

Notatka służbowa nr 4	
Temat:	Regulator PID na sterowniku VersaMax
Wykonanie:	Zuzanna Mejer, 259382
Termin zajęć:	poniedziałek TP, 10:55
Data:	16.12.2022

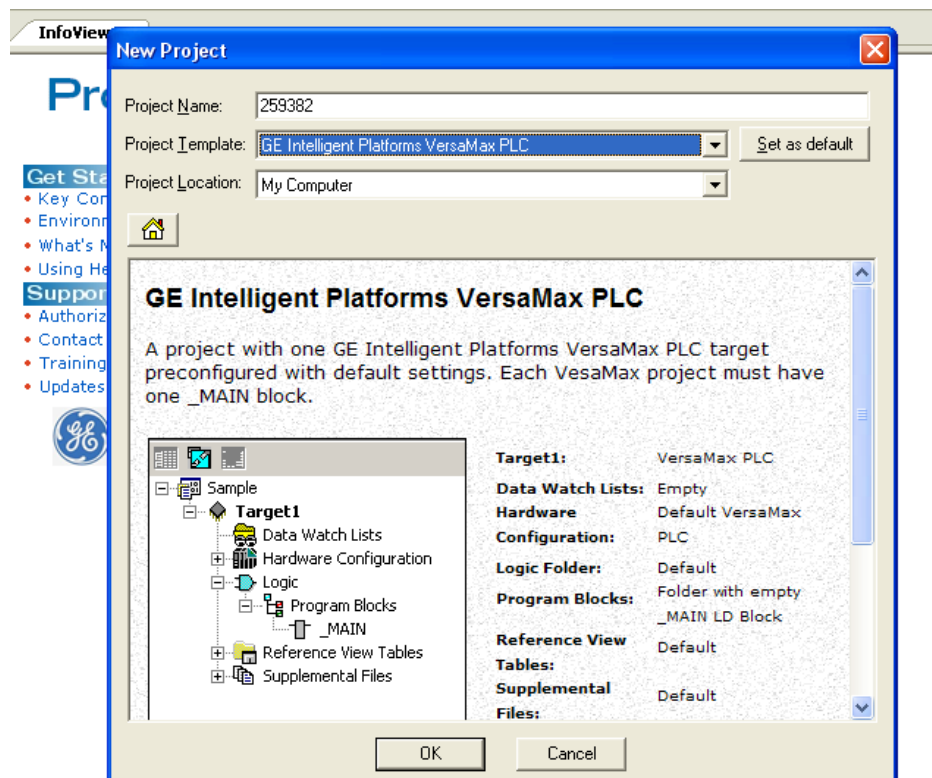
## 1. Cel ćwiczenia

Celem ćwiczenia było utworzenie projektu regulacji PID oraz analiza wykresów regulacji. Ćwiczenie wykonano na sterowniku PLC VersaMax z wykorzystaniem programu Proficy Machine Edition 8.0.

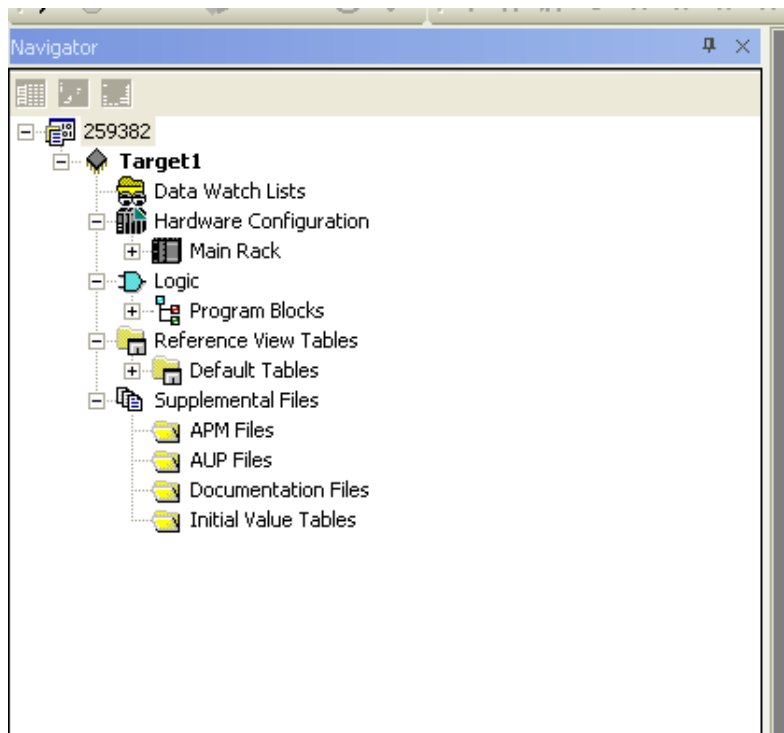
## 2. Uruchomienie oprogramowania i konfiguracja sterownika

Przed rozpoczęciem pracy na sterowniku, należało uruchomić oprogramowanie Proficy Machine Edition i skonfigurować sterownik. W tym celu wykonano poniższe czynności:

1. Utworzono nowy projekt wybierając z głównego menu *File* → *New Project*. Wpisano tytuł projektu oraz wybrano szablon *GE Intelligent Platform VersaMax PLC* (rys. 1). Poprawnie utworzony projekt widoczny w zakładce *Navigator* przedstawiono na rys. 2.



Rys. 1: Utworzenie nowego projektu w programie Proficy Machine Edition



Rys. 2: Poprawnie utworzony projekt

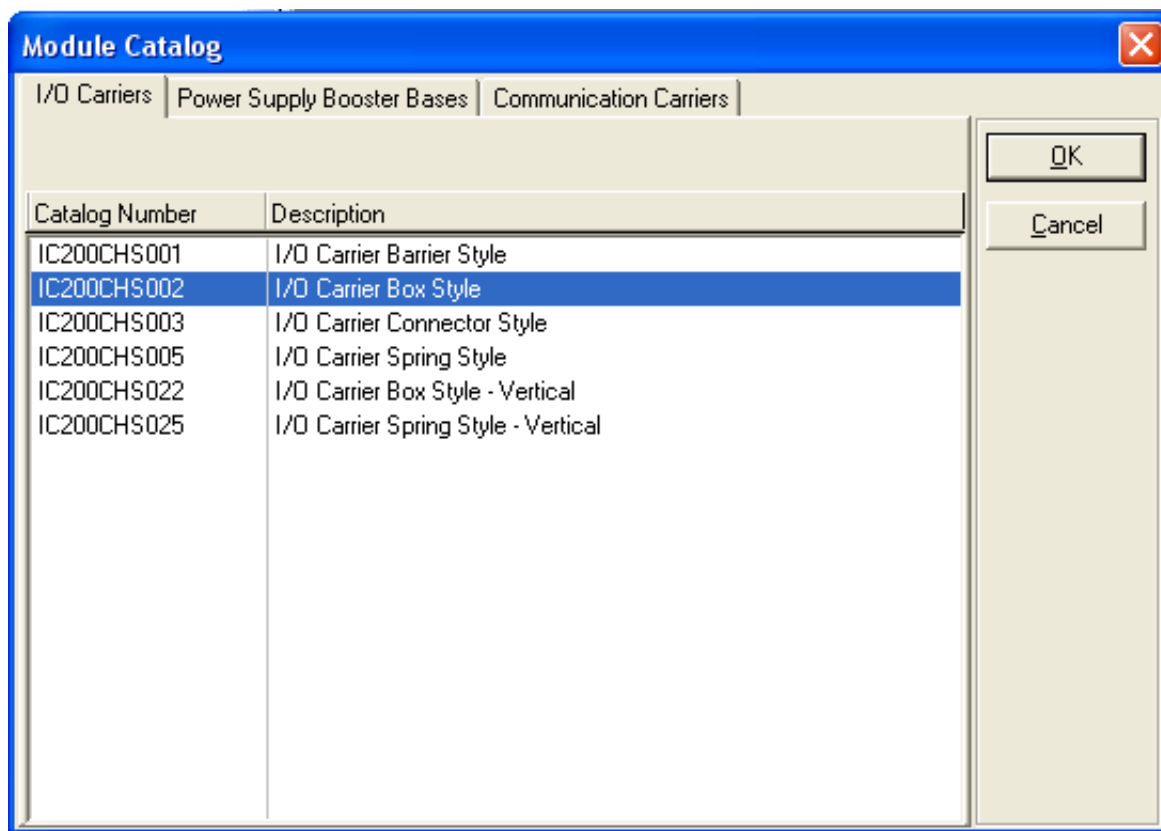
2. Rozwijając *Hardware Configuration* → *Main Rack* pokazały się moduły sterownika. Najpierw, klikając na moduł *PWR* prawym klawiszem i wybierając *Replace Module*, uzupełniono nazwę modułu jako *IC200PWR002/012* (rys. 3). O poprawności wybrania modułu PWR świadczy pojawienie się zielonego elementu przy ikonie, co przedstawia rys. 4.



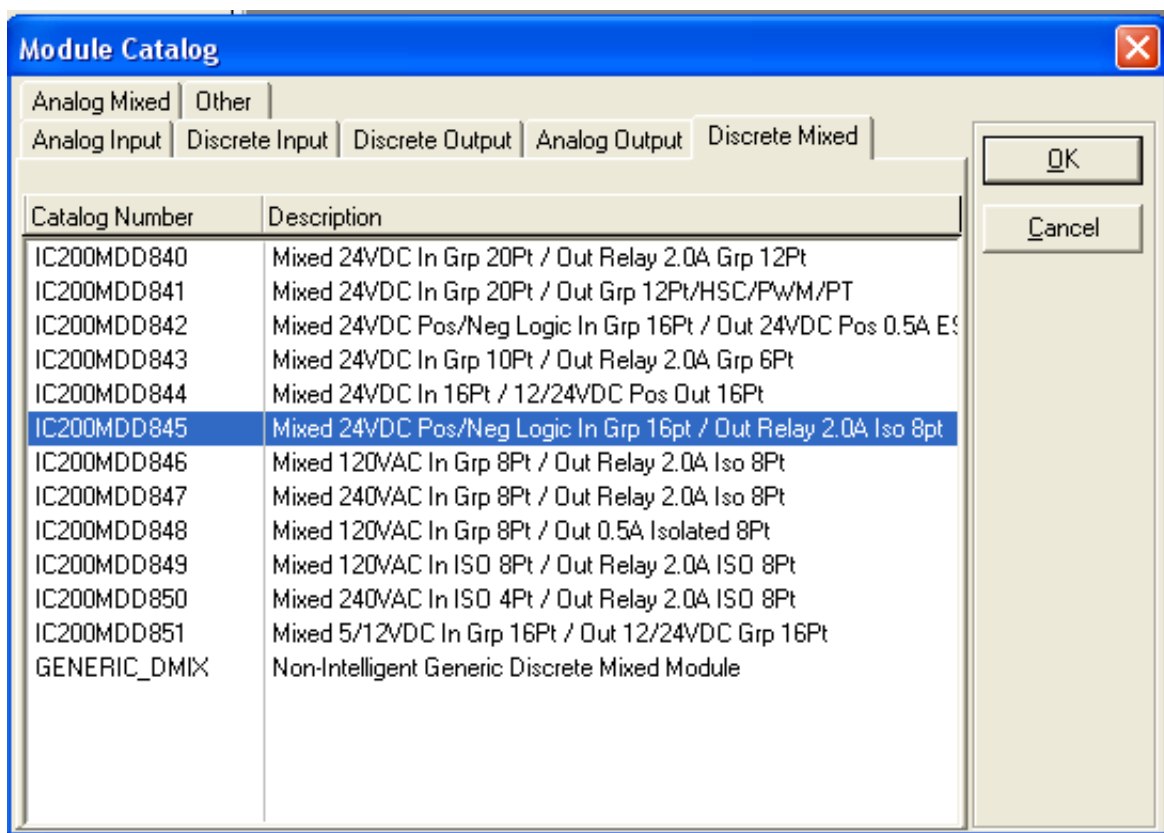
z IC200CPU001 na IC200CPUE05. Pozostałe 2 „kasety” należało dodać najpierw wybierając podstawkę (opcja *Add Carrier/Base*) - rys. 5, a następnie wybierając właściwy moduł opcją *Add Module* - rys. 6. W ten sposób zostały zdefiniowane poszczególne „kasety”:

Slot 1 → IC200MDD845

Slot 2 → IC200ALG430.

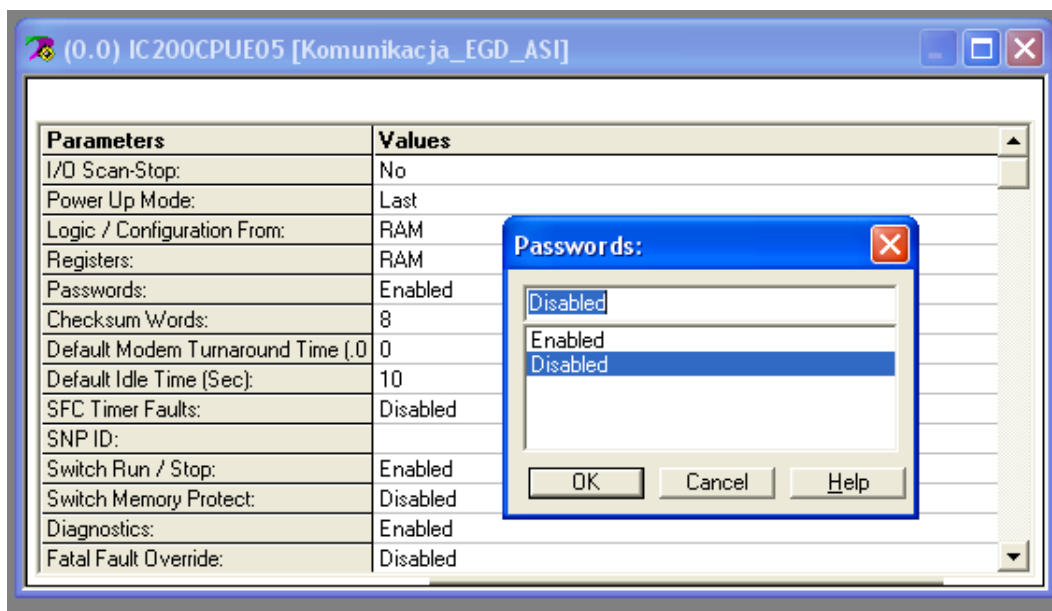


Rys. 5: Przykład wybranej podstawki

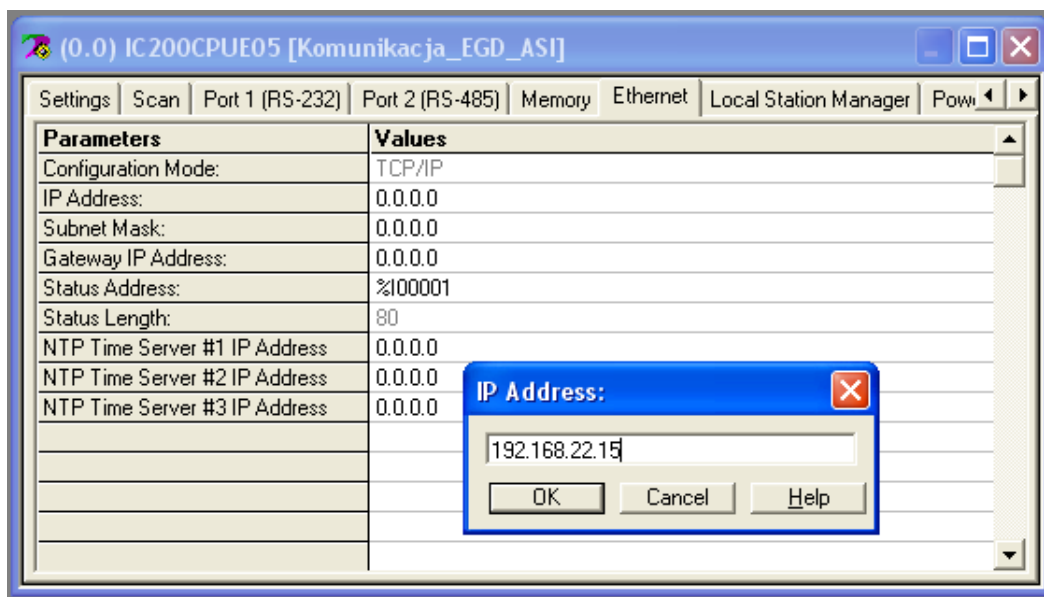


Rys. 6: Przykład wybranego modułu

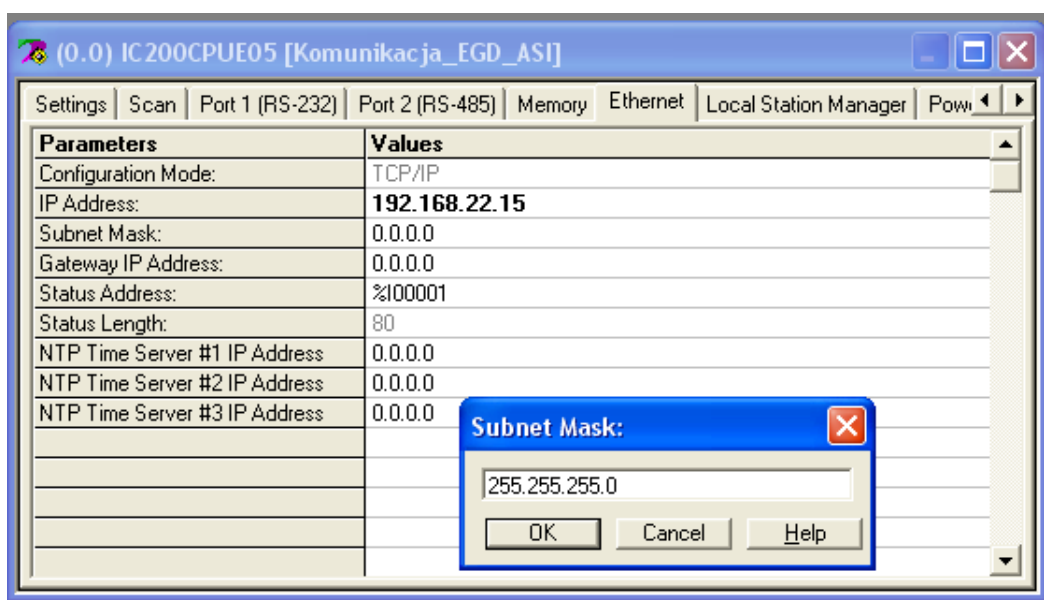
- Następnie wykonano konfigurację jednostki centralnej, to znaczy: zdezaktywowano hasło (rys. 7), uzupełniono adres IP (rys. 8) oraz maskę (rys. 9), oraz przeniesiono obszar pamięci statusu od adresu początkowego %I100 (rys. 10).



Rys. 7: Zdezaktywowanie hasła



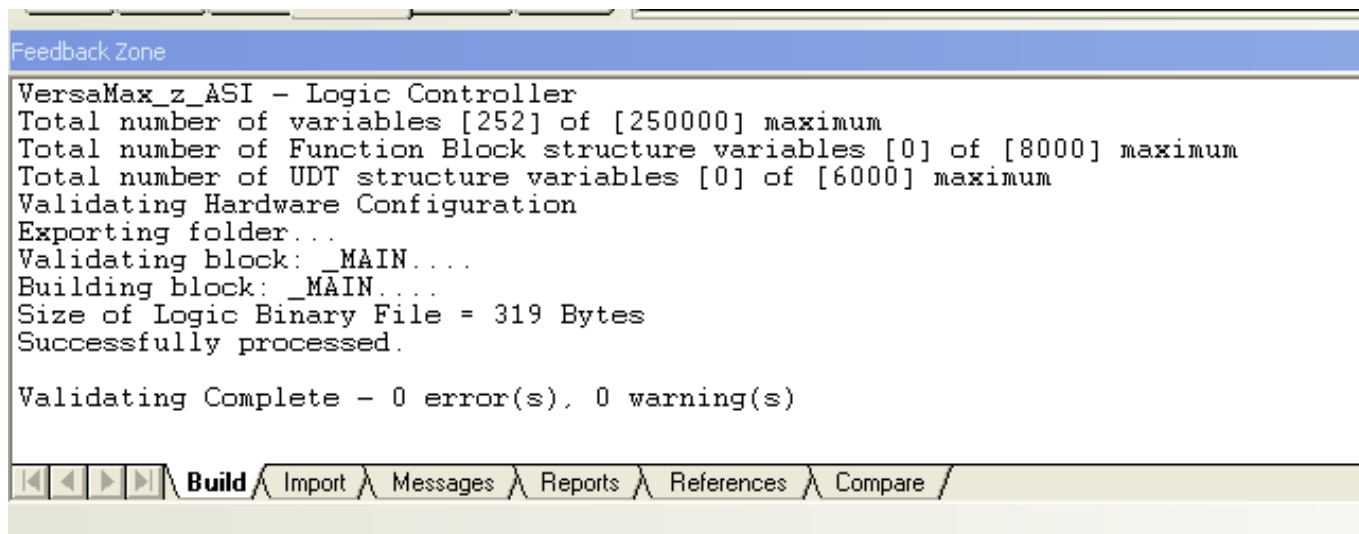
Rys. 8: Zmiana adresu IP



Rys. 9: Uzupełnienie maski



6. Po zakończeniu konfiguracji sprzętowej przeprowadzono walidację projektu (rys. 12).



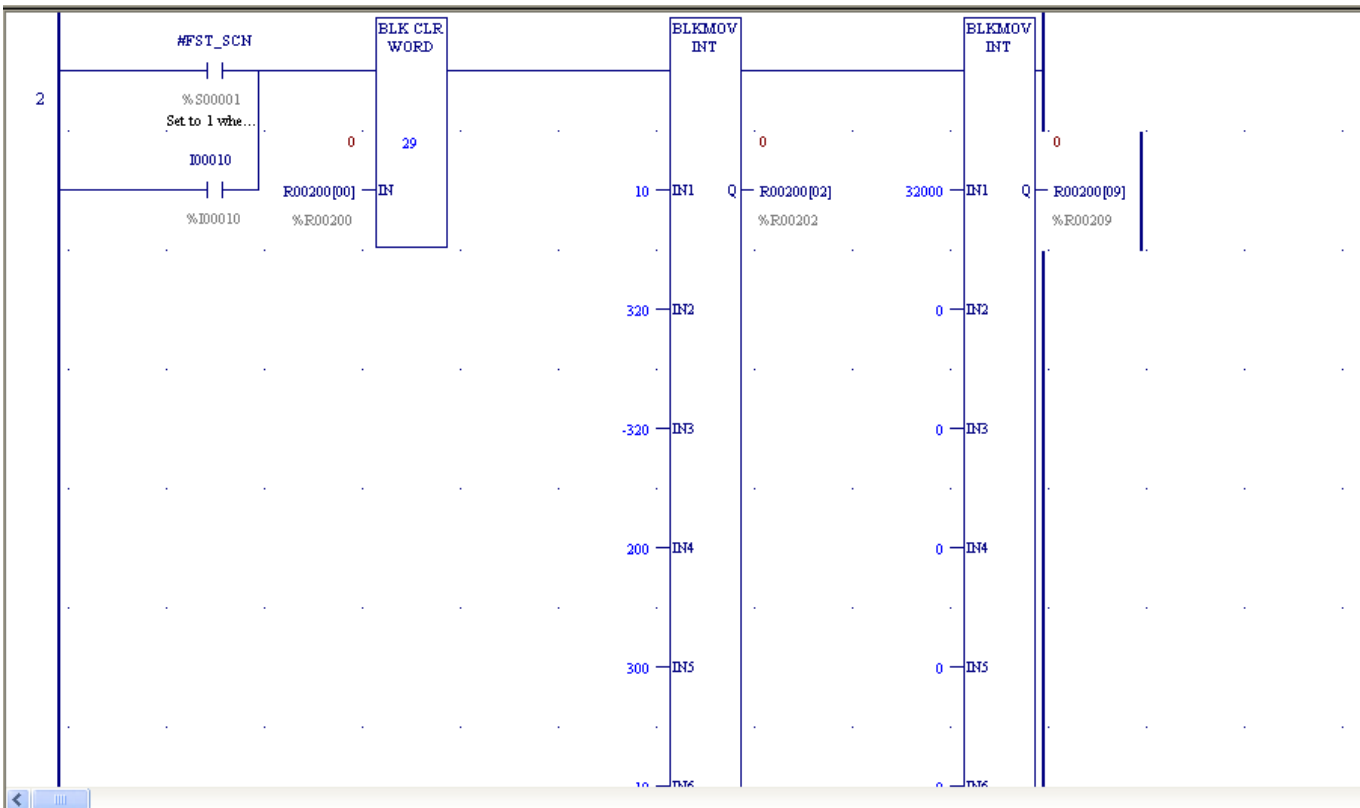
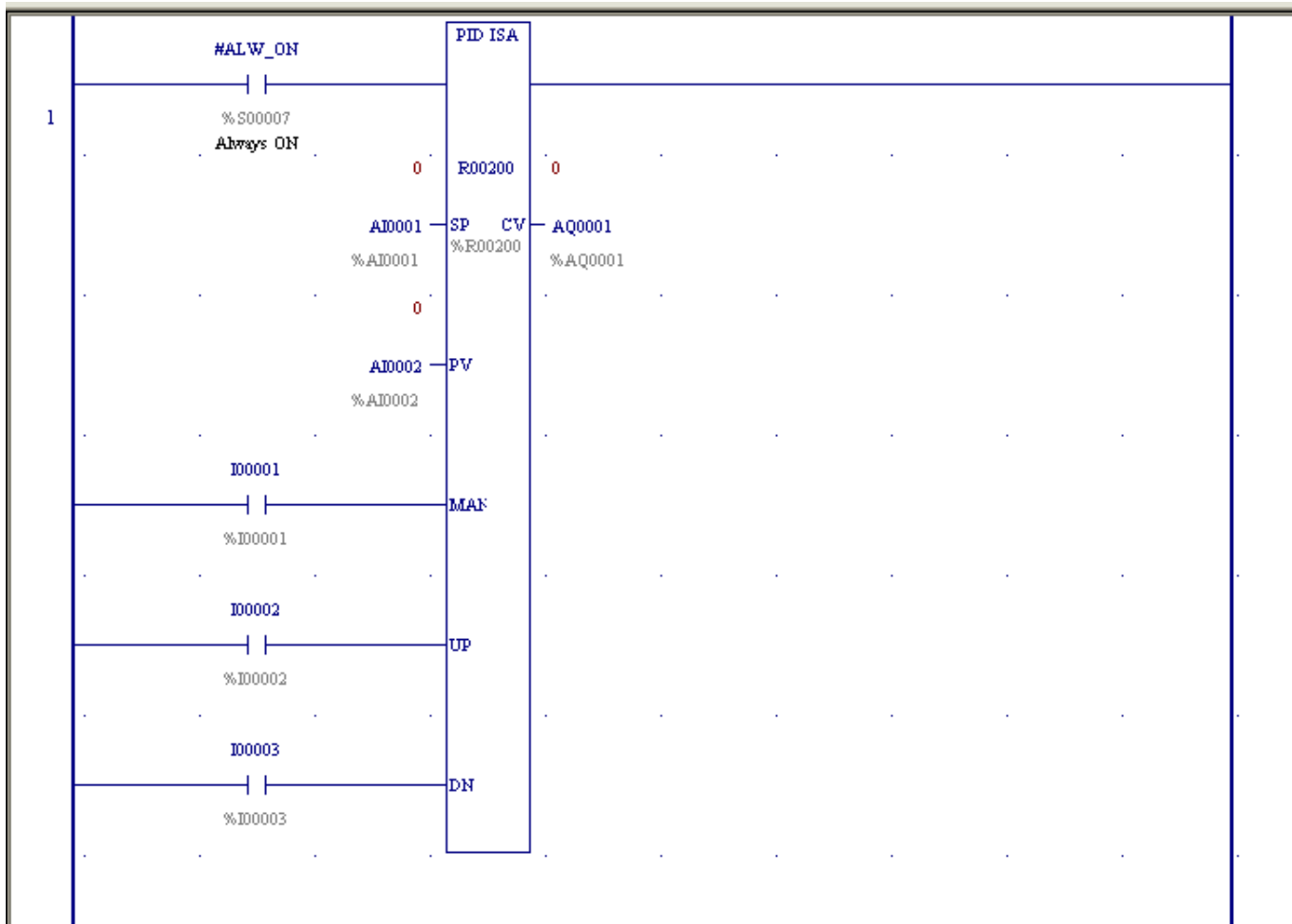
Rys. 12: Walidacja projektu

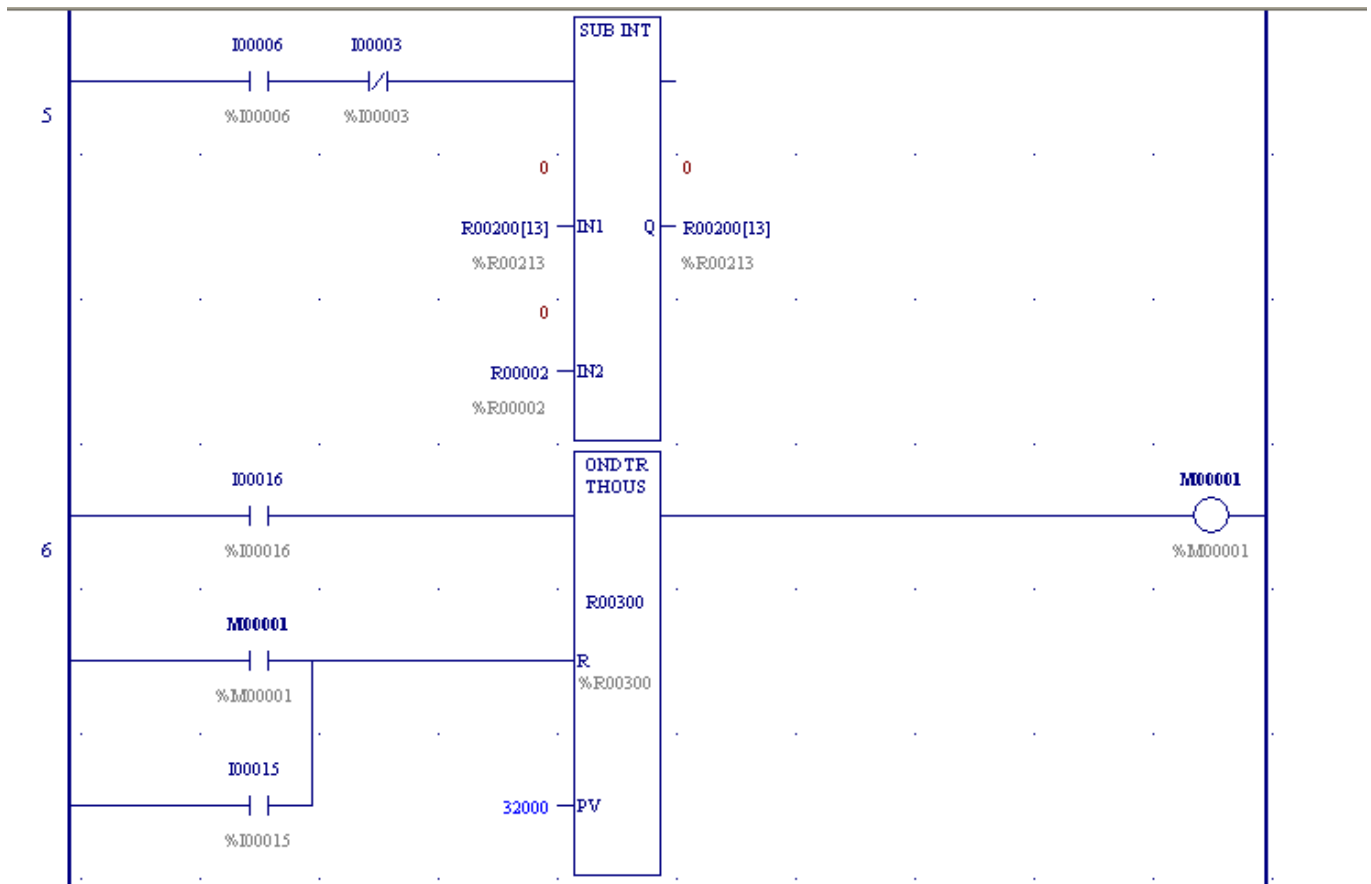
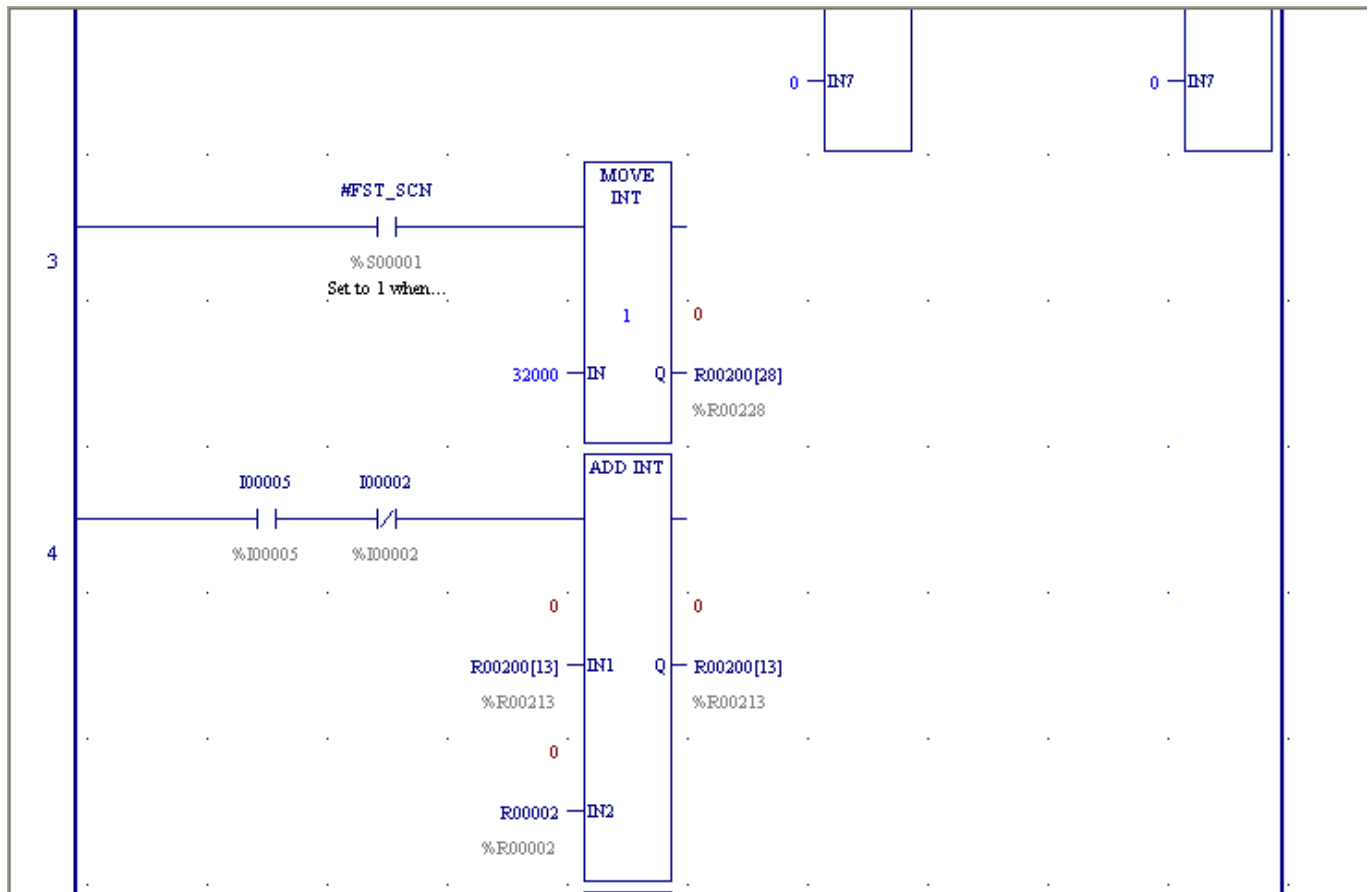
### 3. Program do regulacji PID

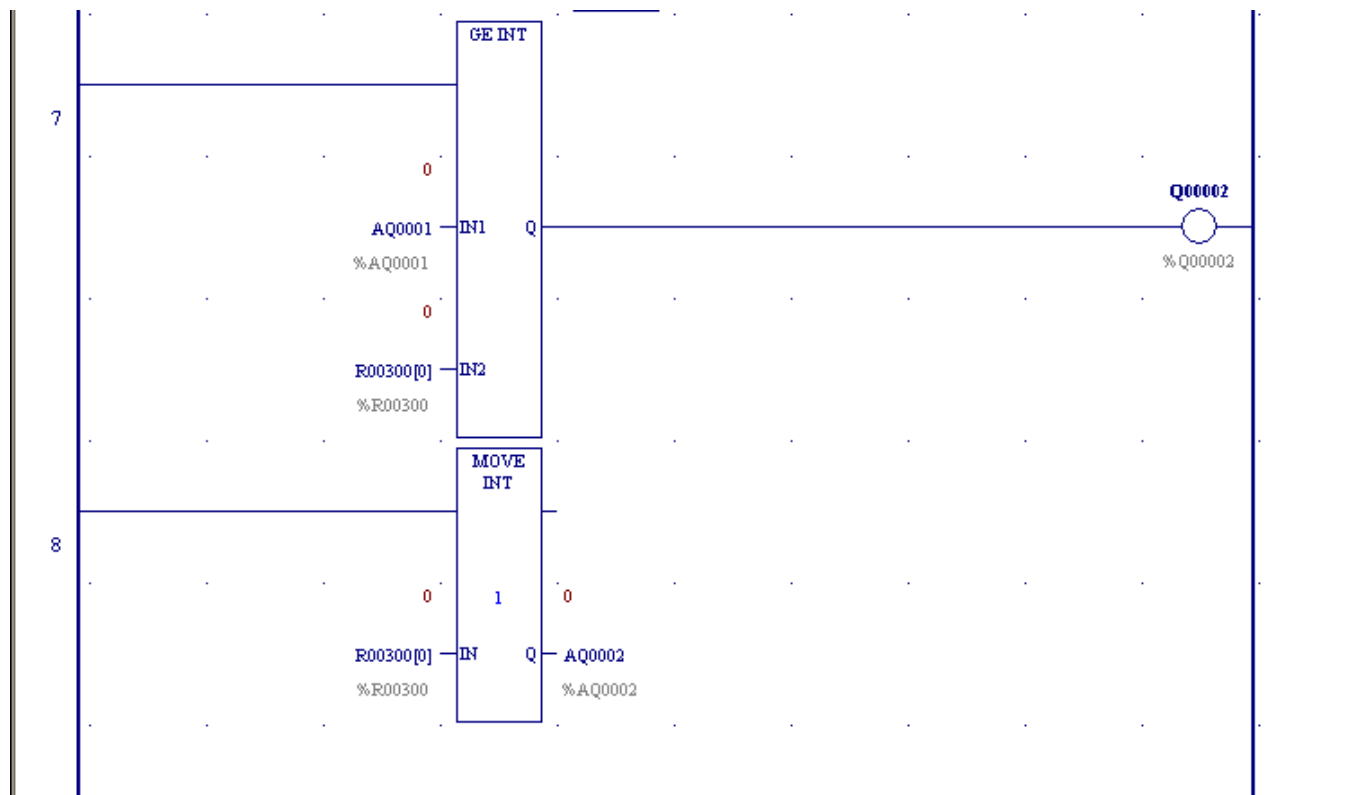
#### 3.1. Program w języku drabinkowym

W celu przeprowadzenia badania odpowiedzi skokowych regulatora PID wprowadzony został następujący program w sekcji *Logic* → *Program Blocks* → *MAIN*:









Rys. 13: Program regulatora PID w języku drabinkowym

### 3.2. Uruchomienie i działanie programu

Po powtórzeniu walidacji całego programu, przesłano go do sterownika. Przed poleceniem połączenia wybrano port fizyczny komputera Ethernet i wpisano adres IP sterownika (rys. 14).

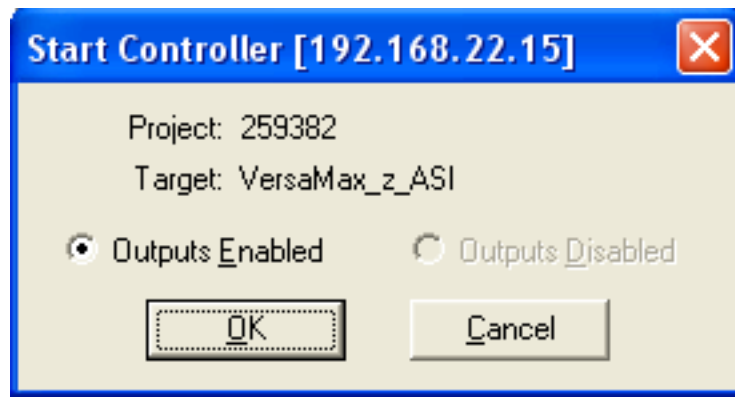
Inspector	
Target	
Name	VersaMax_z_ASI
Type	GE IP Controller
Description	
Documentation Address	
Family	VersaMax PLC
Controller Target Name	A593822
Update Rate (ms)	250
Sweep Time (ms)	Offline
Controller Status	Offline
Enable Shared Variables	False
Physical Port	ETHERNET
<b>IP Address</b>	192.168.22.15
Additional Configuration	

Rys. 14: Port fizyczny Ethernet i adres IP

Przesyłanie rozpoczyna się od otwarcia okna *Download to Controller* (rys. 15), następnie pojawia się okno *Start Controller* (rys. 16).

Download to Controller [192.168.22....]	
<input checked="" type="checkbox"/>	Hardware Configuration
<input checked="" type="checkbox"/>	Logic
<input type="checkbox"/>	Initial/Forced Values
Project: 259382	
Target: VersaMax_z_ASI	
<input type="checkbox"/>	Write ALL items to flash memory
<input type="button" value="Ok"/> <input type="button" value="Cancel"/> <input type="button" value="Help"/>	

Rys. 15: *Download to Controller*

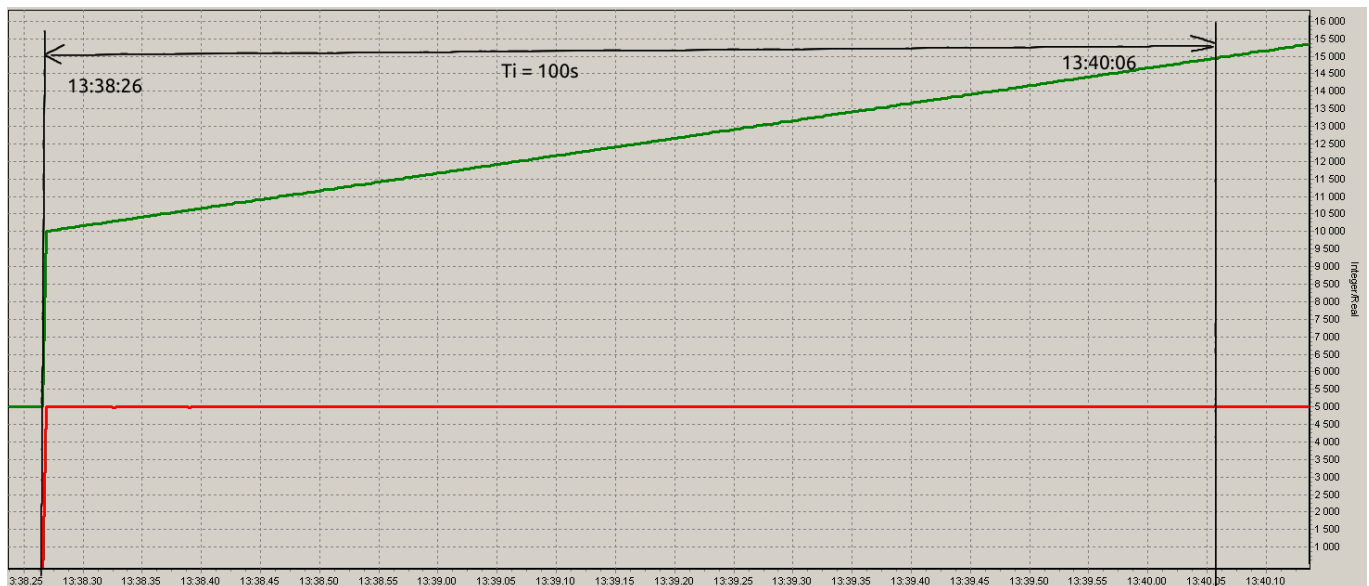


Rys. 16: *Start Controller*

## 4. Analiza przebiegów

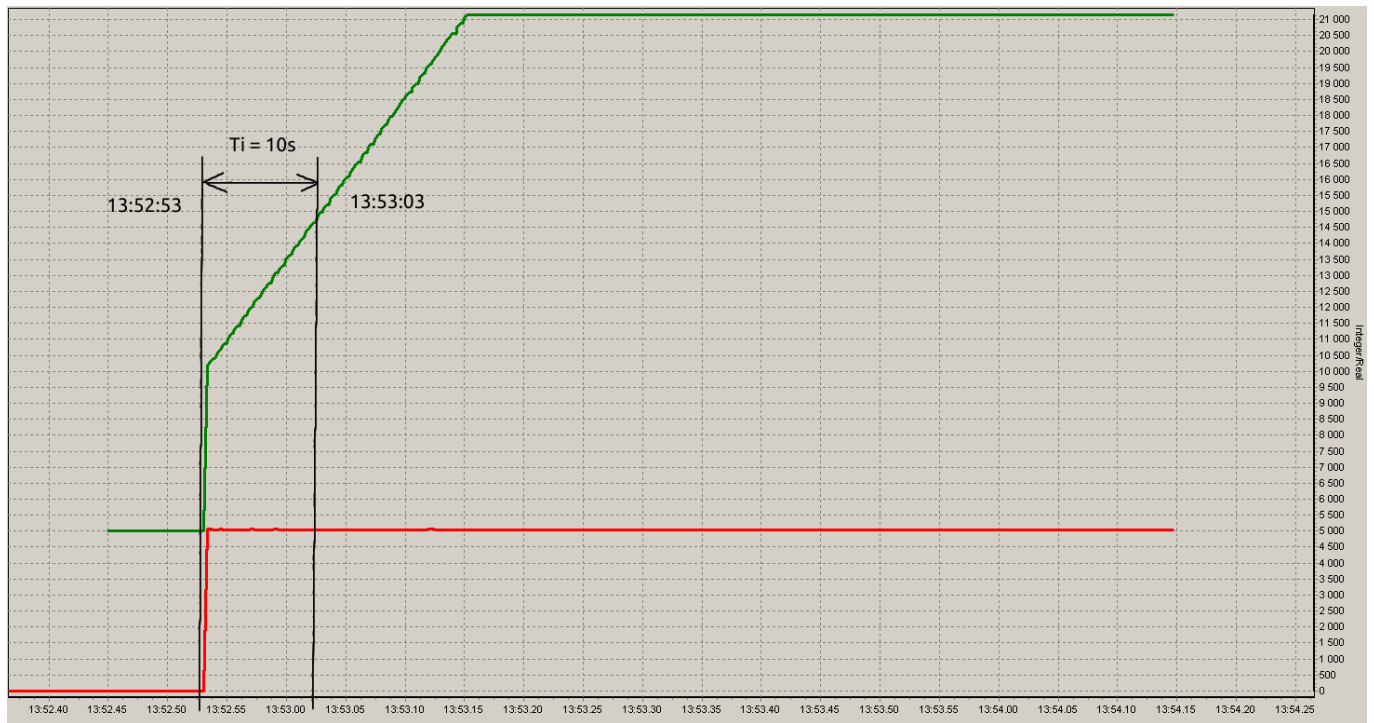
### 4.1. Regulator PI

Poprawność działania regulacji całkującej sprawdzono dla następujących danych: współczynnik wzmocnienia  $K_p = 1$ , czas zdwojenia  $T_i = 100s$  oraz  $T_i = 10s$ . Na wejście podano skok jednostkowy i uzyskano następujące przebiegi:



Rys. 17: Odpowiedź regulatora PI przy  $T_i = 100s$

Po 100 sekundach wartość odpowiedzi układu powinna być dwukrotną wartością początkową. Po 100 sekundach wartość wyniosła około 15000.

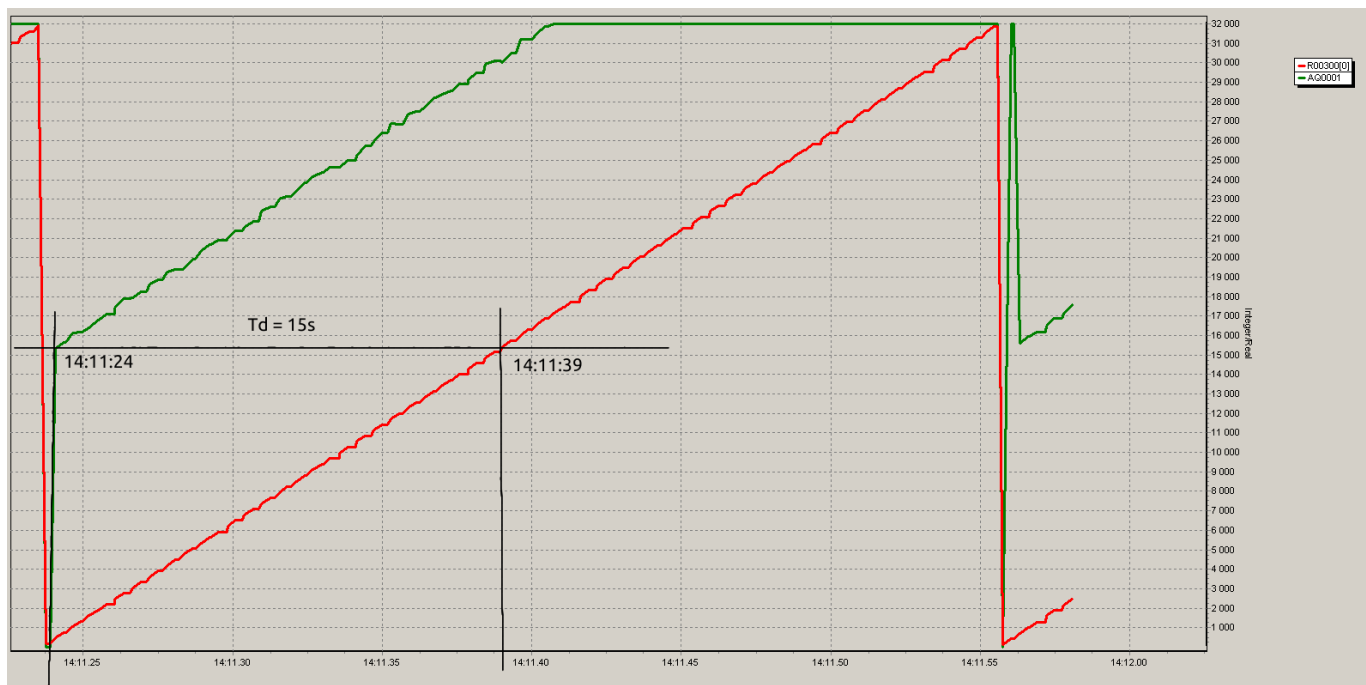


Rys. 18: Odpowiedź regulatora PI przy  $T_i = 10s$

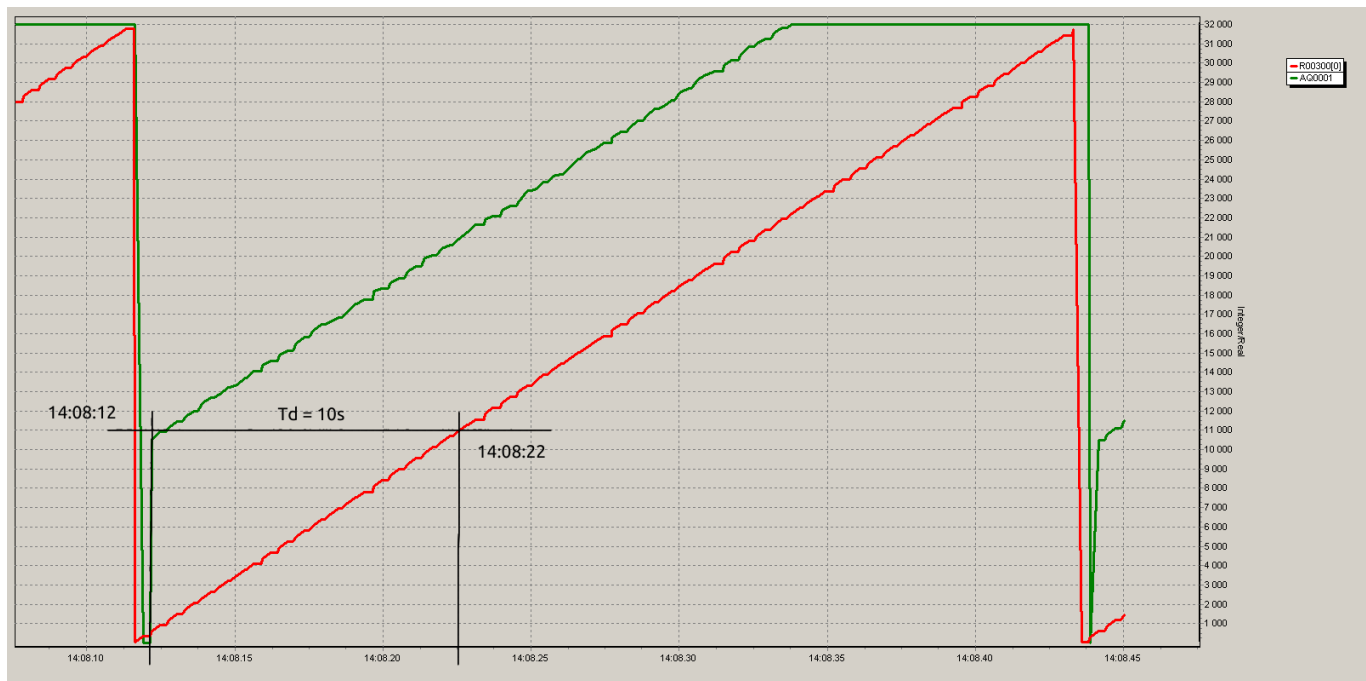
Po 10 sekundach wartość odpowiedzi skokowej układu wyniosła około 15000.

## 4.2. Regulator PD

Poprawność działania regulacji różniczkującej sprawdzono dla następujących danych: współczynnik wzmocnienia  $K_p = 1$ , czas wyprzedzenia  $T_d = 15s$  oraz  $T_d = 10s$ . Na wejście podano sygnał narastający liniowo i uzyskano następujące przebiegi:



Rys. 19: Odpowiedź regulatora PD przy  $T_d = 15s$



Rys. 20: Odpowiedź regulatora PD przy  $T_d = 10s$

## 5. Podsumowanie

Ćwiczenie nie sprawiło większych problemów. Najwięcej czasu zajął sam proces poprawnej konfiguracji sterownika oraz poprawnego przydzielenia adresów pamięci w programie. Mimo małych

trudności, ostatecznie udało się przesłać do sterownika działający program i przetestować generowanie przebiegów regulacji.

Na podstawie wykresów widocznych na rys. 17 oraz 18 udało się wyznaczyć czas zdwojenia. Osiągnięte wartości po tym czasie nie są dokładne. Rzeczywiste zdwojenie, które można odczytać z wykresów występuje po dłuższym czasie niż zadane.

Na podstawie wykresów widocznych na rys. 19 oraz 20 udało się wyznaczyć czas wyprzedzenia. Wartość osiągnięta przez sygnał narastający liniowo po zadanym czasie wyprzedzenia rzeczywiście osiągała wartość, jaką odpowiedź skokowa układu osiągnęła już na początku badania. Świadczy to o poprawności działania regulacji.