

Sprawozdanie z laboratorium 5

IO-LINK (S7-1200)

Łukasz Janusz
Marek Generowicz
13.03.2025



AGH

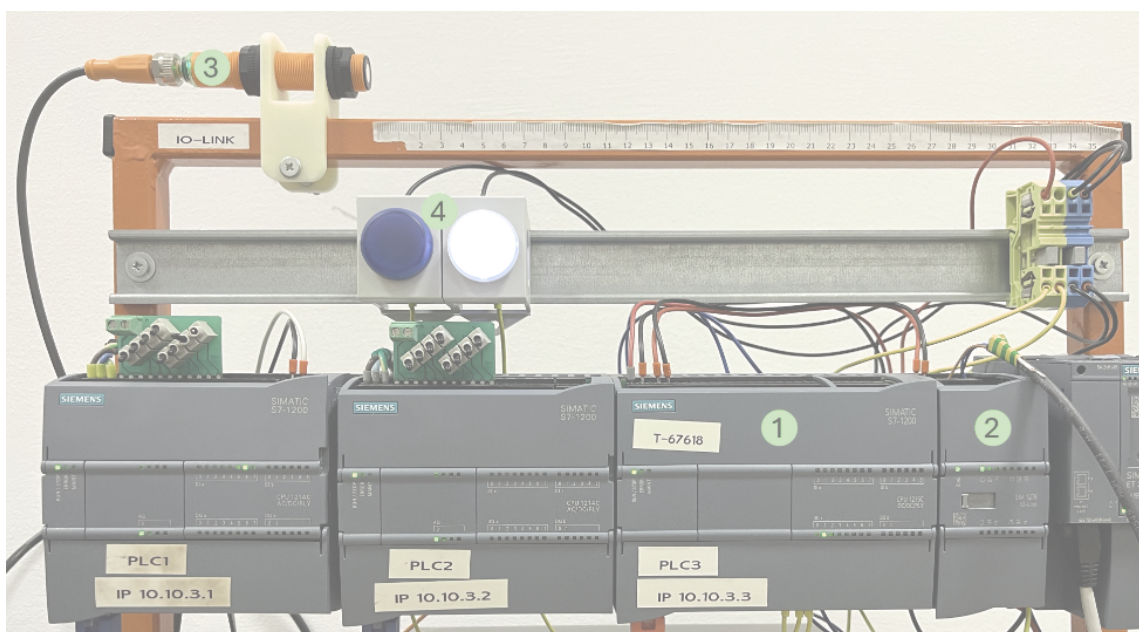
AGH UNIVERSITY OF KRAKOW

1 Wstęp

Na laboratoriach należało zapoznać się z cyfrowym, szeregowym protokołem komunikacyjnym IO-Link oraz jego konfiguracją w sterowniku S7-1200. Protokół ten jest szeroko używany w przemyśle do komunikacji sterowników z urządzeniami peryferyjnymi. IO-Link jest systemem typu point-to-point, co oznacza że każde urządzenie jest połączone bezpośrednio z masterem. IO-Link pozwala na łączenie się z urządzeniami analogowymi i cyfrowymi, a także może zostać użyty jako zasilanie. Ze względu na swoją specyfikę, stosowany jest głównie w warunkach lokalnych.

2 Opis Stanowiska

Na stanowisku laboratoryjnym (Zdjęcie 1) znajdował się sterownik S7-1200 marki Siemens wyposażony w czterokanałowy moduł IO-Link. Do sterownika podłączono ultradźwiękowy czujnik odległości UGT205. Zestaw uzupełniały kontrolki LED w kolorze niebieskim i białym.



Rysunek 1: Stanowisko laboratoryjne (zdjęcie z instrukcji)

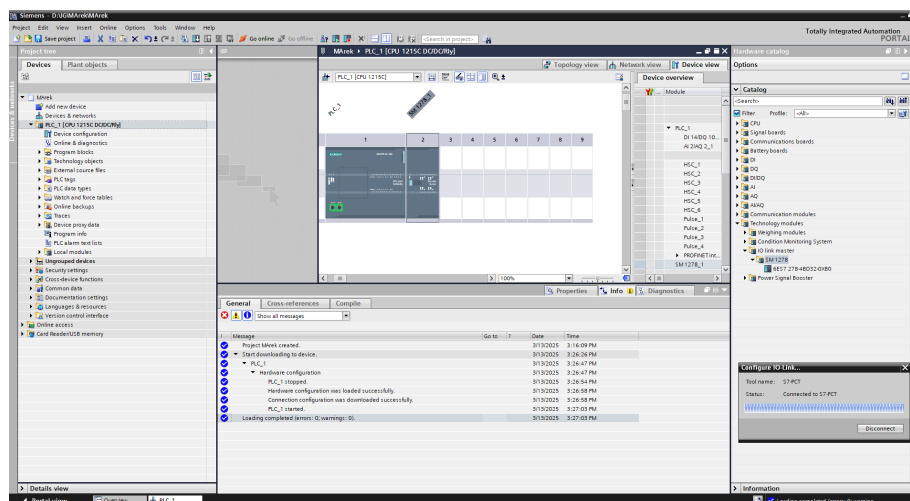
3 Przebieg ćwiczenia

Aby wykonać ćwiczenie należało wykonać następujące kroki:

- konfiguracją sterownika
- programowanie S7-PCT (Port Configuration Tool)
- odczyt danych z PLC

3.1 Konfiguracja sterownika

Po poprawnym skonfigurowaniu CPU oraz M 1278 4xIO-Link master z programem TIA Portal V19, otrzymywało się wirtualny odpowiednik stanowiska laboratoryjnego. Wygląd w programie TIA Portal przedstawiono na zdjęciu 2.

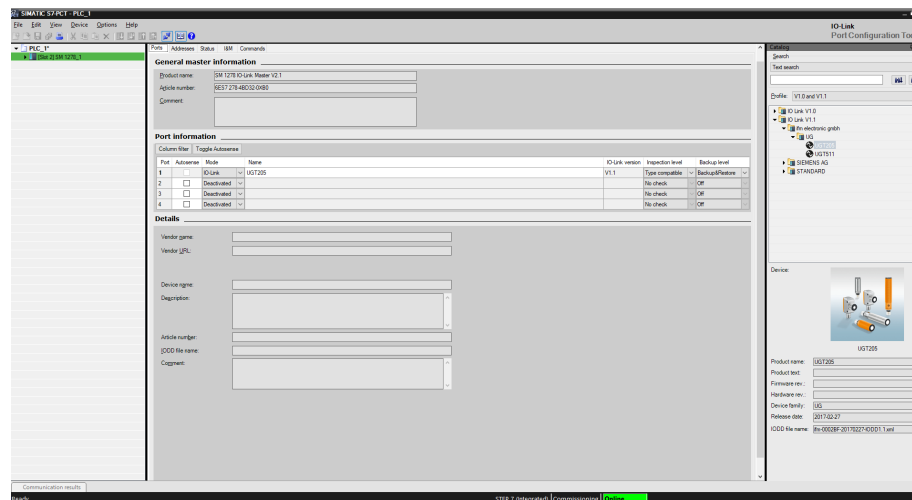


Rysunek 2: Schemat stanowiska w programie TIA Portal

Aby dokończyć konfigurację należało przeprowadzić identyfikację sprzętową modułu dla portu pierwszego oraz przesłać konfigurację do sterownika. Po zakończeniu konfiguracji należało przejść do programowania S7-PCT.

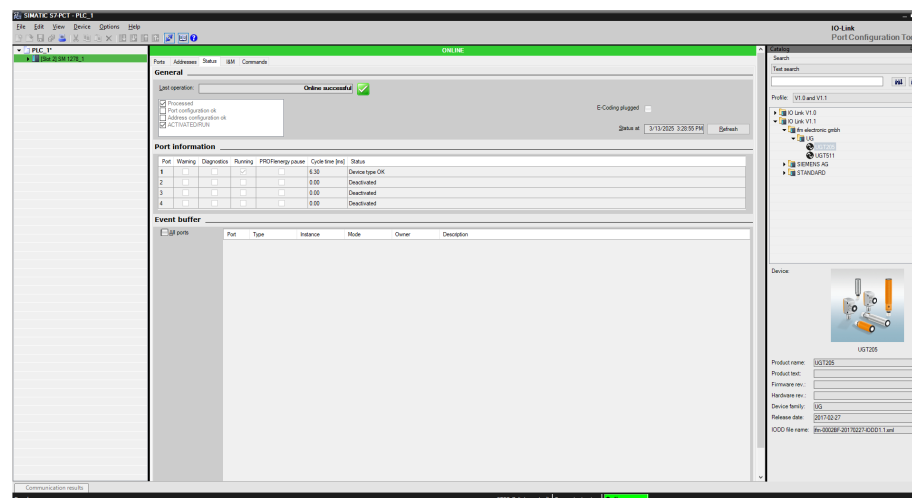
3.2 Programowanie S7-PCT

Aby rozpocząć należało kliknąć moduł IO-LINK prawym przyciskiem myszy a następnie wybrać *Start Device Tool*. Otworzy się wtedy okno przedstawione na zdjęciu 3



Rysunek 3: Okno Device Tool

Następnie należało przejść do zakładki status i połączyć się z urządzeniem wynikiem czego powinno wyświetlić się kno jak na zdjęciu 4.

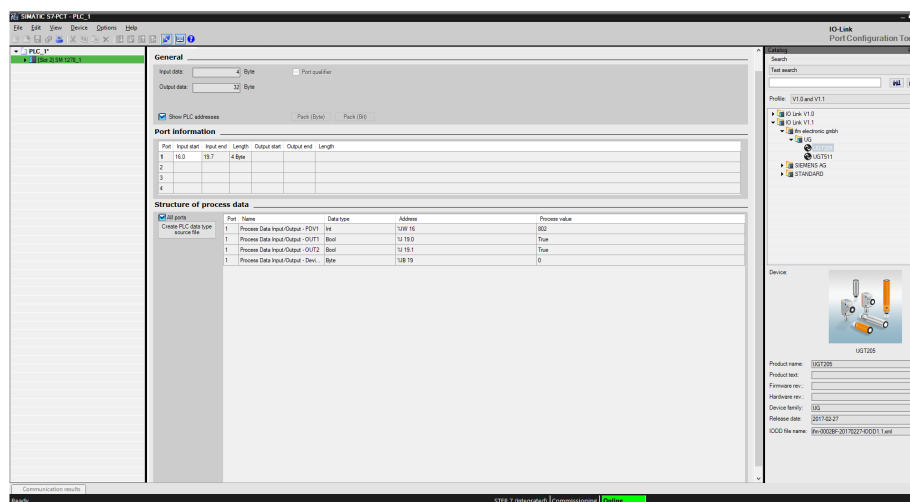


Rysunek 4: Okno statusu po pomyślnym nawiązaniu połączenia.

W następnej kolejności należało przejść do zakładki *Adresses*, a w nim zaznaczyć *Show PLC addresses* oraz *All ports*. Umożliwia to odczyt adresów zmiennych procesowych urządzenia w PLC. Na rysunku 5.

W programie dostajemy cztery outputy, które oznaczają odpowiednio:

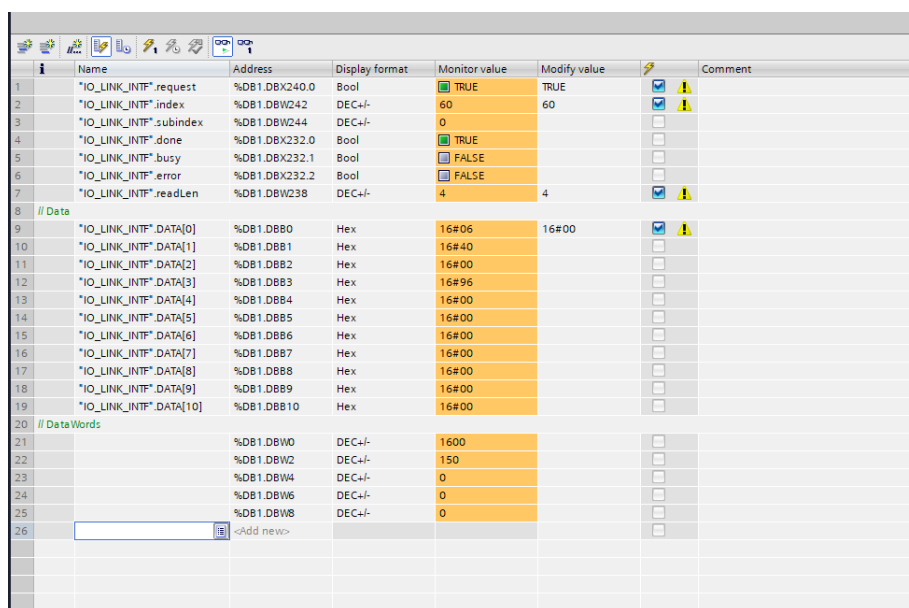
- **PDV 1(int)** - odległość od czujnika w milimetrach
- **OUT 1(bool)** - status *Swiching Signal Channel 1*
- **OUT 2(bool)** - status *Output Configuration 2*
- **Device Status(byte)** - bajt statusowy



Rysunek 5: Okno adresów

4 Odczyt danych z PLC

Po poprawnym skonfigurowaniu i podłączeniu sprzętu, można było przystąpić do odczytu danych z PLC. W tym celu należało uruchomić program TIA Portal i dodać nową *Tag table* o nazwie *UGT205*. Do podglądu danych z czujnika posłuży *Watch table*. Dzięki temu można było obserwować zmiany wartości parametrów w czasie rzeczywistym oraz modyfikować je w trakcie działania programu.

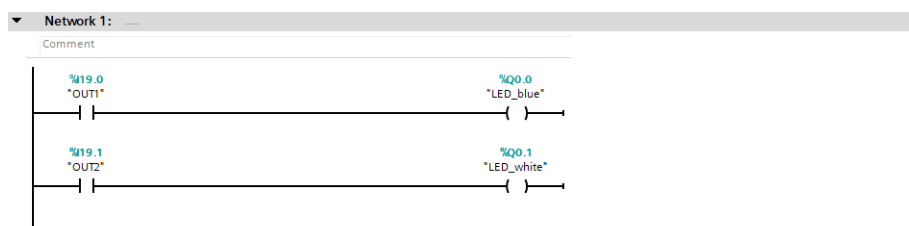


	Name	Address	Display format	Monitor value	Modify value		Comment
1	"IO_LINK_INTF".request	%DB1.DBX240.0	Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
2	"IO_LINK_INTF".index	%DB1.DBW242	DEC+/-	60	60	<input checked="" type="checkbox"/>	
3	"IO_LINK_INTF".subindex	%DB1.DBW244	DEC+/-	0		<input checked="" type="checkbox"/>	
4	"IO_LINK_INTF".done	%DB1.DBX232.0	Bool	TRUE		<input type="checkbox"/>	
5	"IO_LINK_INTF".busy	%DB1.DBX232.1	Bool	FALSE		<input type="checkbox"/>	
6	"IO_LINK_INTF".error	%DB1.DBX232.2	Bool	FALSE		<input type="checkbox"/>	
7	"IO_LINK_INTF".readLen	%DB1.DBW238	DEC+/-	4	4	<input checked="" type="checkbox"/>	
8	// Data						
9	"IO_LINK_INTF".DATA[0]	%DB1.DBB0	Hex	16#06	16#00	<input checked="" type="checkbox"/>	
10	"IO_LINK_INTF".DATA[1]	%DB1.DBB1	Hex	16#40		<input type="checkbox"/>	
11	"IO_LINK_INTF".DATA[2]	%DB1.DBB2	Hex	16#00		<input type="checkbox"/>	
12	"IO_LINK_INTF".DATA[3]	%DB1.DBB3	Hex	16#96		<input type="checkbox"/>	
13	"IO_LINK_INTF".DATA[4]	%DB1.DBB4	Hex	16#00		<input type="checkbox"/>	
14	"IO_LINK_INTF".DATA[5]	%DB1.DBB5	Hex	16#00		<input type="checkbox"/>	
15	"IO_LINK_INTF".DATA[6]	%DB1.DBB6	Hex	16#00		<input type="checkbox"/>	
16	"IO_LINK_INTF".DATA[7]	%DB1.DBB7	Hex	16#00		<input type="checkbox"/>	
17	"IO_LINK_INTF".DATA[8]	%DB1.DBB8	Hex	16#00		<input type="checkbox"/>	
18	"IO_LINK_INTF".DATA[9]	%DB1.DBB9	Hex	16#00		<input type="checkbox"/>	
19	"IO_LINK_INTF".DATA[10]	%DB1.DBB10	Hex	16#00		<input type="checkbox"/>	
20	// DataWords						
21		%DB1.DBW0	DEC+/-	1600		<input type="checkbox"/>	
22		%DB1.DBW2	DEC+/-	150		<input type="checkbox"/>	
23		%DB1.DBW4	DEC+/-	0		<input type="checkbox"/>	
24		%DB1.DBW6	DEC+/-	0		<input type="checkbox"/>	
25		%DB1.DBW8	DEC+/-	0		<input type="checkbox"/>	
26		<Add new>				<input type="checkbox"/>	

Rysunek 6: Utworzone watch table

Następnym krokiem było kontrolne sprawdzenie działania układu. Przekonał się, że czujnik nie jest w stanie poprawnie odczytać wartości mniejszej niż 10 cm. Wartość ta była zbyt mała dla czujnika ultradźwiękowego.

Później, poprzez dwie drabinki, nadaliśmy sterownikowi PLC poniższą logikę:

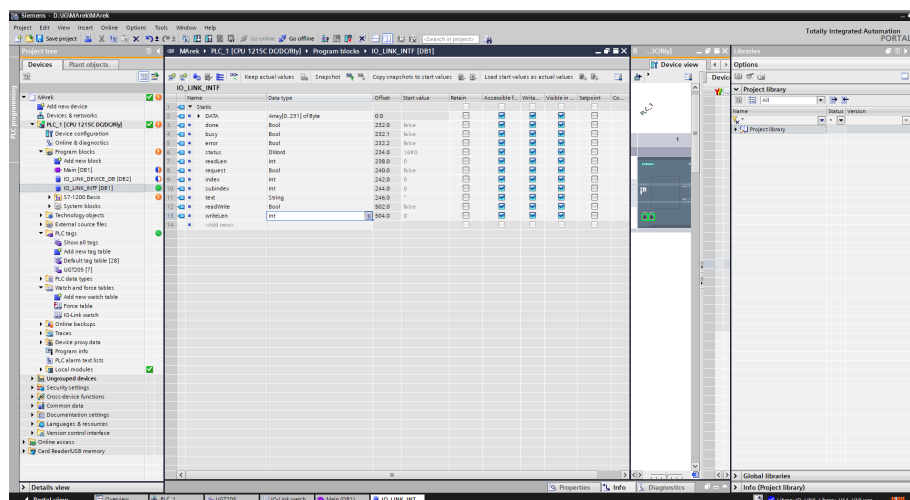


Rysunek 7: Logika sterownika PLC

W ten sposób wyjście z czujnika *OUT 1* zaświeca kontrolkę niebieską, a *OUT 2* kontrolkę białą.

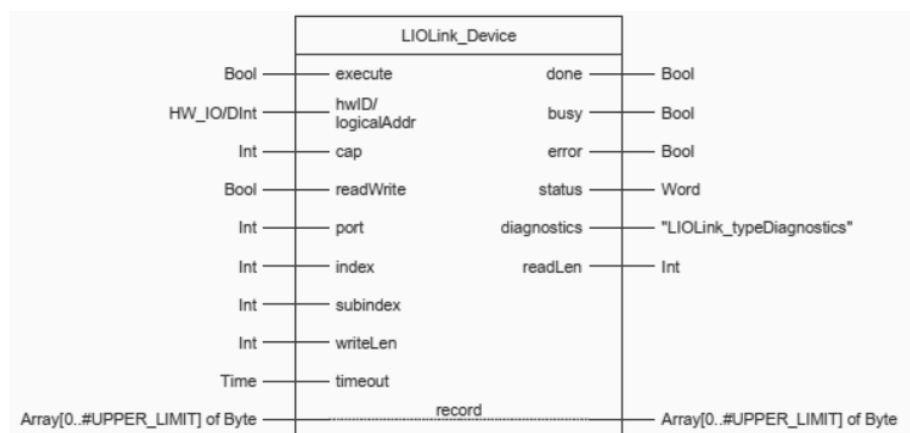
4.1 Odczyt danych na żądanie

Aby umożliwić odczyt danych na żądanie, należało posłużyć się data blokiem. Stworzony przez nas został nazwany *IO_LINK_DEVICE_DB*. Został w nim wyłączony atrybut *Optimized block access*, a zdefiniowane w nim tagi zostały przypisane do odpowiednich adresów zmiennych procesowych urządzenia.



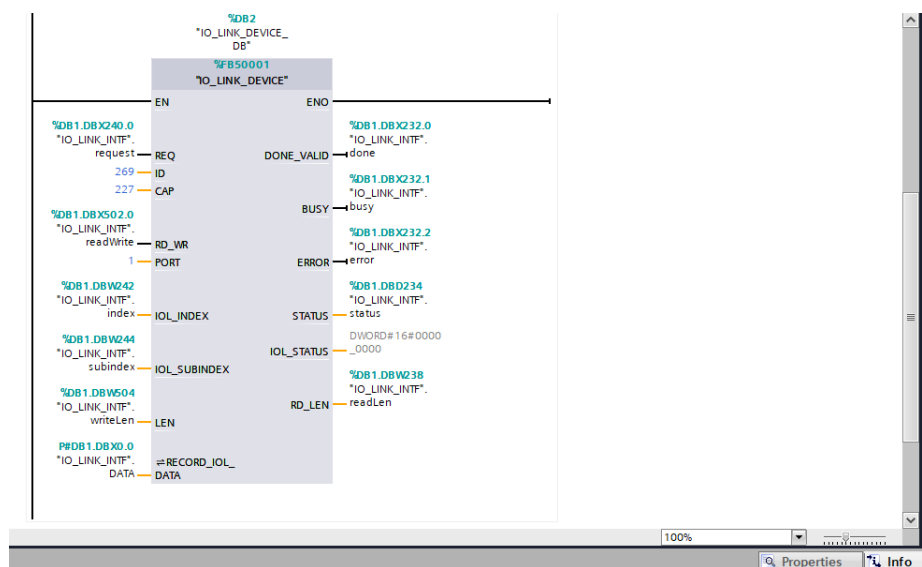
Rysunek 8: Data block IO_LINK_DEVICE_DB

Tagi, zdefiniowane w powyższy sposób są zgodne z dokumentacją biblioteki IO_Link.



Rysunek 9: Fragment dokumentacji biblioteki IO_Link

Kolejnym krokiem było dodanie bloku IO_LINK_DEVICE do sekcji *Global libraries* w programie TIA Portal. Następnie należało dodać blok do programu i przypisać go do odpowiedniego urządzenia IO-Link.



Rysunek 10: Blok dodany do programu

Parametry, które podczas działania programu nie będą ulegały zmianom zostały zdefiniowane jako *constant*. W ten sposób zabezpieczamy się przed przypadkową zmianą wartości. Każde z pozostałych parametrów zostały podpięte do uprzednio przygotowanych tagów. Tak przygotowany program został przesłany do sterownika.

Zgodnie z dokumentacją, parametry SP1 i SP2 powinny być ustawione na fabryczne wartości, czyli kolejno 1600 [mm] i 150 [mm]. Możemy to zaobserwować patrząc w sekcję DATA Words na wiersze %DB1.DBW0 i %DB1.DBW2 Rysunek 6.

4.2 Zapis danych na żądanie

W celu sprawdzenia działania zapisu danych na żądanie zmieniliśmy parametry SP1 i SP2 na wartości 300 [mm] oraz 150 [mm].

Następnie dodaliśmy do naszej *Watch table* pola *readWrite* oraz *writeLen* oraz zmieniliśmy wartość *readWrite* na *True* co pozwoliło nam na zapis wartości do tablicy paramterów.

"IO_LINK_INTF".request	%DB1.DBX240.0	Bool	<input checked="" type="checkbox"/> TRUE	TRUE	<input checked="" type="checkbox"/>
"IO_LINK_INTF".index	%DB1.DBW242	DEC+/-	60	60	<input checked="" type="checkbox"/>
"IO_LINK_INTF".subindex	%DB1.DBW244	DEC+/-	0		<input checked="" type="checkbox"/>
"IO_LINK_INTF".readWrite	%DB1.DBX502.0	Bool	<input type="checkbox"/> FALSE	TRUE	<input checked="" type="checkbox"/>
"IO_LINK_INTF".writeLen	%DB1.DBW504	DEC+/-	0	4	<input checked="" type="checkbox"/>
"IO_LINK_INTF".done	%DB1.DBX232.0	Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
"IO_LINK_INTF".busy	%DB1.DBX232.1	Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
"IO_LINK_INTF".error	%DB1.DBX232.2	Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
"IO_LINK_INTF".readLen	%DB1.DBW238	DEC+/-	4		<input type="checkbox"/>

Rysunek 11: Dodane pola do watch table

Jednocześnie ustawiliśmy wartość *writeLen* na 4, aby zapis odbywał się na czterech bitach. Po

wygenrowaniu rosnącego zbocza na zmiennej request, przeszliśmy do trybu odczytu i zobaczyliśmy nasze parametry w sekcji *DATA Words* Rysunek 12.

	Name	Address	Display format	Monitor value	Modify value		Comment
1	"IO_LINK_INTF".request	%DB1.DBX240.0	Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
2	"IO_LINK_INTF".index	%DB1.DBW242	DEC+/-	60	60	<input checked="" type="checkbox"/>	
3	"IO_LINK_INTF".subindex	%DB1.DBW244	DEC+/-	0		<input type="checkbox"/>	
4	"IO_LINK_INTF".done	%DB1.DBX232.0	Bool	TRUE		<input type="checkbox"/>	
5	"IO_LINK_INTF".busy	%DB1.DBX232.1	Bool	FALSE		<input type="checkbox"/>	
6	"IO_LINK_INTF".error	%DB1.DBX232.2	Bool	FALSE		<input type="checkbox"/>	
7	"IO_LINK_INTF".readLen	%DB1.DBW238	DEC+/-	4	4	<input checked="" type="checkbox"/>	
8	// Data						
9	"IO_LINK_INTF".DATA[0]	%DB1.DB80	Hex	16#01	16#01	<input checked="" type="checkbox"/>	
10	"IO_LINK_INTF".DATA[1]	%DB1.DB81	Hex	16#2C	16#2C	<input checked="" type="checkbox"/>	
11	"IO_LINK_INTF".DATA[2]	%DB1.DB82	Hex	16#00	16#00	<input checked="" type="checkbox"/>	
12	"IO_LINK_INTF".DATA[3]	%DB1.DB83	Hex	16#96	16#96	<input checked="" type="checkbox"/>	
13	"IO_LINK_INTF".DATA[4]	%DB1.DB84	Hex	16#00		<input type="checkbox"/>	
14	"IO_LINK_INTF".DATA[5]	%DB1.DB85	Hex	16#00		<input type="checkbox"/>	
15	"IO_LINK_INTF".DATA[6]	%DB1.DB86	Hex	16#00		<input type="checkbox"/>	
16	"IO_LINK_INTF".DATA[7]	%DB1.DB87	Hex	16#00		<input type="checkbox"/>	
17	"IO_LINK_INTF".DATA[8]	%DB1.DB88	Hex	16#00		<input type="checkbox"/>	
18	"IO_LINK_INTF".DATA[9]	%DB1.DB89	Hex	16#00		<input type="checkbox"/>	
19	"IO_LINK_INTF".DATA[10]	%DB1.DB810	Hex	16#00		<input type="checkbox"/>	
20	// DataWords						
21		%DB1.DBW0	DEC+/-	300		<input type="checkbox"/>	
22		%DB1.DBW2	DEC+/-	150		<input type="checkbox"/>	
23		%DB1.DBW4	DEC+/-	0		<input type="checkbox"/>	
24		%DB1.DBW6	DEC+/-	0		<input type="checkbox"/>	
25		%DB1.DBW8	DEC+/-	0		<input type="checkbox"/>	
26		<Add new>				<input type="checkbox"/>	

Rysunek 12: Zmienione wartości SP1 i SP2

5 Podsumowanie

Podczas laboratorium zapoznaliśmy się z protokołem IO-Link oraz jego konfiguracją w sterowniku S7-1200. Wykonaliśmy ćwiczenie, w którym skonfigurowaliśmy sterownik, a następnie odczytaliśmy dane z czujnika ultradźwiękowego. Następnie zaimplementowaliśmy logikę sterownika PLC, która w zależności od odległości od czujnika, zaświecała kontrolki niebieską i białą. Nauczyliśmy się również jak zapisywać dane na żądanie oraz jak odczytywać je w czasie rzeczywistym.