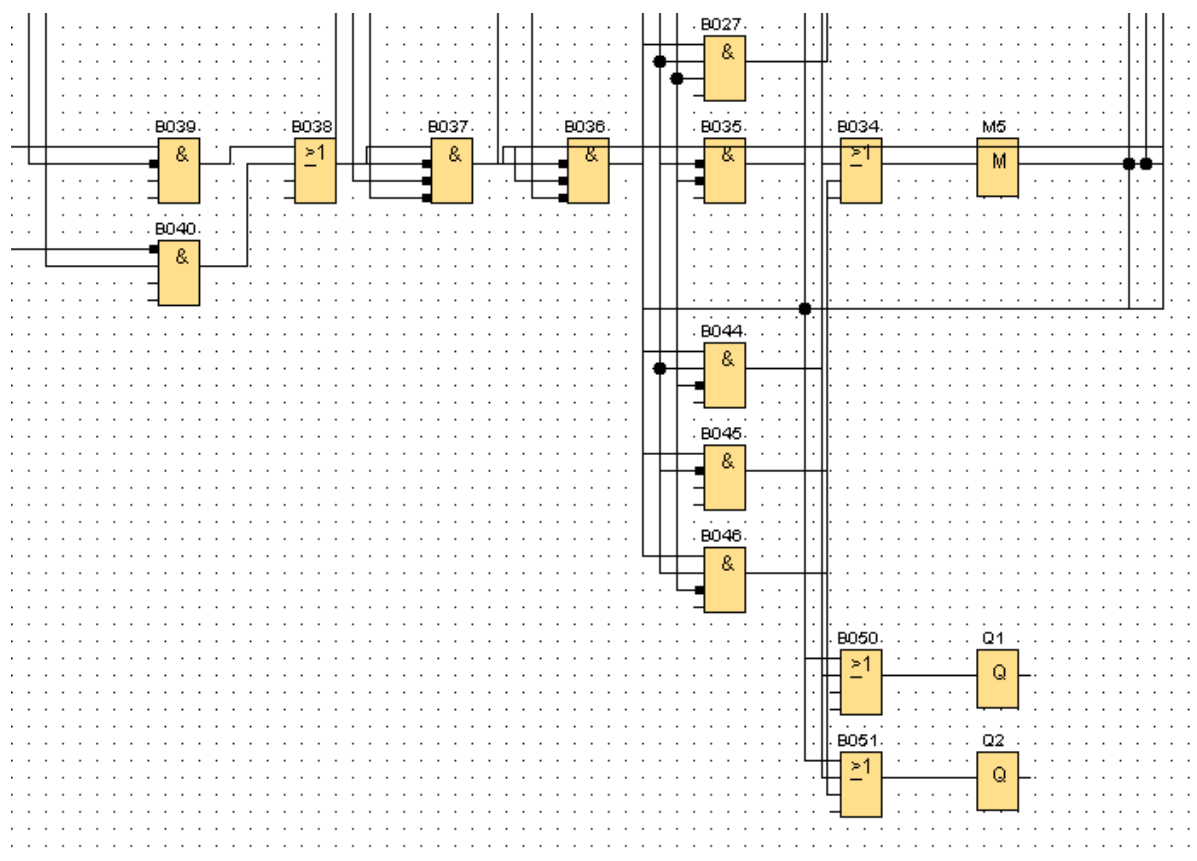
14. Program w języku FBD



Rys 8. Diagram FBD.



Rys 9. Diagram FBD – c.d.



Rys 10. Diagram FBD – c.d.

15. Tabulogram Arduino

```
#define czyRaport 1

#define lag 0

//Przylacza Arduino

int Button1Pin = A1; //nazwa pinu dla "Przycisk żądanie 1-go poziomu"
int Button2Pin = A2; //nazwa pinu dla "Przycisk żądanie 2-go poziomu"
int Button3Pin = A3; //nazwa pinu dla "Przycisk żądanie 3-go poziomu"
int Sensor1Pin = A5; //nazwa pinu dla "Sensor obecności kabiny na 1-wszym poziomie"
int Sensor2Pin = 12; //nazwa pinu dla "Sensor obecności kabiny na 2-gim poziomie"
int Sensor3Pin = 13; //nazwa pinu dla "Sensor obecności kabiny na 3-cim poziomie"
int OutputQ2Pin = 4; //numer pinu dla Control Signal of Relay 4 / Output Signal Q2 /
Start/Stop Signal
int OutputQ1Pin = 5; //numer pinu dla Control Signal of Relay 3 / Output Signal Q1 /
Forward/Reverse Signal

//Zmienne globalne
boolean Q1 = 0; //Forward/Reverse Control Signal Relay 3
boolean Q2 = 0; //Start/Stop Control Signal Relay 4
boolean I2 = 0; //Odczyt z "Przycisk żądanie 1-go poziomu"
boolean I3 = 0; //Odczyt z "Przycisk żądanie 2-go poziomu"
boolean I4 = 0; //Odczyt z "Przycisk żądanie 3-go poziomu"
boolean I6 = 0; //Odczyt z "Sensor obecności kabiny na 1. poziomie"
boolean I7 = 0; //Odczyt z "Sensor obecności kabiny na 2. poziomie"
boolean I8 = 0; //Odczyt z "Sensor obecności kabiny na 3. poziomie"

//Flagi stanów
boolean M1 = 0;
boolean M2 = 0;
boolean M3 = 0;
boolean M4 = 0;
boolean M5 = 0;
boolean M6 = 0;
boolean M7 = 0;
```



```
boolean M1p = 0;
boolean M2p = 0;
boolean M3p = 0;
boolean M4p = 0;
boolean M5p = 0;
boolean M6p = 0;
boolean M7p = 0;
void raport()
{
    Serial.println("Odczyt przyciskow ");
    Serial.print(I2);
    Serial.print(I3);
    Serial.print(I4);
    Serial.println();
    Serial.println("Odczyt sensorow ");
    Serial.print(I6);
    Serial.print(I7);
    Serial.print(I8);
    Serial.println();
    Serial.println("Flagi ");
    Serial.print(M1);
    Serial.print(M2);
    Serial.print(M3);
    Serial.print(M4);
    Serial.print(M5);
    Serial.print(M6);
    Serial.print(M7);
    Serial.println();
    Serial.println("Wyjscia ");
    Serial.print(Q1);
```

```

Serial.print(Q2);
Serial.println();
}
void odczytWejsc()
{
    I2 = digitalRead(Button1Pin);
    I3 = digitalRead(Button2Pin);
    I4 = digitalRead(Button3Pin);
    I6 = digitalRead(Sensor1Pin);
    I7 = digitalRead(Sensor2Pin);
    I8 = digitalRead(Sensor3Pin);
}
void funkcjaPrzejscia()
{
    //Obliczenie wartości funkcji przejść stanów (flag)
    M1p = !M1&!M2&!M3&!M4&!M5&!M6&!M7&I6&!I7&!I8 | M1&!I3&!I4 | M5&I6 |
    M1&I3&I4;
    M2p = M1&I3&!I4 | M2&!I7;
    M3p = M1&!I3&I4 | M3&!I8 | M4&!I2&I4;
    M4p = M2&I7 | M4&!I2&I4 | M7&I7 | M4&I2&I4;
    M5p = !M1&!M2&!M3&!M4&!M5&!M6&!M7&!I6&(I7&!I8 | !I7&I8) | M4&I2&!I4 |
    M5&!I6 | M6&I2&!I3&!I4;
    M6p = M3&I8 | M6&!I2&!I3 | M6&I2&I3;
    M7p = M6&!I2&I3 | M7&!I7;
    //Przepisanie "nowych" wartości do "starych"
    M1 = M1p;
    M2 = M2p;
    M3 = M3p;
    M4 = M4p;
    M5 = M5p;
    M6 = M6p;

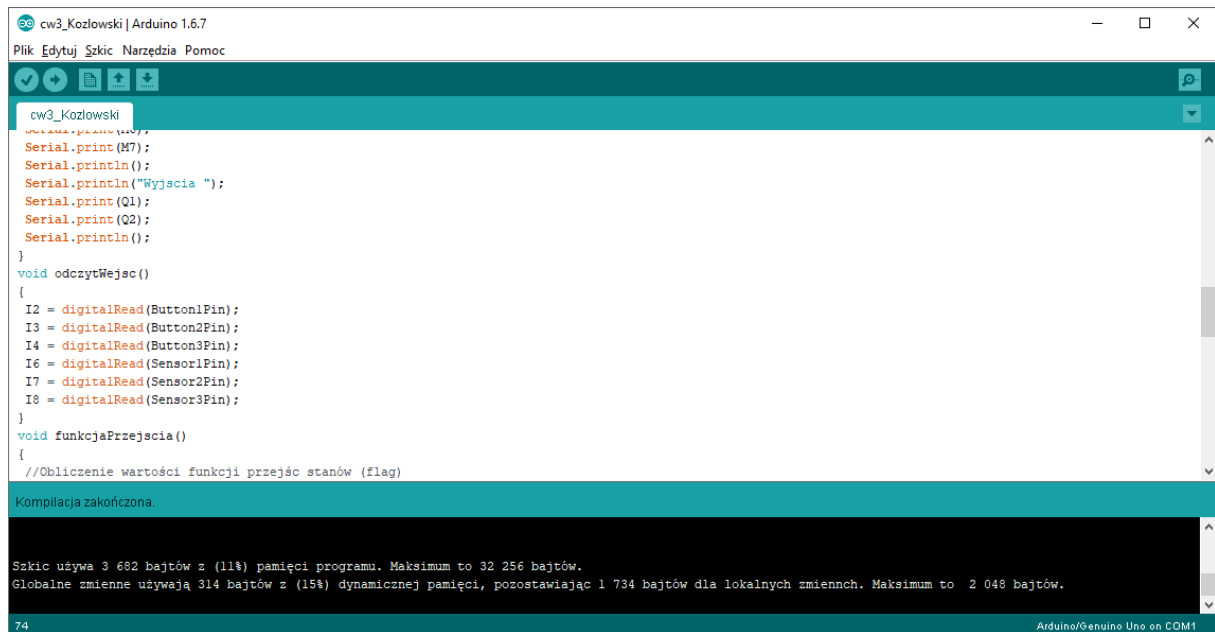
```

```

M7 = M7p;
}
void funkcjaWyjscia()
{
    Q1 = M5 | M7;
    Q2 = M1 | M4 | M6;
}
void zapisWyjscia()
{
    digitalWrite(OutputQ1Pin, Q1);
    digitalWrite(OutputQ2Pin, Q2);
}
void setup()
{
    pinMode(Button1Pin, INPUT);
    pinMode(Button2Pin, INPUT);
    pinMode(Button3Pin, INPUT);
    pinMode(Sensor1Pin, INPUT);
    pinMode(Sensor2Pin, INPUT);
    pinMode(Sensor3Pin, INPUT);
    pinMode(OutputQ1Pin, OUTPUT);
    pinMode(OutputQ2Pin, OUTPUT);
    Serial.begin(9600);
    Serial.println("Koniec wykonywania Setup");
}
void loop()
{
    odczytWejsc();
    funkcjaPrzejscia();
    funkcjaWyjscia();
}

```

```
zapisWyjscia();  
  
raport();  
  
}
```



Rys 11. Kompilacja programu.

16. Wnioski

Zrealizowano projekt windy zgodnie z wytycznymi. Sprawdzono poprawność przejść stanów w programie LOGO!Soft. Winda realizuje poprawne przejścia zaprogramowane dla podanych wartości stanów.