

Jakub Gubar nr albumu 29205

05.11.2024r.

Sprawozdanie

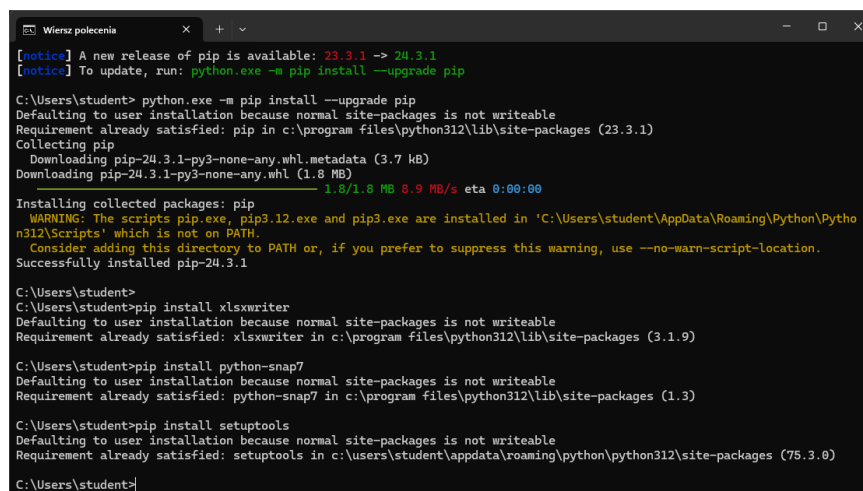
Internet rzeczy: STEROWNIK LOGO 8 12/24RCE – TRANSMISJA DANYCH ZE STEROWNIKA LOGO DO PC

1. Cel ćwiczenia - Celem ćwiczenia było zapoznanie z transmisją danych ze sterownika do PC za pomocą języka Python.

2. Przebieg ćwiczenia.

2.1. Monitorowanie tabeli stanów wejść i wyjść sterownika w czasie symulacji.

Jeszcze przed przystąpieniem do ćwiczenia, posługując się zamieszczonym wstępem teoretycznym, zainstalowaliśmy wymagane biblioteki do obsługi sterownika Logo.



```
Wiensz polecenia
[notice] A new release of pip is available: 23.3.1 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\student> python.exe -m pip install --upgrade pip
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pip in c:\program files\python312\lib\site-packages (23.3.1)
Collecting pip
  Downloading pip-24.3.1-py3-none-any.whl.metadata (3.7 kB)
  Downloading pip-24.3.1-py3-none-any.whl (1.8 MB)
    1.8/1.8 MB 8.9 MB/s eta 0:00:00
Installing collected packages: pip
  WARNING: The scripts pip.exe, pip3.12.exe and pip3.exe are installed in 'C:\Users\student\AppData\Roaming\Python\Python312\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pip-24.3.1

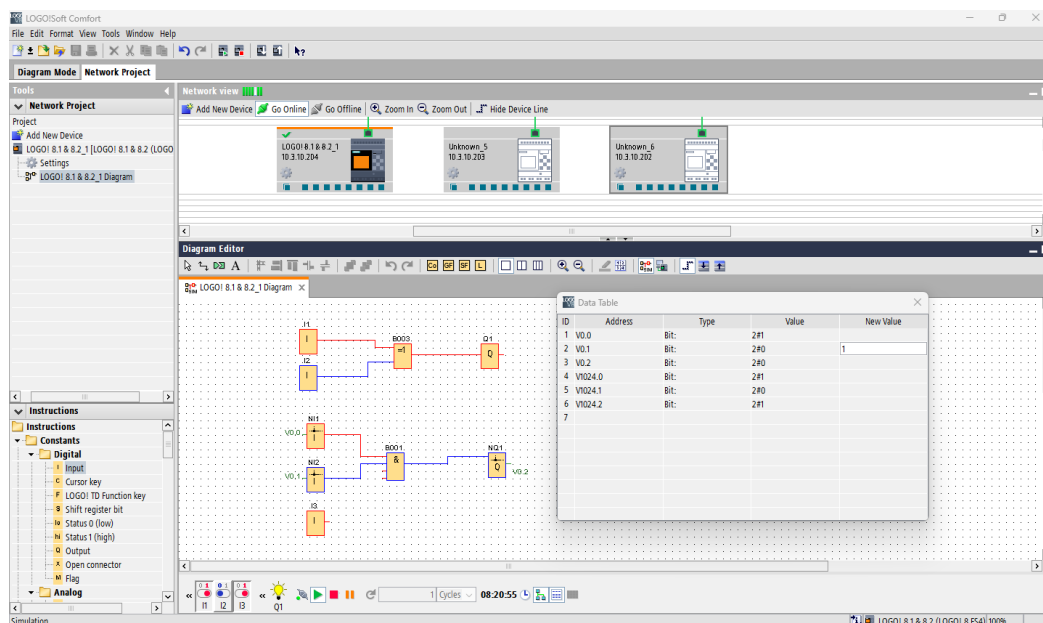
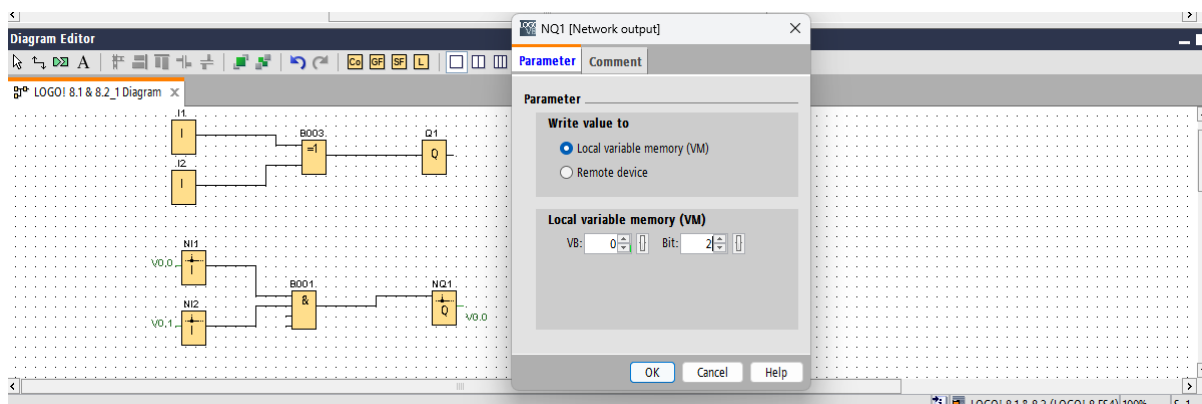
C:\Users\student>
C:\Users\student>pip install xlswriter
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: xlswriter in c:\program files\python312\lib\site-packages (3.1.9)

C:\Users\student>pip install python-snap7
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: python-snap7 in c:\program files\python312\lib\site-packages (1.3)

C:\Users\student>pip install setuptools
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: setuptools in c:\users\student\appdata\roaming\python\python312\site-packages (75.3.0)

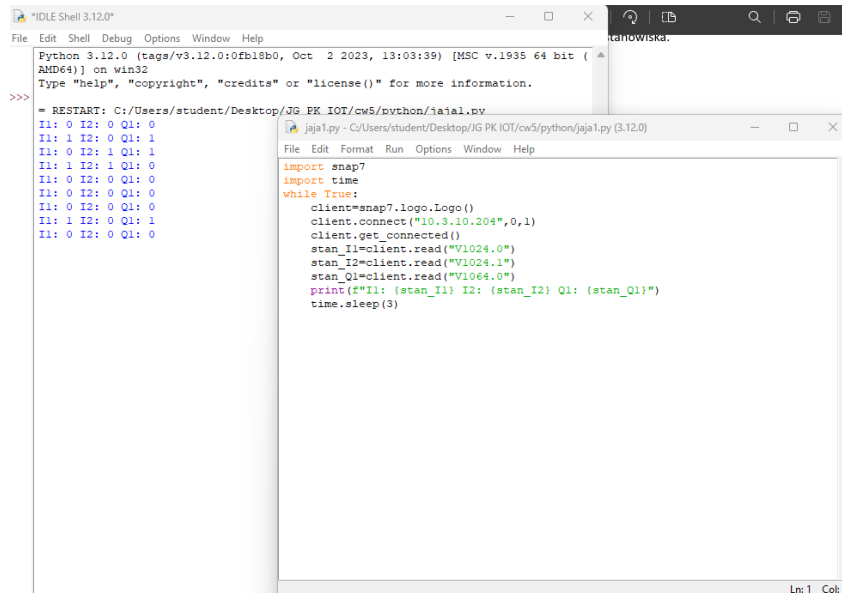
C:\Users\student>
```

Postępując zgodnie z instrukcją stworzyliśmy program, w którym zaadresowaliśmy wejścia i wyjścia sieciowe tak, aby odnosiły się one do pamięci wirtualnej VM.



2.2. Python, odczyt stanu wejść i wyjść cyfrowych.

Kolejno usunęliśmy wejście I3 dodane w poprzednim zadaniu, a następnie uruchomiliśmy edytor języka Python. Wpisaliśmy program monitorujący i wyświetlający odczytane wartości z wejścia I1, I2 oraz wyjścia Q1.



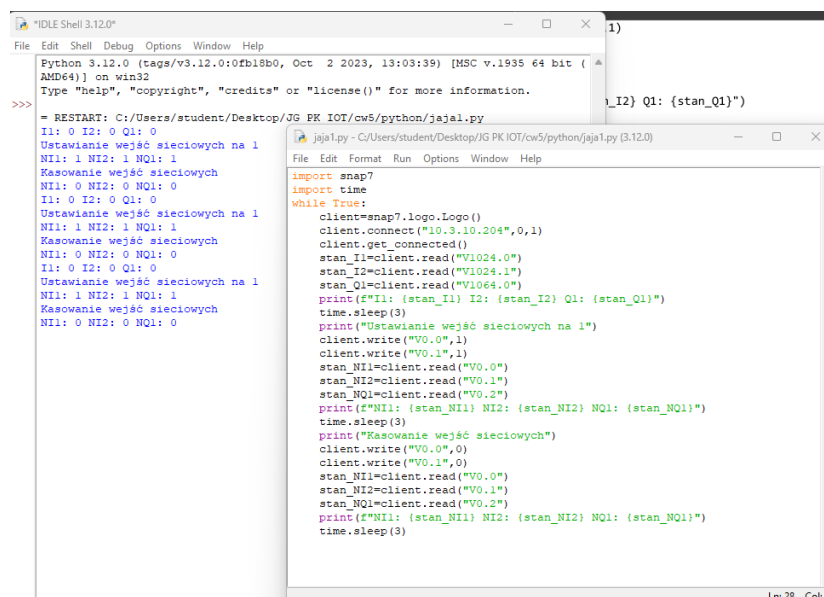
```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/student/Desktop/JG PK IOT/cw5/python/jajal.py
I1: 0 I2: 0 Q1: 0
I1: 1 I2: 0 Q1: 1
I1: 1 I2: 1 Q1: 1
I1: 0 I2: 1 Q1: 0
I1: 0 I2: 0 Q1: 0
I1: 0 I2: 0 Q1: 0
I1: 1 I2: 0 Q1: 1
I1: 0 I2: 0 Q1: 0
```

```
import snap7
import time
while True:
    client=snap7.logo.Logo()
    client.connect("10.3.10.204",0,1)
    client.get_connected()
    stan_I1=client.read("V1024.0")
    stan_I2=client.read("V1024.1")
    stan_Q1=client.read("V1064.0")
    print(f"I1: {stan_I1} I2: {stan_I2} Q1: {stan_Q1}")
    time.sleep(3)
```

Powyższy program wyświetlał co 3 sekundy stan rzeczywisty I1, I2 oraz Q1.

2.3. Odczyt, zapis i kasowanie stanów wejść sieciowych.

Rozbudowaliśmy poprzedni program o funkcję załączania stanu wysokiego na wejścia sieciowe NI1, NI2 oraz wyjścia sieciowego NQ1.

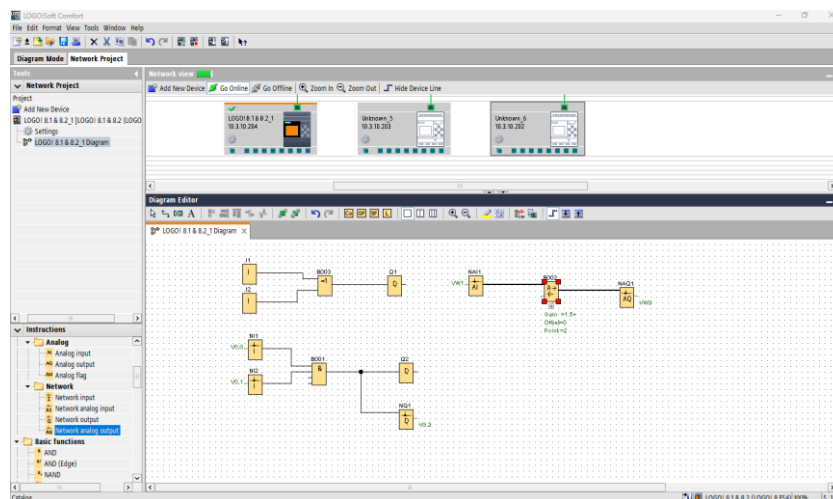


```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/student/Desktop/JG PK IOT/cw5/python/jajal.py
I1: 0 I2: 0 Q1: 0
Ustawianie wejść sieciowych na 1
NI1: 1 NI2: 1 NQ1: 1
Kasowanie wejść sieciowych
NI1: 0 NI2: 0 NQ1: 0
I1: 0 I2: 0 Q1: 0
Ustawianie wejść sieciowych na 1
NI1: 1 NI2: 1 NQ1: 1
Kasowanie wejść sieciowych
NI1: 0 NI2: 0 NQ1: 0
```

```
import snap7
import time
while True:
    client=snap7.logo.Logo()
    client.connect("10.3.10.204",0,1)
    client.get_connected()
    stan_I1=client.read("V1024.0")
    stan_I2=client.read("V1024.1")
    stan_Q1=client.read("V1064.0")
    print(f"I1: {stan_I1} I2: {stan_I2} Q1: {stan_Q1}")
    time.sleep(3)
    print("Ustawianie wejść sieciowych na 1")
    client.write("W0.0",1)
    client.write("W0.1",1)
    stan_NI1=client.read("V0.0")
    stan_NI2=client.read("V0.1")
    stan_NQ1=client.read("V0.2")
    print(f"NI1: {stan_NI1} NI2: {stan_NI2} NQ1: {stan_NQ1}")
    time.sleep(3)
    print("Kasowanie wejść sieciowych")
    client.write("W0.0",0)
    client.write("W0.1",0)
    stan_NI1=client.read("V0.0")
    stan_NI2=client.read("V0.1")
    stan_NQ1=client.read("V0.2")
    print(f"NI1: {stan_NI1} NI2: {stan_NI2} NQ1: {stan_NQ1}")
    time.sleep(3)
```

2.4. Odczyt i zapis wejść/wyjść sieciowych analogowych.

Rozbudowaliśmy układ wgrany do sterownika dodając wejście i wyjście sieciowe analogowe oraz wzmacniacz analogowy.



Parametry nowo dodanych bloków ustawiliśmy według podanych wytycznych.

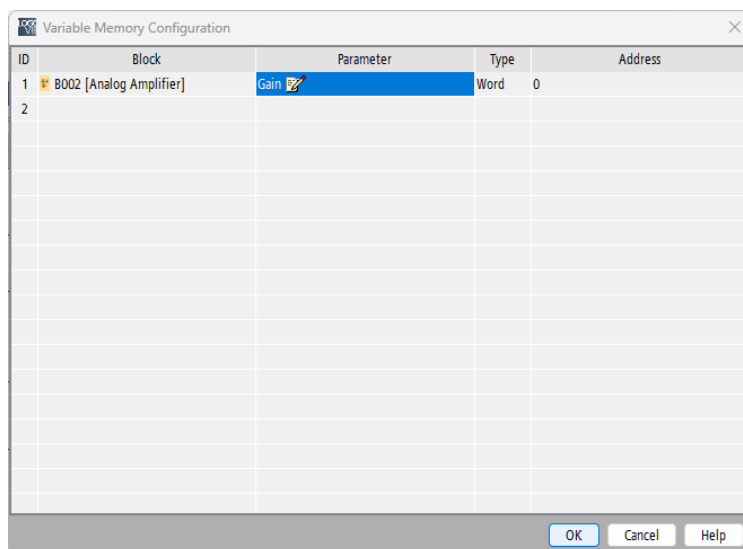
Następnie rozbudowaliśmy program w Pythonie w taki sposób, aby przy każdym kolejnym cyklu pętli „while” wartość na analogowym wejściu sieciowym była zwiększana o 5.

```
"IDLE Shell 3.12.0"
File Edit Shell Debug Options Window Help
Python 3.12.0 (tags/v3.12.0:0Zb19b0, Oct 2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/student/Desktop/JG PK IOT/cw5/python/jaja1.py
I1: 0 I2: 0 Q1: 0
Ustawianie wejść sieciowych na 1
N11: 1 N12: 1 NQ1: 1
Kasowanie wejść sieciowych
N11: 0 N12: 0 NQ1: 0
Ustawianie wejść i wyjść analogowych
N11: 5 N12: 8
N11: 0 N12: 0
I1: 0 I2: 0 Q1: 0
Ustawianie wejść sieciowych na 1
N11: 1 N12: 1 NQ1: 1
Kasowanie wejść sieciowych
N11: 0 N12: 0 NQ1: 0
Ustawianie wejść i wyjść analogowych
N11: 5 N12: 8
N11: 10 N12: 15
I1: 0 I2: 0 Q1: 0
Ustawianie wejść sieciowych na 1
N11: 1 N12: 1 NQ1: 1
Kasowanie wejść sieciowych
N11: 0 N12: 0 NQ1: 0
Ustawianie wejść i wyjść analogowych
N11: 15 N12: 23
I1: 0 I2: 0 Q1: 0
Ustawianie wejść sieciowych na 1
N11: 1 N12: 1 NQ1: 1
Kasowanie wejść sieciowych
N11: 0 N12: 0 NQ1: 0

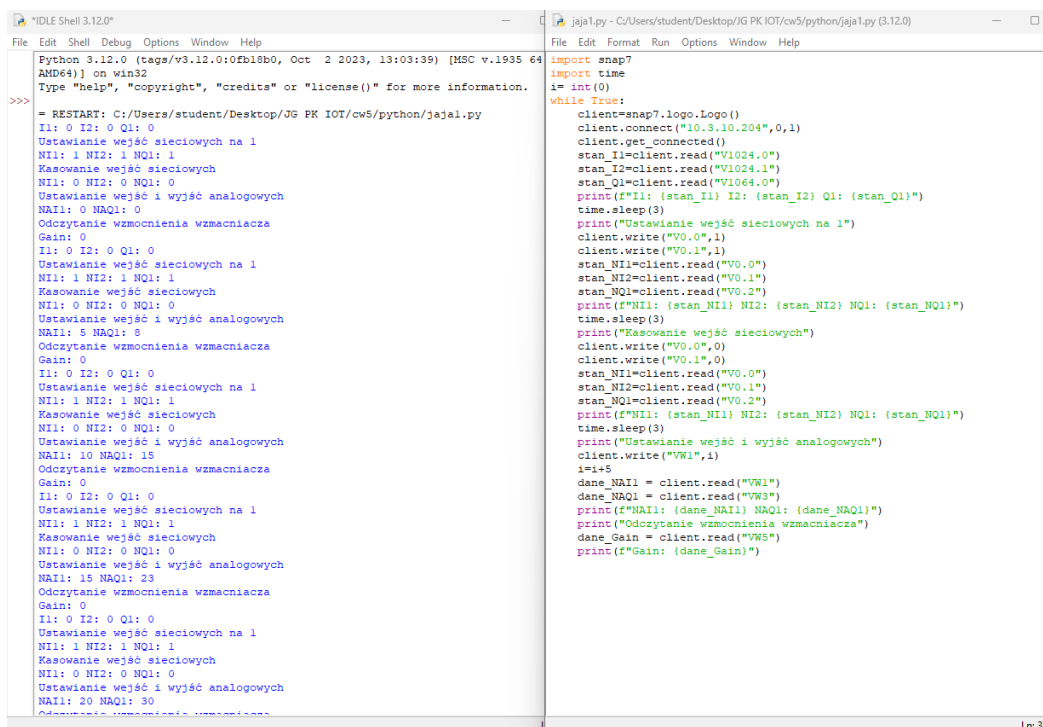
jaja1.py - C:/Users/student/Desktop/JG PK IOT/cw5/python/jaja1.py (3.12.0)
File Edit Shell Debug Options Window Help
import snap7
import time
from datetime import datetime
while True:
    client=snap7.logo.Logo()
    client.connect("10.3.10.204",0,1)
    client.get_connected()
    stan_I1=client.read("V1024.0")
    stan_I2=client.read("V1024.1")
    stan_Q1=client.read("V1064.0")
    print(f"I1: {stan_I1} I2: {stan_I2} Q1: {stan_Q1}")
    time.sleep(3)
    print("Ustawianie wejść sieciowych na 1")
    client.write("V0.0",1)
    client.write("V0.1",1)
    stan_N11=client.read("V0.0")
    stan_N12=client.read("V0.1")
    stan_NQ1=client.read("V0.2")
    print(f"N11: {stan_N11} N12: {stan_N12} NQ1: {stan_NQ1}")
    time.sleep(3)
    print("Kasowanie wejść sieciowych")
    client.write("V0.0",0)
    client.write("V0.1",0)
    stan_N11=client.read("V0.0")
    stan_N12=client.read("V0.1")
    stan_NQ1=client.read("V0.2")
    print(f"N11: {stan_N11} N12: {stan_N12} NQ1: {stan_NQ1}")
    time.sleep(3)
    print("Ustawianie wejść i wyjść analogowych")
    client.write("VW1",1)
    i=i+5
    dane_N11 = client.read("VW1")
    dane_NA1 = client.read("VW3")
    print(f"N11: {dane_N11} NA1: {dane_NA1}")
```

2.5. Odczyt parametrów z bloków programu wpisanego do pamięci sterownika.

W kolejnym punkcie dodaliśmy kod odczytujący wartość wzmacnienia ustawioną we wzmacniaczu. Dla parametru „Gain” przypisaliśmy adres bajtu nr 5.



W Pythonie dopisaliśmy kod służący do odczytu wartości wzmacnienia wzmacniacza z pamięci sterownika. Poniżej kod oraz zrzut z działającego programu.



2.6. Rejestracja danych ze sterownika w MS Excel – zadanie do samodzielnego wykonania.

W Pythonie napisaliśmy program według wytycznych, który odczytywał stany I1, I2, Q1, NI1, NI2, Q2, NAI1, NAQ1 oraz zapisywał je w arkuszu MS Excel w kolejnych wierszach. Udało się osiągnąć szukany rezultat, ale wystąpił problem z realizacją odczytu daty i czasu ze sterownika.

```
IAJADRAMAT2.py - C:/Users/student/Desktop/JG PK IOT/cw5/python/IAJADRAMAT2.py (3.12.0)
File Edit Format Run Options Window Help
worksheet.write("F1","Q2")
worksheet.write("G1","NAI1")
worksheet.write("H1","NAQ1")
worksheet.write("I1","ROK")
worksheet.write("J1","MIESIĄC")
worksheet.write("K1","DZIEŃ")
worksheet.write("L1","GODZINY")
worksheet.write("M1","MINUTY")
worksheet.write("N1","SEKUNDY")
try:
    while True:
        data=client.read("V1024.0")
        worksheet.write(row,0,data)
        data=client.read("V1024.1")
        worksheet.write(row,1,data)
        data=client.read("V1064.0")
        worksheet.write(row,2,data)
        data=client.read("V0.0")
        worksheet.write(row,3,data)
        data=client.read("V0.1")
        worksheet.write(row,4,data)
        data=client.read("V1064.1")
        worksheet.write(row,5,data)
        data=client.read("VW1")
        worksheet.write(row,6,data)
        data=client.read("VW2")
        worksheet.write(row,7,data)
        data=client.read("B985")
        worksheet.write(row,8,data)
        data=client.read("V986.0")
        worksheet.write(row,9,data)
        data=client.read("V987.0")
        worksheet.write(row,10,data)
        data=client.read("V988.0")
        worksheet.write(row,11,data)
        data=client.read("V989.0")
        worksheet.write(row,12,data)
        data=client.read("V990.0")
        worksheet.write(row,13,data)
        row = row +1
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	I1	I2	Q1	NI1	NI2	Q2	NAI1	NAQ1	ROK	MIESIĄC	DZIEŃ	GODZINY	MINUTY	SEKUNDY
2		0	0	0	0	0	0	35	53	0	0	0	0	0
3		0	0	0	0	0	0	35	53	0	0	0	0	1
4		0	0	0	0	0	0	35	53	0	0	0	0	0
5		0	0	0	0	0	0	35	53	0	0	0	0	1
6														

3. Wnioski

- Dzięki zastosowaniu bibliotek Snap7 oraz XlsxWriter, możliwe było nawiązanie komunikacji między sterownikiem Logo a komputerem, co pozwala na monitorowanie i kontrolowanie stanów wejść i wyjść sterownika oraz zapisywanie tych danych w arkuszu kalkulacyjnym.
- Ćwiczenie obejmowało przypisywanie adresów do różnych typów danych w pamięci sterownika (np. wejścia cyfrowe, wejścia i wyjścia sieciowe), co pozwala na bardziej precyzyjną kontrolę nad danymi.
- Zadania polegające na implementacji nowych funkcji, takich jak kasowanie stanów wejść czy obsługa wejść/wyjść analogowych, umożliwiają praktyczne zrozumienie, jak sterownik reaguje na zmienne wartości i jak można go dostosować do potrzeb konkretnej aplikacji.