

# Sprawozdanie z laboratorium 1

Podstawy PLC (IDEC)

**Łukasz Janusz**  
**Marek Generowicz**  
10.04.2025



**AGH**

**AGH UNIVERSITY OF KRAKOW**

# 1 Wstęp

Na zajęciach należało zaprogramować sterownik FC6A marki IDEC. Wykonano to w środowisku WindLDR V8.

Sterownik znajdował się w skrzynce sterowniczej, która była podłączona do komputera. W skrzynce znajdowały się dwa moduły rozszerzeń, które były podłączone do sterownika. Użytkownik miał możliwość kontrolowania 6 wyjść analogowych oraz panelu sterowniczego, jednak na zajęciach należało się skupić tylko na części z nich. Wnętrze skrzynki sterowniczej przedstawia zdjęcie 1a.

Stanowisko składało się z kaskadowego układu zbiorników bezciśnieniowych. Ruch cieczy do zbiorników wyżej położonych umożliwiał silnik elektryczny, zasilany falownikiem umożliwiającym regulację prędkości silnika. W dół ciecz spływała grawitacyjnie. Połączenia między zbiornikami były kontrolowane za pomocą zaworów kulowych lub za pomocą zaworów regulacyjnych z pozycjonerami z elektrycznymi siłownikami silnikowymi. Cały układ przedstawia zdjęcie 1b.



(a) Skrzynka sterownicza (zdjęcie z konspektu)



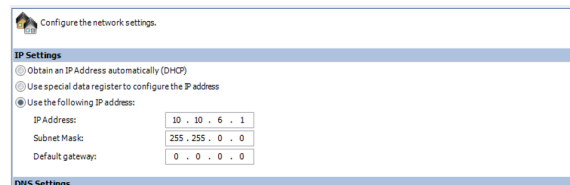
(b) Całe stanowisko (zdjęcie z konspektu)

Zdjęcie 1: Skrzynka sterownicza oraz stanowisko (zdjęcia z konspektu)

## 2 Konfiguracja sterownika

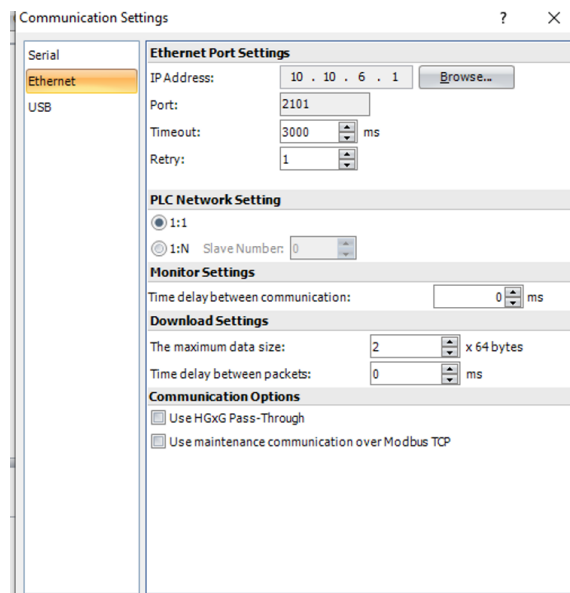
Przed przystąpieniem do programowania sterownika, należało skonfigurować parametry komunikacji.

Na zdjęciu 2 pokazane są parametry, które należało ustawić jako parametry sieci w PLC, aby ten mógł poprawnie połączyć się z fizycznym sterownikiem.



Zdjęcie 2: Konfiguracja sieci dla sterownika

Następnie skonfigurowano ustawienia komunikacji programu. W tym celu należało ustawić komunikację przez *Ethernet* oraz ustawić odpowiednie parametry. Na zdjęciu 3 pokazano jakie dokładnie wartości należało ustawić.



Zdjęcie 3: Konfiguracja komunikacji programu

Po skonfigurowaniu połączenia możliwe jest automatyczne podpięcie modułów wejść oraz wyjść w aplikacji. Dzięki temu do układu zostały podpięte dwa moduły analogowe, które następnie należało skonfigurować zgodnie z dokumentacją stanowiska, moduł pierwszy, zgodnie z zdjęciem 4a odpowiada za wejścia, a moduł drugi, zgodnie z zdjęciem 4b odpowiada za wyjścia. Dzięki takiej konfiguracji możliwe jest podpięcie odczyt oraz kontrola zmiennych procesowych.

Channel	Filter	Signal Type	Data Type	Min.	Max.
AI0	0	0 to 10V DC	Optional range (16bit)	0	10000
AI1	0	0 to 10V DC	Optional range (16bit)	0	10000
AI2		Unused			
AI3	0	4 to 20mA DC	Optional range (16bit)	0	10000
AI4	0	4 to 20mA DC	Optional range (16bit)	0	10000
AI5	0	4 to 20mA DC	Optional range (16bit)	0	10000
AI6	0	0 to 10V DC	Optional range (16bit)	0	10000
AI7	0	0 to 10V DC	Optional range (16bit)	0	10000

(a) Moduł I

Channel	Filter	Signal Type	Data Type	Min.	Max.
AQ0	0	0 to 10V DC	Optional range	0	10000
AQ1	0	0 to 10V DC	Optional range	0	10000
AQ2		Unused			
AQ3	0	4 to 20mA DC	Optional range	0	10000

(b) Moduł II

Zdjęcie 4: Konfiguracja rejestrów dla modułów

Po podpięciu modułów, możliwe jest odczytanie ich wartości. W tym celu należało uruchomić opcje *Monitor* w zakładce *Module Configuration*. Dzięki temu możliwe jest odczytanie wartości wejść, które są podpięte do modułu 1. Zdjęcie 5 przedstawia jak wygląda moduł 1 po połączeniu z sterownikiem.

Ważne aby pamiętać że wyniki wejść nie są w jednostkach fizycznych, a w jednostkach cyfrowych. W celu przeliczenia ich na jednostki fizyczne, w tym celu należy podzielić podaną wartość przez 100 lub dla przepływu przez 400.

Slot 1: FC6A-J8A4 (Analog I/O Module)							
Status: No error							
System software version: 1.03							
Channel	Filter	Signal Type	Data Type	Min.	Max.	Data	Status
AI0	0	0 to 10V DC	Optional range (16bit)	0	10000	2485	0
AI1	0	0 to 10V DC	Optional range (16bit)	0	10000	2457	0
AI2		Unused					
AI3	0	4 to 20mA DC	Optional range (16bit)	0	10000	613	0
AI4	0	4 to 20mA DC	Optional range (16bit)	0	10000	570	0
AI5	0	4 to 20mA DC	Optional range (16bit)	0	10000	0	0
AI6	0	0 to 10V DC	Optional range (16bit)	0	10000	0	0
AI7	0	0 to 10V DC	Optional range (16bit)	0	10000	3886	0

Zdjęcie 5: Wartości zmiennych w module 1

Alternatywnie można użyć opcji *Batch Monitor*, która pozwala na odczytanie które z zmiennych procesowych są używane w danym momencie. Zdjęcie 6 przedstawia jak wygląda opcja *Batch Monitor*.

Warto zauważyć że do dyspozycji w wierszu pierwszym mamy 14 komórek. W przypadku naszego sterownika mamy do dyspozycji 14 wejść więc w przypadku odczytu wejść jedynie przydatny pierwszy wiersz tabeli.

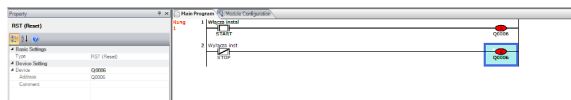
Device:	[ Input4 ]														0	Monitor Type:	BIN (B)																		
Comment:																																			
	+17	+16	+15	+14	+13	+12	+11	+10	+9	+8	+7	+6	+5	+4	+3	+2	+1	+0	DEC (N)																
20000			0	0	0	0	0	1	0		1	1	1	1	1	0	1	0	378																

Zdjęcie 6: Batch monitor

### 3 Zmienne procesowe

Po skonfigurowaniu sterownika oraz podpięciu modułów, należało przestąpić do programowania sterownika tak aby można było go kontrolować.

W pierwszej kolejności należało stworzyć drabinę pozwalającą włączać oraz resetować przełącznik  $K4$ , który jest przypisany do zmiennej  $Q6$ . Aby to zrobić, warto było stworzyć tagi rejestrów nazywającym się *START* oraz *STOP*. Następnie tak przypisane tagi należało odpowiednio podpiąć do zmiennej  $Q6$ . Program powinien wyglądać jak na zdjęciu 7.



Zdjęcie 7: Program do włączania oraz resetowania przełącznika

Następnie należało dodać drabinę dzięki której wraz z zmianą wartości zmiennej  $Q6$  zmienia zapala się zielona lampka jak na zdjęciu 8. W tym celu należało ustawić że wartość zmiennej  $Q4$  zmienia się wraz z zmianą wartości zmiennej  $Q6$ .



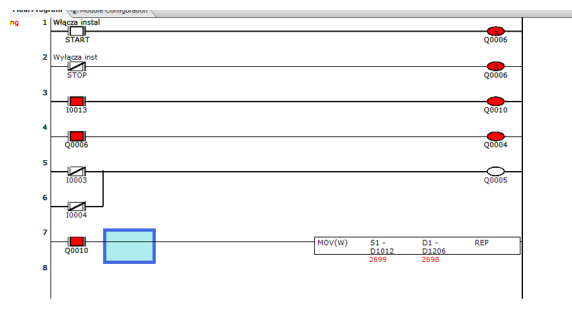
Zdjęcie 8: Działanie zielonej lampki

Dodatkowo należało stworzyć drabinki które za zadanie miały:

- Kontrole pompy za pomocą zmiennej  $I13$  odpowiadającej za przełącznik na skrzynce sterowniczej.
- Kontrole pomarańczowej lampki, która zapala się przy dostaniu sygnału od jednej z czujek poziomu przypisanym do wartości  $I03$  oraz  $I04$ .
- Kontrole wydajności pompy za pomocą zmiennej  $Q10$  odpowiadającego potencjometrowi na skrzynce sterowniczej.

Dodatkowo należało w zakładce *Monitor/Custom* dodać zmienne procesowe  $D1200$  oraz  $D1202$  odpowiadającym za kontrolowanie przepustowości zaworów.

Zbudowanie takiego programu pozwala na kontrolowanie całego układu. Kod odpowiadający za logikę programu przedstawia zdjęcie 9.



Zdjęcie 9: Program do kontroli stanowiska

Na zdjęciach 10 przedstawiono jak wygląda włączony układ w przypadku przepełnienia jednego z zbiorników, powodujące włączenie pomarańczowej lampki.

Slot 1: FC6A-K4A4 (Analog I/O Module)

Status: No error

System software version: 1.03

Channel	Filter	Signal Type	Data Type	Min.	Max.	Data	Status
AI0	0	0 to 10V DC	Optional range (16bit)	0	10000	2432	0
AI1	0	0 to 10V DC	Optional range (16bit)	0	10000	2417	0
AI2		Unused					
AI3	0	4 to 20mA DC	Optional range (16bit)	0	10000	4208	0
AI4	0	4 to 20mA DC	Optional range (16bit)	0	10000	571	0
AI5	0	4 to 20mA DC	Optional range (16bit)	0	10000	575	0
AI6	0	0 to 10V DC	Optional range (16bit)	0	10000	4160	0
AI7	0	0 to 10V DC	Optional range (16bit)	0	10000	9890	0

Slot 2: FC6A-K4A4 (Analog I/O Module)

Status: No error

System software version: 1.03

Channel	Filter	Signal Type	Data Type	Min.	Max.	Data	Status
AQ0	0	0 to 10V DC	Optional range	0	10000	10000	0
AQ1	0	0 to 10V DC	Optional range	0	10000	10000	0
AQ2		Unused					
AQ3	0	4 to 20mA DC	Optional range	0	10000	9890	0

(a) Modul I

(b) Modul II

(a) Moduł I

(b) Moduł II

Zdjęcie 10: Stan awaryjny w przypadku przepełnienia zbiornika

## 4 Podsumowanie

Na zajęciach mieliśmy możliwość zapoznania się z sterownikiem PLC firmy IDEC, oraz również nauczenia się jak można zaprogramować analogowe wejścia oraz wyjścia znajdujące się w skrzynce sterowniczej. Dzięki temu można było nauczyć się jak można skonfigurować sterownik oraz jak można go zaprogramować. Dodatkowo można było nauczyć się jak można kontrolować zmienne procesowe oraz jak można je odczytać. Dodatkowo można było się nauczyć jak zaprogramować sterownik aby nie wysyłał tylko informacji zero-jedynkowej ale również ciągłe, co posłużyło nam do kontroli pompy.

Całość zajęć była bardzo interesująca, ponieważ można było zobaczyć jak można wykorzystać sterownik PLC do kontroli układu. Dodatkowo można było nauczyć się jak można skonfigurować oraz zaprogramować sterownik PLC.