

Wojskowa Akademia Techniczna

Wprowadzenie do Automatyki

Sprawozdanie z pracy laboratoryjnej nr 3

**Projekt układu sterowania z wykorzystaniem
PLC i mikrokontrolerów**

Data wykonania ćwiczenia: 18.04.2024 r.

Prowadzący ćwiczenia: mgr inż. Małgorzata Rudnicka-Schmidt

Wykonał: Szymon Florek

Grupa: WCY22IY2S1

Treść zadania:

Wariant 102

Symon Florek

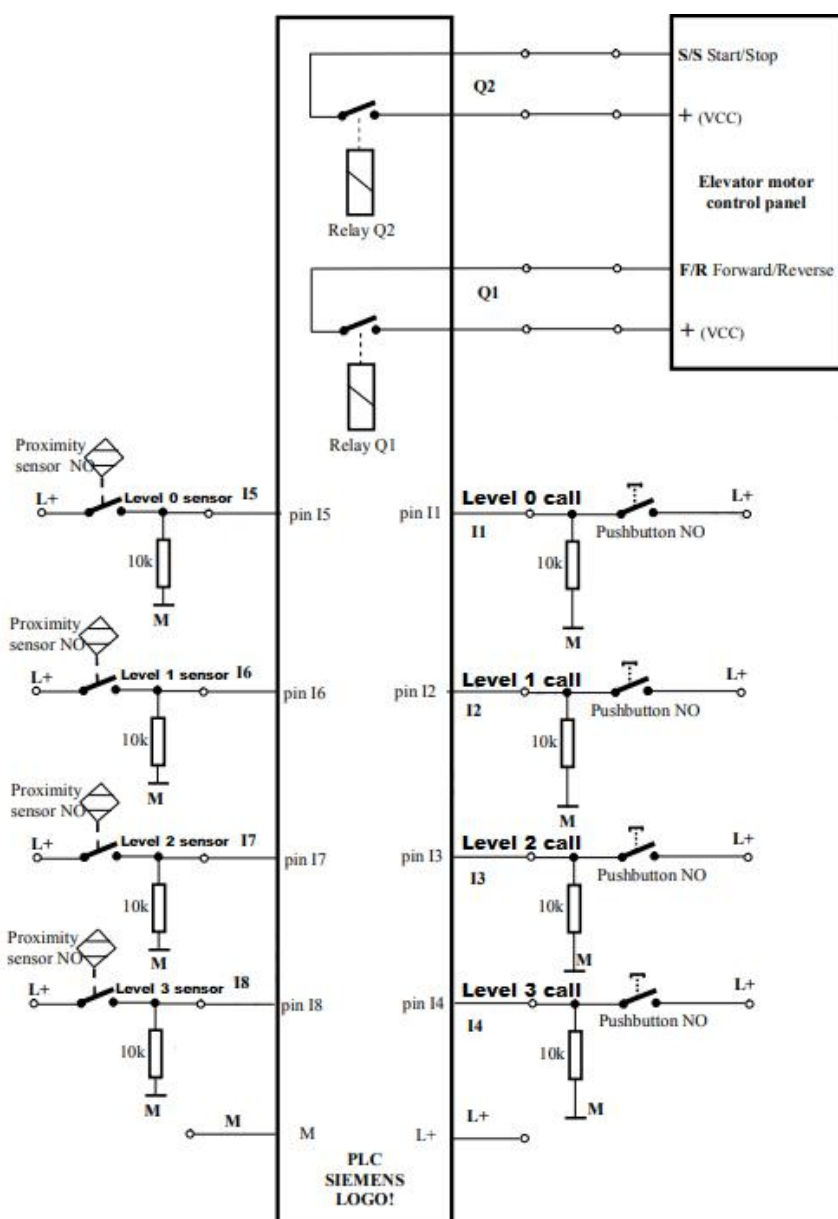
Ćwiczenie 3: Projekt układu sterowania z wykorzystaniem PLC i mikrokontrolerów

Zaprojektować układ sterowania silnikiem windy.

Winda porusza się między trzema następującymi kondygnacjami: 0, 1, 2.

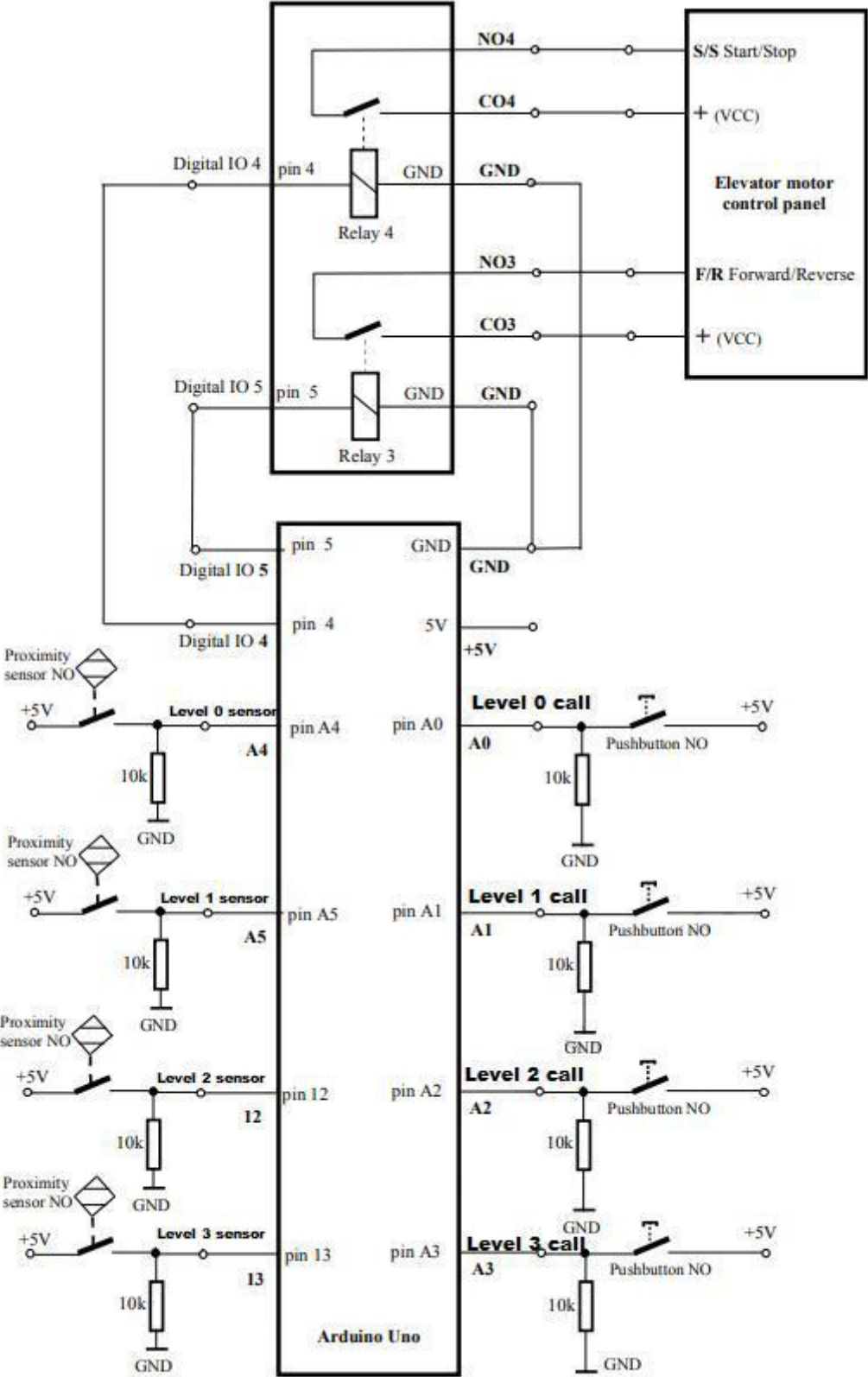
Zadany jest stan początkowy (zachowanie windy w momencie uruchomienia programu na sterowniku PLC i na mikrokontrolerze). Stan początkowy – winda ustawia się na 1 kondygnacji

Schemat podłączenia PLC:



Schemat podłączenia Arduino:

Fig. 2.10. 127



Założenia

- 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)
```

Definicja stanów maszyny stanowej:

- * **0000 0** – winda ustawia się na **1 piętrze** – stan początkowy
- * **0001 1** – winda znajduje się na 0 piętrze
- * **0010 2** – winda znajduje się na 1 piętrze
- * **0011 3** – winda znajduje się na 2 piętrze
- * **0100 4** – winda porusza się z 0 na 1 piętro
- * **0101 5** – winda porusza się z 0 na 2 piętro
- * **0110 6** – winda porusza się z 1 na 2 piętro
- * **0111 7** – winda porusza się z 1 na 0 piętro
- * **1000 8** – winda porusza się z 2 na 1 piętro
- * **1001 9** – winda porusza się z 2 na 0 piętr

Sposób kodowania stanów maszyny stanowej:

Stan	X1	X2	X3	X4
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Diagram przejść maszyny stanowej:

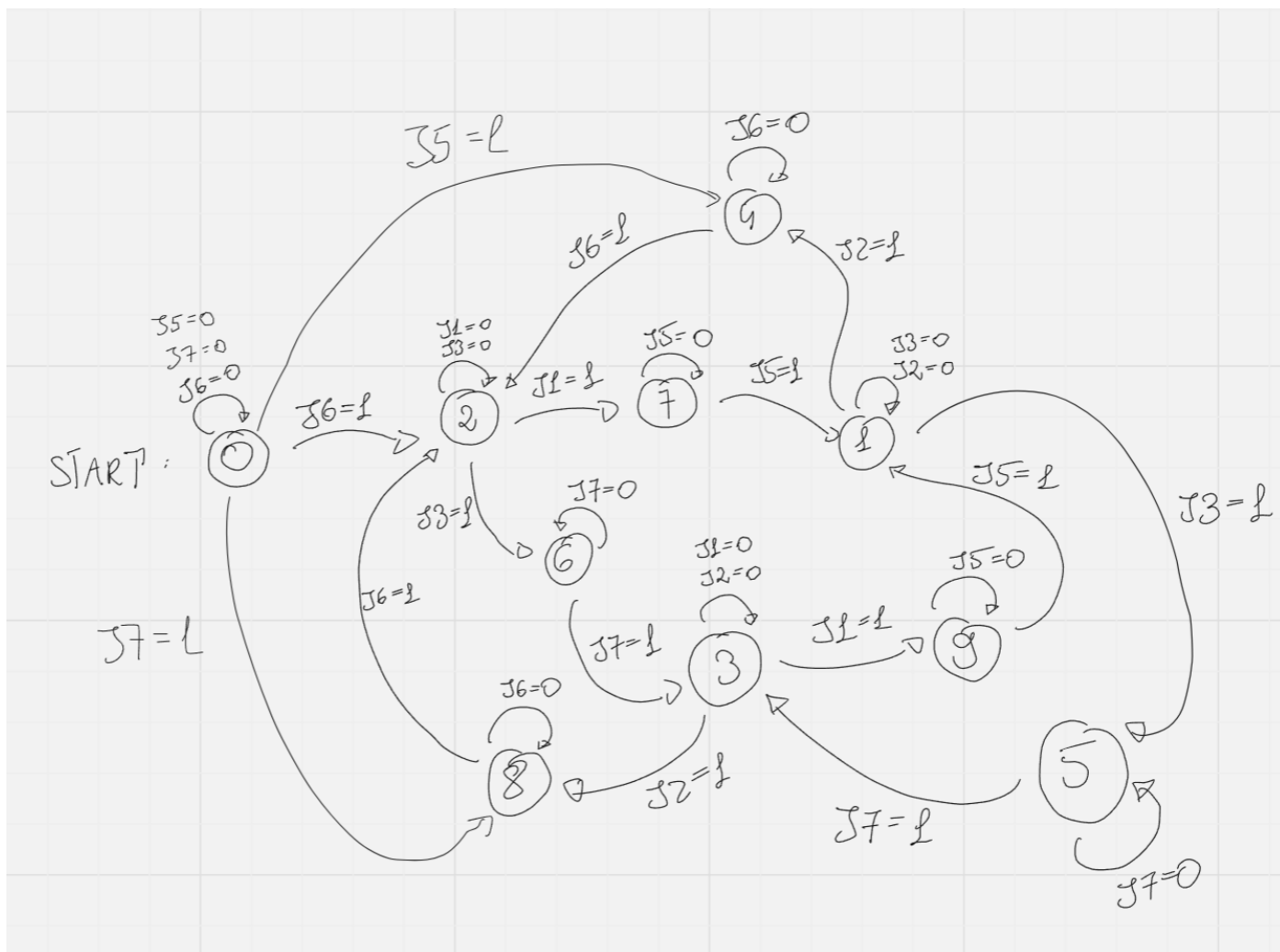


Tabela przejść stanów:

X1(t)	X2(t)	X3(t)	X4(t)	I1	I2	I3	I5	I6	I7	X1(t+1)	X2(t+1)	X3(t+1)	X4(t+1)	STANY
0	0	0	0	*	*	*	0	*	*	0	0	0	0	0
0	0	0	0	*	*	*	1	*	*	0	1	0	0	
0	0	0	0	*	*	*	*	*	0	0	0	0	0	
0	0	0	0	*	*	*	*	*	1	1	0	0	0	
0	0	0	0	*	*	*	*	0	*	0	0	0	0	0
0	0	0	0	*	*	*	*	1	*	0	0	1	0	0
0	0	0	1	*	1	*	*	*	*	0	1	0	0	1
0	0	0	1	*	*	1	*	*	*	0	1	0	1	
0	0	1	0	1	*	*	*	*	*	0	1	1	1	2
0	0	1	0	*	*	1	*	*	*	0	1	1	0	
0	0	1	1	1	*	*	*	*	*	1	0	0	1	3
0	0	1	1	*	1	*	*	*	*	1	0	0	0	
0	1	0	0	*	*	*	*	0	*	0	1	0	0	4
0	1	0	0	*	*	*	*	1	*	0	0	1	0	
0	1	0	1	*	*	*	*	*	0	0	1	0	1	5
0	1	0	1	*	*	*	*	*	1	0	0	1	1	
0	1	1	0	*	*	*	*	*	0	0	1	1	0	6
0	1	1	0	*	*	*	*	*	1	0	0	1	1	
0	1	1	1	*	*	*	0	*	*	0	1	1	1	7
0	1	1	1	*	*	*	1	*	*	0	0	0	1	
1	0	0	0	*	*	*	*	0	*	1	0	0	0	8
1	0	0	0	*	*	*	*	1	*	0	0	1	0	
1	0	0	1	*	*	*	0	*	*	1	0	0	1	9
1	0	0	1	*	*	*	1	*	*	0	0	0	1	
0	0	0	1	*	0	0	*	*	*	0	0	0	1	1
0	0	1	0	0	*	0	*	*	*	0	0	1	0	2
0	0	1	1	0	0	*	*	*	*	0	0	1	1	3

Funkcje przejścia:

Dla X1(t+1) :

$$X_1(t+1) = \bar{I}_2 \bar{X}_2 X_3 X_4 \bar{I}_1 + \bar{X}_1 \bar{X}_2 X_3 X_4 \bar{I}_2 + X_1 \bar{X}_2 \bar{X}_3 \bar{X}_4 \bar{I}_6 + X_1 \bar{X}_2 \bar{X}_3 X_4 \bar{I}_5 + \bar{X}_1 \bar{X}_2 \bar{X}_3 \bar{X}_4 \bar{I}_7$$

Dla X2(t+1):

$$X_2(t+1) = \bar{X}_1 \bar{X}_2 \bar{X}_3 X_4 \bar{I}_2 + \bar{X}_1 \bar{X}_2 \bar{X}_3 X_4 \bar{I}_3 + \bar{X}_1 \bar{X}_2 X_3 \bar{X}_4 \bar{I}_1 + \bar{X}_1 \bar{X}_2 X_3 \bar{X}_4 \bar{I}_3 + \\ + \bar{X}_1 X_2 \bar{X}_3 \bar{X}_4 \bar{I}_6 + \bar{X}_1 X_2 \bar{X}_3 X_4 \bar{I}_7 + \bar{X}_1 X_2 X_3 \bar{X}_4 \bar{I}_7 + \bar{X}_1 X_2 X_3 X_4 \bar{I}_5 + \bar{X}_1 \bar{X}_2 \bar{X}_3 \bar{X}_4 \bar{I}_5$$

Dla X3(t+1):

$$X_3(t+1) = \bar{X}_1 \bar{X}_2 \bar{X}_3 \bar{X}_4 \bar{I}_6 + \bar{X}_1 \bar{X}_2 X_3 \bar{X}_4 \bar{I}_1 + \bar{X}_1 \bar{X}_2 X_3 \bar{X}_4 \bar{I}_3 + \bar{X}_1 X_2 \bar{X}_3 \bar{X}_4 \bar{I}_6 + \\ + \bar{X}_1 X_2 \bar{X}_3 X_4 \bar{I}_7 + \bar{X}_1 X_2 X_3 \bar{X}_4 \bar{I}_7 + \bar{X}_1 X_2 X_3 X_4 \bar{I}_5 + X_1 \bar{X}_2 \bar{X}_3 \bar{X}_4 \bar{I}_6 + \bar{X}_1 \bar{X}_2 X_3 \bar{X}_4 \bar{I}_1 \bar{I}_3 + \bar{X}_1 \bar{X}_2 X_3 X_4 \bar{I}_1 \bar{I}_2$$

Dla X4(t+1):

$$\begin{aligned} \cdot x_4(t+1) = & \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \bar{J}_3 + \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 \bar{J}_1 + \bar{x}_1 \bar{x}_2 x_3 x_4 \bar{J}_1 + \bar{x}_1 x_2 \bar{x}_3 x_4 \bar{J}_7 + \bar{x}_1 x_2 \bar{x}_3 x_4 \bar{J}_7 + \bar{x}_1 x_2 x_3 \bar{x}_4 \bar{J}_7 \\ & + \bar{x}_1 x_2 x_3 x_4 \bar{J}_5 + \bar{x}_1 x_2 x_3 x_4 \bar{J}_5 + x_1 \bar{x}_2 \bar{x}_3 x_4 \bar{J}_5 + x_1 \bar{x}_2 \bar{x}_3 x_4 \bar{J}_5 + \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \bar{J}_2 \bar{J}_3 + \bar{x}_1 \bar{x}_2 x_3 x_4 \bar{J}_1 \bar{J}_2 \end{aligned}$$

Tabela funkcja wyjścia:

Stan	X1	X2	X3	X4	FR(Q1)	S/S(Q2)
0	0	0	0	0	1	0
1	0	0	0	1	0	1
2	0	0	1	0	0	1
3	0	0	1	1	0	1
4	0	1	0	0	0	0
5	0	1	0	1	0	0
6	0	1	1	0	0	0
7	0	1	1	1	1	0
8	1	0	0	0	1	0
9	1	0	0	1	1	0

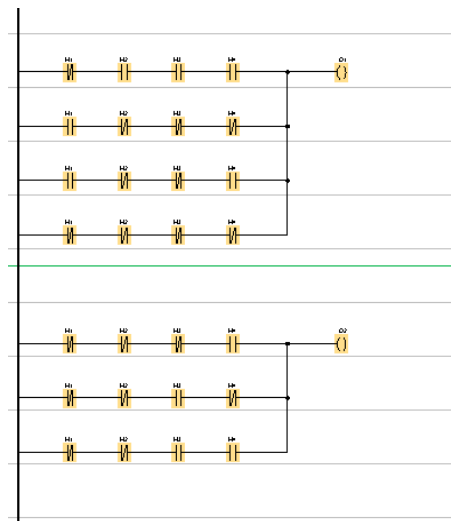
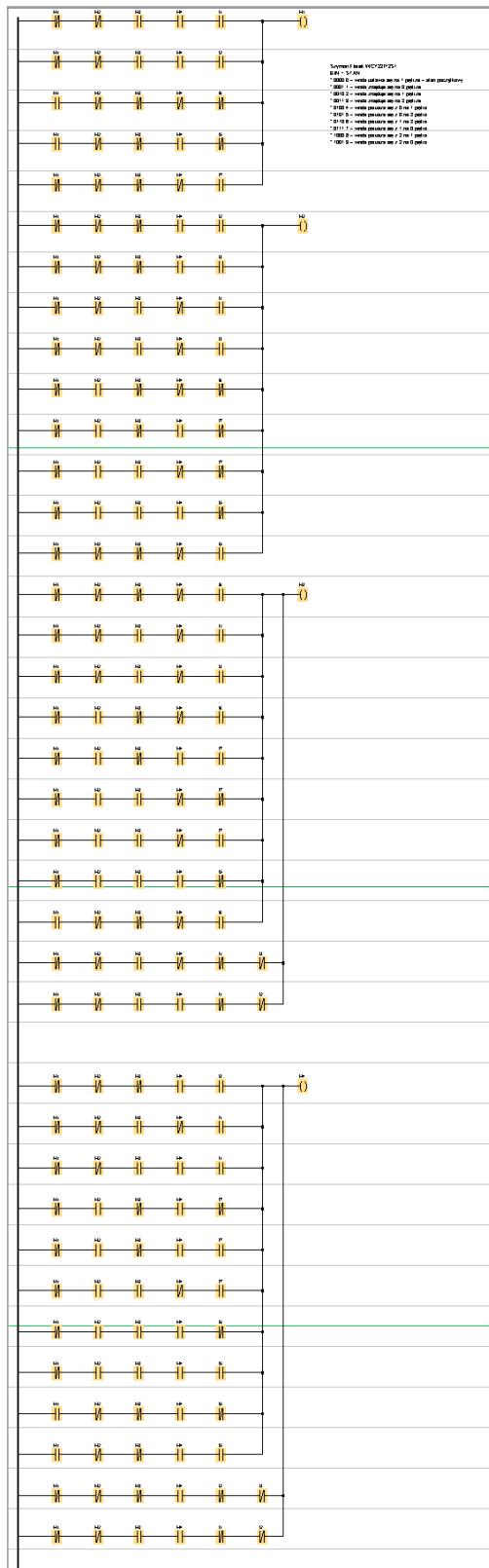
Dla Q1(FR):

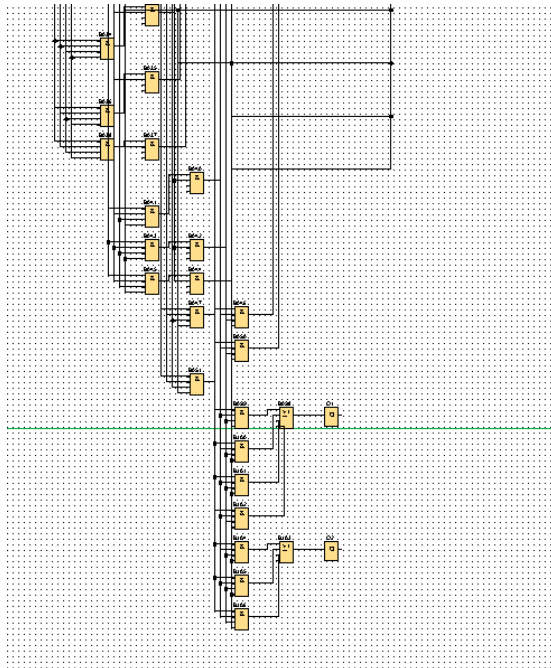
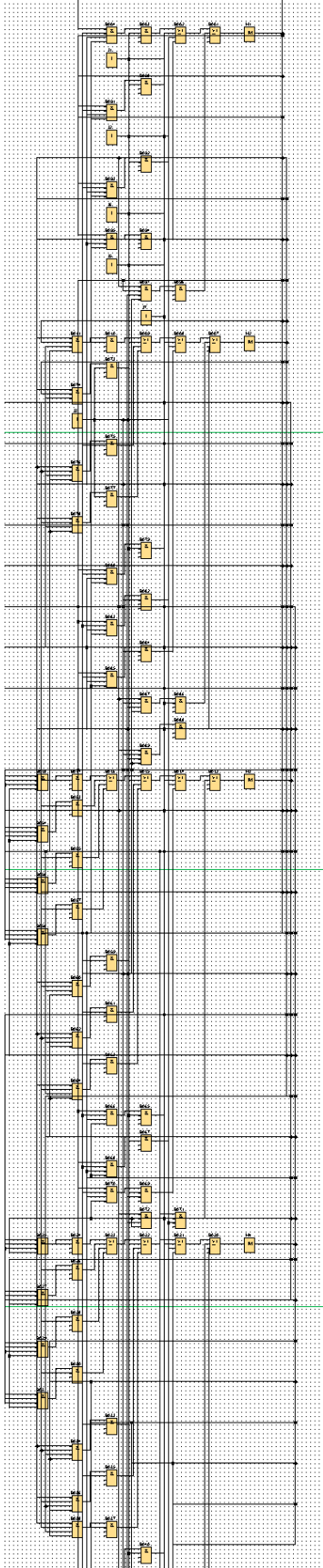
$$Q_1 = \bar{x}_1 x_2 x_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$$

Dla Q2(S/S):

$$Q_2 = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 x_3 x_4$$

Program LD i FBD:





Program arduino:

```
/* Szymon Florek WCY22IY2S1
* Winda pozusza sie miedzy kondygnacjami 0,1,2
* stan poczatkowy - winda ustawia sie na kondygnacji 1
*
* Opis stanów:
* 0000 0 – winda ustawia się na 1 piętrze – stan początkowy
* 0001 1 – winda znajduje się na 0 piętrze
* 0010 2 – winda znajduje się na 1 piętrze
* 0011 3 – winda znajduje się na 2 piętrze
* 0100 4 – winda porusza się z 0 na 1 piętro
* 0101 5 – winda porusza się z 0 na 2 piętro
* 0110 6 – winda porusza się z 1 na 2 piętro
* 0111 7 – winda porusza się z 1 na 0 piętro
* 1000 8 – winda porusza się z 2 na 1 piętro
* 1001 9 – winda porusza się z 2 na 0 piętro
*/
```

```
#define Button1Pin A0
#define Button2Pin A1
#define Button3Pin A2
#define Button4Pin A3
#define Sensor1Pin A4
#define Sensor2Pin A5
#define Sensor3Pin 12
#define Sensor4Pin 13
#define OutputQ1Pin 5
#define OutputQ2Pin 4
```

```
//variables to store button input
boolean button0 = 0;
boolean button1 = 0;
boolean button2 = 0;
```

```
//variables to store sensor input
boolean sensor0 = 0;
boolean sensor1 = 0;
```

```

boolean sensor2 = 0;
//variables to store state of state machine
boolean X1 = 0;
boolean X2 = 0;
boolean X3 = 0;
boolean X4 = 0;
//variables to store output for the motor
boolean Q1 = 0;
boolean Q2 = 0;

//function that reads input of sensors and buttons
void readInput(){
  button0 = digitalRead(Button1Pin);
  button1 = digitalRead(Button2Pin);
  button2 = digitalRead(Button3Pin);

  sensor0 = digitalRead(Sensor1Pin);
  sensor1 = digitalRead(Sensor2Pin);
  sensor2 = digitalRead(Sensor3Pin);
}

//setup function to set read/write modes for pins
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);
}

```

```

//function to calculate the next state;
void calculateState(){
boolean X1copy = 0;
boolean X2copy = 0;
boolean X3copy = 0;
boolean X4copy = 0;

//calculate next state based on previous state and input
X1copy = (!X1 && !X2 && X3 && X4 && button0)
|| (!X1 && !X2 && X3 && X4 && button1)
|| (X1 && !X2 && !X3 && !X4 && !sensor1)
|| (X1 && !X2 && !X3 && X4 && !sensor0);

X2copy = (!X1 && !X2 && !X3 && X4 && button1)
|| (!X1 && !X2 && !X3 && X4 && button2)
|| (!X1 && !X2 && X3 && !X4 && button0)
|| (!X1 && !X2 && X3 && !X4 && button2)
|| (!X1 && X2 && !X3 && !X4 && !sensor1)
|| (!X1 && X2 && !X3 && X4 && !sensor2)
|| (!X1 && X2 && X3 && !X4 && !sensor2)
|| (!X1 && X2 && X3 && X4 && !sensor0);

X3copy = (!X1 && !X2 && !X3 && !X4 && sensor1)
|| (!X1 && !X2 && X3 && !X4 && button0)
|| (!X1 && !X2 && X3 && !X4 && button2)
|| (!X1 && X2 && !X3 && !X4 && sensor1)
|| (!X1 && X2 && !X3 && X4 && sensor2)
|| (!X1 && X2 && X3 && !X4 && !sensor2)
|| (!X1 && X2 && X3 && !X4 && sensor2)
|| (!X1 && X2 && X3 && X4 && !sensor0)
|| (X1 && !X2 && !X3 && !X4 && sensor1)
|| (!X1 && !X2 && X3 && !X4 && !button0 && !button2)
|| (!X1 && !X2 && X3 && X4 && !button0 && !button1);

X4copy = (!X1 && !X2 && !X3 && X4 && sensor2)
|| (!X1 && !X2 && X3 && !X4 && button0)
|| (!X1 && !X2 && X3 && X4 && button0)

```

```

|| (!X1 && X2 && !X3 && X4 && !sensor2)
|| (!X1 && X2 && !X3 && X4 && sensor2)
|| (!X1 && X2 && X3 && !X4 && sensor2)
|| (!X1 && X2 && X3 && X4 && !sensor0)
|| (!X1 && X2 && X3 && X4 && sensor0)
|| (X1 && !X2 && !X3 && X4 && !sensor0)
|| (X1 && !X2 && !X3 && X4 && sensor0)
|| (!X1 && !X2 && !X3 && X4 && !button1 && !button2)
|| (!X1 && !X2 && X3 && X4 && !button0 && !button1);

```

```

//make next state current state

```

```

X1 = X1copy;
X2 = X2copy;
X3 = X3copy;
X4 = X4copy;
}

```

```

//function to calculate output for the elevator motor

```

```

void calculateOutput(){
Q1 = (!X1 && X2 && X3 && X4)
|| (X1 && !X2 && !X3 && !X4)
|| (X1 && !X2 && !X3 && X4)
|| (!X1 && !X2 && !X3 && !X4);

```

```

Q2 = (!X1 && !X2 && !X3 && X4)
|| (!X1 && !X2 && X3 && !X4)
|| (!X1 && !X2 && X3 && X4);
}

```

```

//function to set output onto the pins

```

```

void writeOutput(){
digitalWrite(OutputQ1Pin, Q1);
digitalWrite(OutputQ2Pin, Q2);
}

```

```
void report(){  
  Serial.println("Odczyt przyciskow: ");  
  Serial.print("przycisk 0:");  
  Serial.println(button0);  
  Serial.print("przycisk 1:");  
  Serial.println(button1);  
  Serial.print("przycisk 2:");  
  Serial.println(button2);
```

```
  
  Serial.println("Odczyt sensorow: ");  
  Serial.print("sensor 0:");  
  Serial.println(sensor0);  
  Serial.print("sensor 1:");  
  Serial.println(sensor1);  
  Serial.print("sensor 2:");  
  Serial.println(sensor2);
```

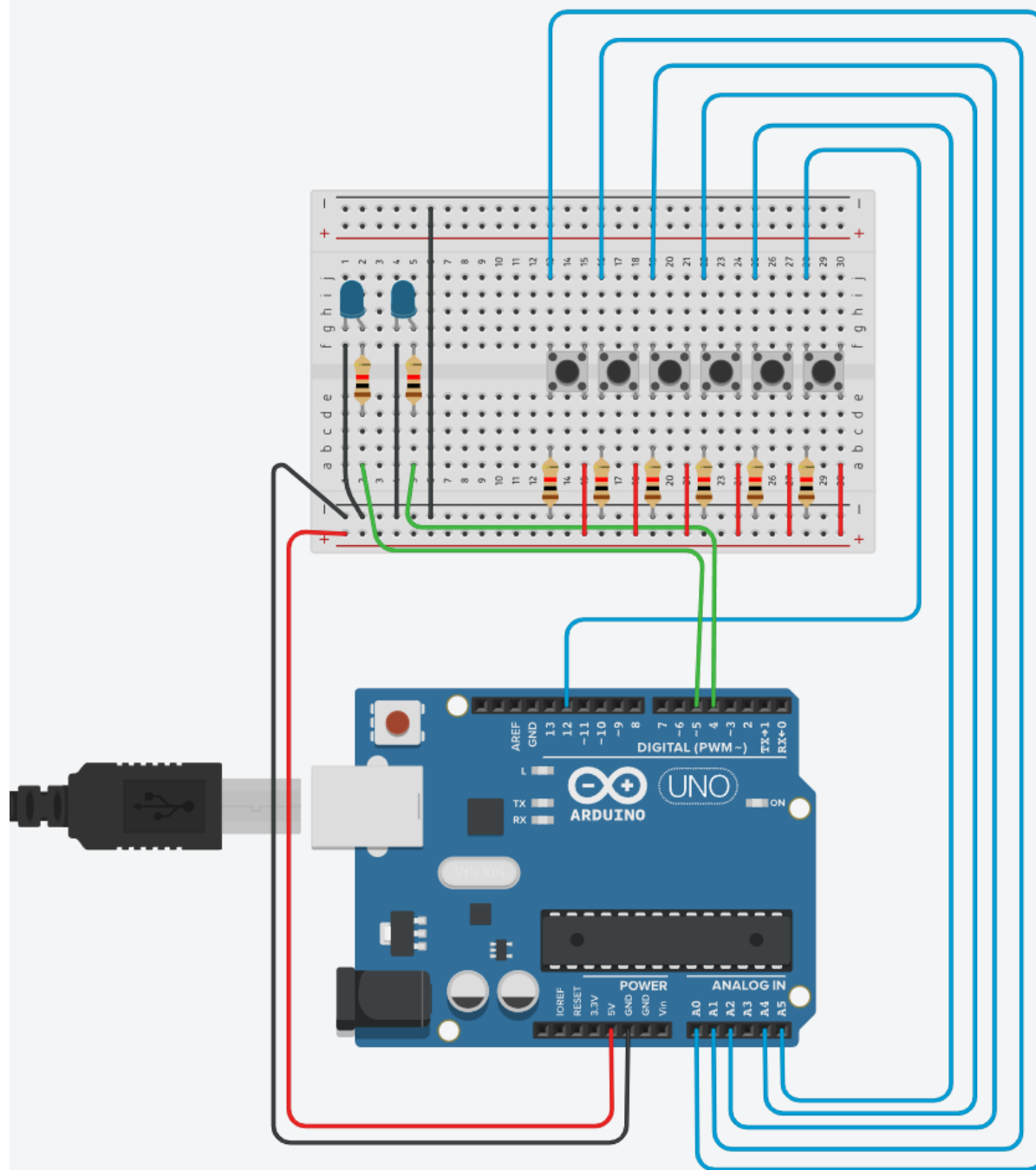
```
  
  Serial.println("Obliczony stan: ");  
  Serial.print(X1);  
  Serial.print(X2);  
  Serial.print(X3);  
  Serial.println(X4);
```

```
  
  Serial.println("Wyjscia (Q1Q2): ");  
  Serial.print(Q1);  
  Serial.println(Q2);  
}
```

```
void loop() {
```

```
  readInput();  
  calculateState();  
  calculateOutput();  
  writeOutput();  
  report();  
}
```

Układ na tkinder.com:



Wnioski:

Program PLC: Przeprowadzając symulację stwierdzono poprawność programu. Maszyna pod wpływem różnych wartości na przyciskach oraz sensorach przechodzi do odpowiednich stanów. Wyjścia sterujące silnik również mają odpowiednie wartości dla odpowiednich stanów.

Program Arduino:

```
Kompilacja zakończona
"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-ar" rcs "C:\Users\szym\AppData\Local\Temp\build86c1372d0
"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-ar" rcs "C:\Users\szym\AppData\Local\Temp\build86c1372d0
"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-gcc" -Os -Wl,--gc-sections -mmcu=atmega328p -o "C:\Users
"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-objcopy" -O ihex -j .eeprom --set-section-flags=.eeprom=al
"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-objcopy" -O ihex -R .eeprom "C:\Users\szym\AppData\Local

Szkic używa 4 248 bajtów z (13%) pamięci programu. Maksimum to 32 256 bajtów.
Globalne zmienne używaj? 350 bajtów z (17%) dynamicznej pamięci, pozostawiając 1 698 bajtów dla lokalnych zmiennych. M
```

Zbudowano układ na stronie tinkercad.com (układ na rzucie ekranu).

6 guzików reprezentuje kolejno:

I1 -> I2 -> I3 -> I5 -> I6 -> I7

Dwie niebieskie diody pokazują nam stany wyjścia dla zadanego ustawienia np:

Q1 (1) Q2 (0) dla stanu początkowego aby winda zaczęła poruszać się na początku w dół na 1 piętro.

DLA LD:

(zakładając podana na zajęciach informację, że winda hardwareowo na początku podczas ładowania programu jedzie na 3 piętro).