



SIEMENS



교육-/학습 문서

Siemens Automation Cooperates with Education
(SCE) | 버전 V14 SP1 부터

TIA Portal Module 051-300

SIMATIC S7-1200 을 위한 PID 컨트롤러

[siemens.com/sce](https://www.siemens.com/sce)

SIEMENS

Global Industry
Partner of
WorldSkills
International



worldskills

이러한 학습-/교육 문서와 일치하는 SCE 교육 담당자 패키지

- **SIMATIC S7-1200 AC/DC/RELAY(6 개 세트) "TIA Portal"**
주문 번호: 6ES7214-1BE30-4AB3
- **SIMATIC S7-1200 DC/DC/DC(6 개 세트) "TIA Portal"**
주문 번호: 6ES7214-1AE30-4AB3
- **업그레이드 SIMATIC STEP 7 BASIC V14 SP1(S7-1200 용)(6 개 세트) "TIA Portal"**
주문 번호: 6ES7822-0AA04-4YE5

이러한 교육 담당자 패키지는 필요 시 후임자 패키지로 대체됩니다. 다음 웹 사이트에서 현재 사용 가능한 SCE 패키지를 대략적으로 확인할 수 있습니다. [siemens.com/sce/tp](https://www.siemens.com/sce/tp)

교육 연장

지역별 Siemens SCE 교육 연장은 지역 SCE 담당자에게 문의하십시오. [siemens.com/sce/contact](https://www.siemens.com/sce/contact)

SCE 관련 추가 정보

[siemens.com/sce](https://www.siemens.com/sce)

사용 관련 정보

통합 자동화 솔루션 TIA(Totally Integrated Automation)에 대한 본 SCE 학습-/교육 문서는 특히 공공 교육 기관 및 R&D 기관의 교육 목적으로 "SCE(Siemens Automation Cooperates with Education)" 프로그램을 위해 준비되었습니다. Siemens AG 는 내용을 보장하지 않습니다.

이 문서는 Siemens 제품/시스템에 대한 최초 교육용으로만 사용해야 합니다. 이 문서의 전체 또는 일부를 복사해 교육을 받는 사람들에게 제공해 교육 범위 내에서 사용할 수 있습니다. 이 학습-/교육 문서 배포 또는 복사와 내용 공유는 교육 목적의 공개 교육 및 고등 교육 기관에서만 허용됩니다.

그 외의 경우에는 다음 Siemens AG 담당자의 서면 동의가 필요합니다. Roland Scheuerer roland.scheuerer@siemens.com.

이를 위반하면 법적 책임을 지게 됩니다. 특히 특허가 부여되거나 실용신안 또는 디자인이 등록되어 있는 경우 번역을 포함해 모든 권리가 보장됩니다.

산업 고객을 위한 과정에서의 사용은 명시적으로 허용되지 않습니다. 본 학습-/교육 문서를 상업적으로 사용하는 데 동의하지 않습니다.

TU Dresden, 특히 Prof. Dr.-Ing. Leon Urbas 및 Michael Dziallas Engineering Corporation 을 비롯한 모든 관계자들에게 이 학습-/교육 문서를 준비하는 동안 보내주신 성원에 대해 감사를 표하고자 합니다.

목차

1	목적	4
2	사전조건	4
3	필수 하드웨어와 소프트웨어	5
4	폐쇄 루프 제어 이론	6
4.1	폐쇄 루프 제어 태스크	6
4.2	제어 루프의 구성 요소	6
4.3	제어 대상 시스템을 분석하기 위한 스텝 평선	8
4.4	자기 조절(Self-regulation)이 되는 제어 대상 시스템	9
4.4.1	시간 지연 없는 비례 시스템	9
4.4.2	시간 지연 있는 비례 시스템	10
4.4.3	시간 지연이 2 회 발생하는 비례 시스템	11
4.4.4	시간 지연이 n 회 발생하는 비례 시스템	12
4.5	자기 조절이 되지 않는 시스템	13
4.6	연속 컨트롤러(Continuous controllers)의 기본 유형	14
4.6.1	비례 컨트롤러(P 컨트롤러)	15
4.6.2	적분 컨트롤러(I 컨트롤러)	17
4.6.3	PI 컨트롤러	18
4.6.4	미분 컨트롤러(D 컨트롤러)	19
4.6.5	PID 컨트롤러	19
4.7	발산 테스트를 사용한 컨트롤러 튜닝	20
4.8	Tu-Tg 근사치를 통한 컨트롤러 조정	21
4.8.1	Ziegler-Nichols 방법에 따른 PI 컨트롤러 조정	22
4.8.2	Chien, Hrones 및 Reswick 방법에 따른 PI 컨트롤러 조정	22
4.9	디지털 컨트롤러	23
5	과제	25
6	계획 수립	25
6.1	PID_Compact 폐쇄 루프 제어 블록	25
6.2	기술 다이어그램	26
6.3	참조 목록	27
7	단계별 따라 해보기	28
7.1	기존 프로젝트 압축 풀기	28
7.2	순환 인터럽트 OB 에서 PID_Compact 컨트롤러 호출	30
7.3	프로그램 저장 및 컴파일	37
7.4	프로그램 다운로드	38
7.5	PID_Compact 모니터링	39
7.6	PID_Compact 사전 조정	41
7.7	PID_Compact 미세 조정	44
7.8	프로젝트 아카이브	47
8	체크리스트	48
9	추가 정보	49

SIMATIC S7-1200 용 PID 컨트롤러

1 목적

이 장에서는 TIA Portal 프로그래밍 도구를 사용한 SIMATIC S7-1200 용 소프트웨어 PID 컨트롤러 사용을 학습합니다.

모듈에서 SIMATIC S7-1200 용 PID 컨트롤러 호출, 연결, 컨피규레이션 및 최적화를 설명합니다. TIA Portal 에서 PID 컨트롤러를 호출하고 사용자 프로그램에 통합하는 단계도 설명합니다.

3 장에 나열된 SIMATIC S7 컨트롤러를 사용할 수 있습니다.

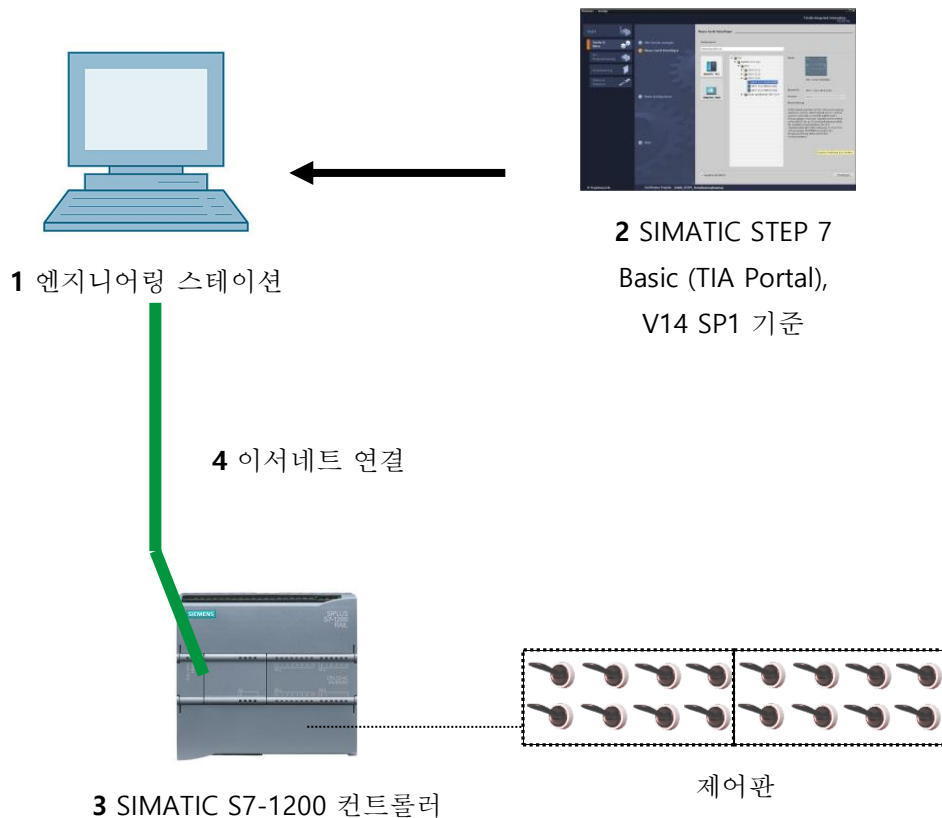
2 사전조건

이 장은 SIMATIC S7 CPU1214C DC/DC/DC 포함 아날로그 값 장을 기반으로 합니다. 이 장의 경우 다음 프로젝트를 사용할 수 있습니다. 예: "SCE_EN_031-500_Analog_Values_S7-1200.zip14".

3 필수 하드웨어와 소프트웨어

- 1 엔지니어링 스테이션: 전제조건에는 하드웨어 및 운영 체제가 포함됩니다.
(추가 정보는 TIA Portal 설치 DVD 에 추가 정보 참조).
- 2 TIA Portal 의 SIMATIC STEP 7 Basic 소프트웨어 – V14 SP1 기준
- 3 SIMATIC S7-1200 컨트롤러, 예: CPU 1214C DC/DC/DC 및 ANALOG OUTPUT SB1232
시그널 보드, 1 AO – V4.2.1 기준 펌웨어

참조: 디지털 입력 및 아날로그 입력과 출력은 제어판으로 전달되어야 합니다.
- 4 엔지니어링 스테이션과 컨트롤러 간의 이서넷 연결



4 폐쇄 루프 제어 이론

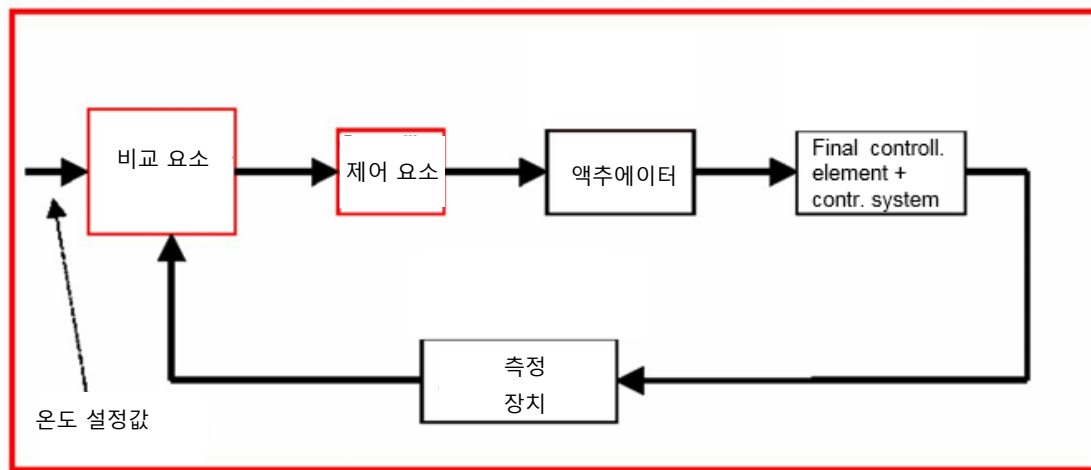
4.1 폐쇄 루프 제어 태스크

폐쇄 루프 제어는 변수 측정 값을 토대로 개입을 하여 변수의 값을 연속적으로 생성 및 유지하는 프로세스입니다.

프로세스는 변수 측정값을 토대로 실행이 되고, 변수는 프로세스 자체에 영향을 받기 때문에 이 모듈에서는 폐쇄 루프(제어 루프)에서 이루어지는 조치 경로를 만들어 보겠습니다.

제어하고자 하는 변수를 연속적으로 측정해서 같은 유형의 사전 설정된 또 다른 변수와 비교합니다. 이러한 비교 결과에 따라 제어하고자 하는 변수가 사전 설정된 변수의 값에 맞게 조정됩니다.

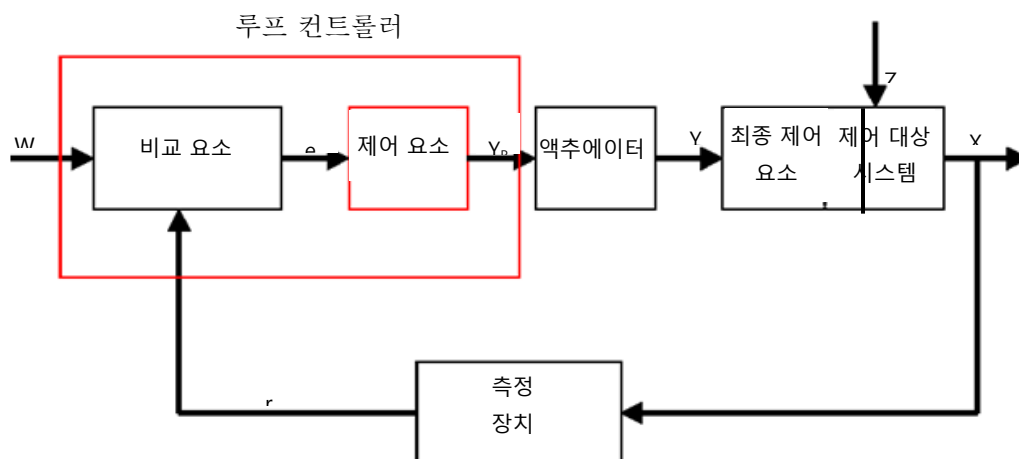
폐쇄 루프 제어를 보여주는 다이어그램



4.2 제어 루프의 구성 요소

폐쇄 루프 제어의 기본적인 개념이 아래에 자세히 설명되어 있습니다.

시작을 돕기 위한 다이어그램을 토대로 한 개요가 여기 나와 있습니다.



1. 제어 대상 변수(Controlled variable) x

폐쇄 루프 제어의 실제 "목표(target)"로, 제어에 영향을 받거나 일정하게 유지되는 변수입니다. 이 예에서, 이는 실내 온도가 될 수 있습니다. 특정 시점에서 제어 대상 변수의 순시 값을 이 시점에서의 "실제값"이라고 합니다.

2. 피드백 변수(Feedback variable) r

제어 루프에서는 원치 않는 변경에 응답할 수 있도록 제어 대상 변수를 연속적으로 확인합니다. 제어 대상 변수에 비례한 측정량을 피드백 변수라고 합니다. "난방시스템" 예에서, 내부 온도계의 측정 전압이 여기에 해당할 수 있습니다.

3. 방해 변수(Disturbance variable) z

방해 변수는 원치 않는 방식으로 제어 대상 변수에 영향 미치는 변수로서, 제어 대상 변수를 현재 설정값에서 벗어나게 만듭니다. 고정 설정값 제어의 경우, 방해 변수가 존재하기 때문에 처음부터 이러한 제어가 필요합니다. 예를 들어, 난방 시스템 검사에서는 외부 온도나 실내 온도를 이상적인 값에서 벗어나게 만드는 기타 모든 변수가 여기에 해당할 수 있습니다.

4. 설정값(Setpoint) w

주어진 시간에서의 설정값은 이 시점에 제어 대상 변수가 이상적으로 가지고 있어야 하는 값입니다. 설정값은 종속 제어에서 지속적으로 변경될 수 있다는 점에 주의해야 합니다. 이 예에서, 설정값은 현재 원하는 실내 온도가 될 수 있습니다.

5. 비교 요소(Comparing element)

현재 측정된 제어 대상 변수와 비교할 피드백 변수의 순시값 비교가 이루어지는 지점입니다. 대부분의 경우, 두 변수 모두 전압으로 측정됩니다. 두 변수의 차이는 "시스템 오류" e 입니다. 이 값은 제어 요소로 전달되어 평가됩니다(아래 참조).

6. 제어 요소(Controlling element)

제어 요소는 폐쇄 루프 제어의 실질적인 핵심입니다. 제어 요소는 시스템 오류, 즉 입력변수로서 현재 설정값에서 제어 대상 변수가 어떻게, 얼마나 많이 벗어났는지에 대한 정보를 평가하고, 이로부터 궁극적으로 제어 대상 변수에 영향을 미치는 "컨트롤러 출력 변수" y_R 을 얻습니다. 난방 시스템 예에서는, 믹서 모터의 전압이 컨트롤러 출력 변수가 될 수 있습니다.

제어 요소가 시스템 오류로부터 컨트롤러 출력 변수를 결정하는 방식이 폐쇄 루프 제어의 주요 기준이 됩니다.

7. 액추에이터(Actuator)

액추에이터는 폐쇄 루프 제어의 "실행 기관"이라 할 수 있습니다. 컨트롤러 출력 변수의 형태로 제어 대상 변수가 어떤 영향을 받는지 나타내는 정보를 제어 요소로부터 수신하여 이를 "조작 변수(manipulated variable)"로 변환합니다. 이 예에서, 이는 믹서 모터 컨트롤러가 될 수 있습니다.

8. 최종 제어 요소(Final controlling element)

조작 변수 y 의 평선으로서 제어 대상 변수에 (다소 직접적으로) 영향을 주는 제어 루프의 요소입니다. 이 예에서는, 믹서, 난방 라인 및 라디에이터의 조합이 여기에 해당할 수 있습니다. 믹서(조작 변수)의 조절은 믹서 모터(액추에이터)에 의해 이루어지며, 수온을 통해 실내 온도에 영향을 미칩니다.

9. 제어 대상 시스템(Controlled system)

제어 대상 시스템은 제어할 변수를 포함하고 있는 시스템으로서, 난방 시스템 예에서 거주 공간이 여기에 해당됩니다.

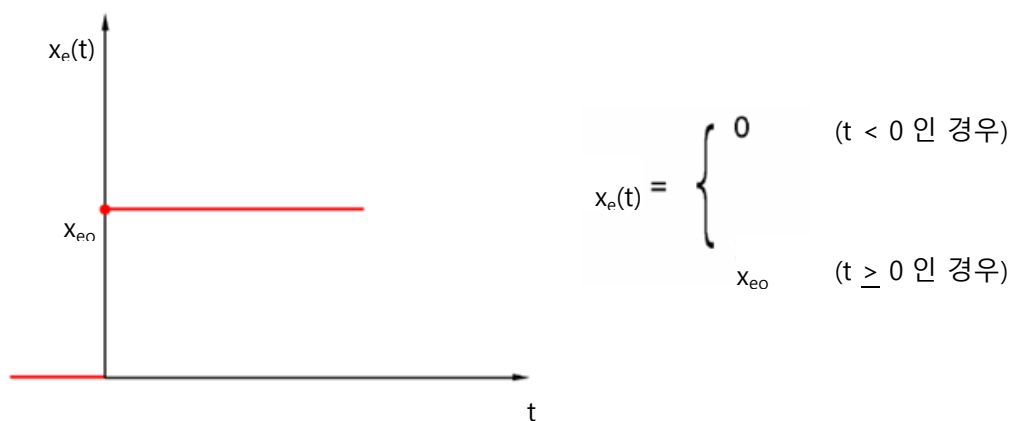
10. 정지 시간(Dead time)

정지 시간이란 컨트롤러 출력 변수가 변경되고 제어 대상 시스템에 측정 가능한 응답이 나타날 때까지의 경과 시간을 뜻합니다. 이 예에서는, 믹서 모터의 전압 변화와 이로 인한 실내 온도의 측정 가능한 변화 사이의 시간일 수 있습니다.

4.3 제어 대상 시스템을 분석하기 위한 스텝 평선

제어 대상 시스템, 컨트롤러 및 제어 루프의 응답을 분석하기 위해 입력 신호에 대해 통일된 형식의 평선, 즉 스텝 평선이 사용됩니다.

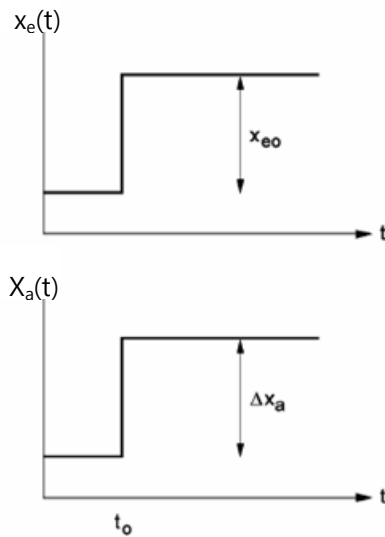
제어 루프 요소를 분석하느냐 전체 제어 루프를 분석하느냐에 따라 제어 대상 변수 $x(t)$, 조작 변수 $y(t)$, 참조 변수 $w(t)$ 또는 방해 변수 $z(t)$ 를 스텝 평선으로 지정할 수 있습니다. 흔히 입력 신호에는 $x_e(t)$ 가, 출력 신호에는 $x_a(t)$ 가 지정됩니다.



4.4 자기 조절(Self-regulation)이 되는 제어 대상 시스템

4.4.1 시간 지연 없는 비례 시스템

이러한 제어 대상 시스템을 간단히 P 시스템이라고 합니다.



t_0 에서 입력 변수 i 의 갑작스러운 변경

제어 대상 변수/조작 변수:

$$x = K_{ss} \cdot y$$

K_{ss} : 조작 변수 변경을 위한 비례 계수:

$$K_{ss} = \frac{\Delta x}{\Delta y} = \tan \alpha$$

제어 변수/방해 변수:

$$x = K_{sz} \cdot z$$

K_{sz} : 방해 변수 변경을 위한 비례 값

범위:

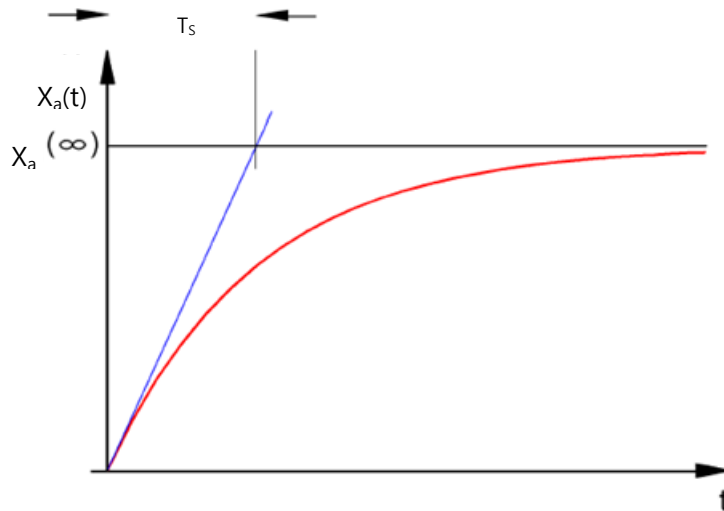
$$y_h = y_{\max} - y_{\min}$$

제어 범위:

$$x_h = x_{\max} - x_{\min}$$

4.4.2 시간 지연 있는 비례 시스템

이러한 제어 시스템을 간단히 P-T1 시스템이라고 합니다.



일반적인 입력 신호를 위한 미분 방정식 $x_e(t)$:

$$T_S \cdot \dot{x}_a(t) + x_a(t) = K_{PS} \cdot x_e(t)$$

입력 시 스텝 펄스(스텝 응답)을 위한 미분 방정식 해법

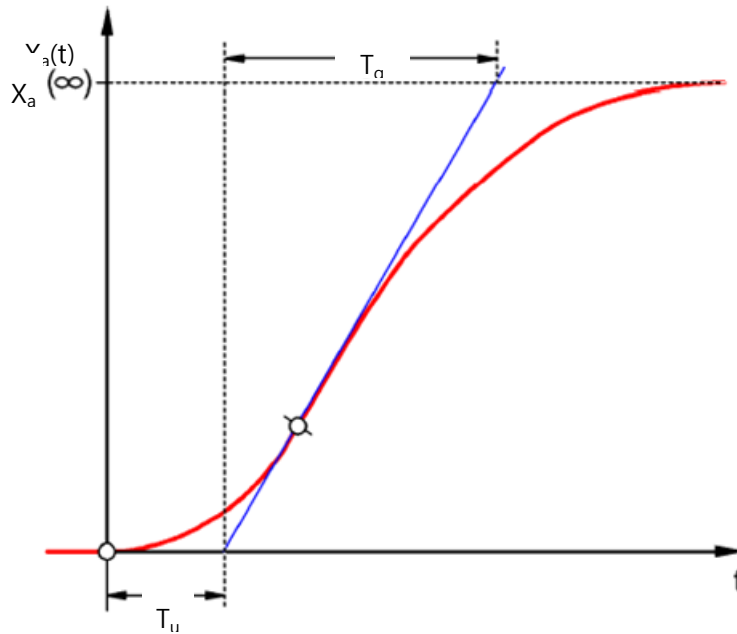
$$x_a(t) = K_{PS} (1 - e^{-t/T_S}) \cdot x_{eo}$$

$$x_a(t = \infty) = K_{PS} \cdot x_{eo}$$

T_S : 시상수

4.4.3 시간 지연이 2 회 발생하는 비례 시스템

이러한 시스템을 간단히 P-T2 시스템이라고 합니다.



Tu: 지연 시간 **Tg:** 보상 시간

시간 상수 TS1 및 TS2 를 갖는 두 P-T1 시스템을 반응으로부터 자유롭도록 직렬 연결하여 시스템이 생성됩니다.

P-Tn 시스템의 제어 가능성:

$$\frac{T_u}{T_n} < \frac{1}{10} \rightarrow \boxed{\text{손쉽게 제어 가능}} \quad \frac{T_u}{T_n} \approx \frac{1}{6} \rightarrow \boxed{\text{여전히 제어 가능}} \quad \frac{T_u}{T_g} > \frac{1}{3} \rightarrow \boxed{\text{제어가 어려움}}$$

Tu/Tg 비율이 증가할수록 시스템 제어가 더 어려워집니다.

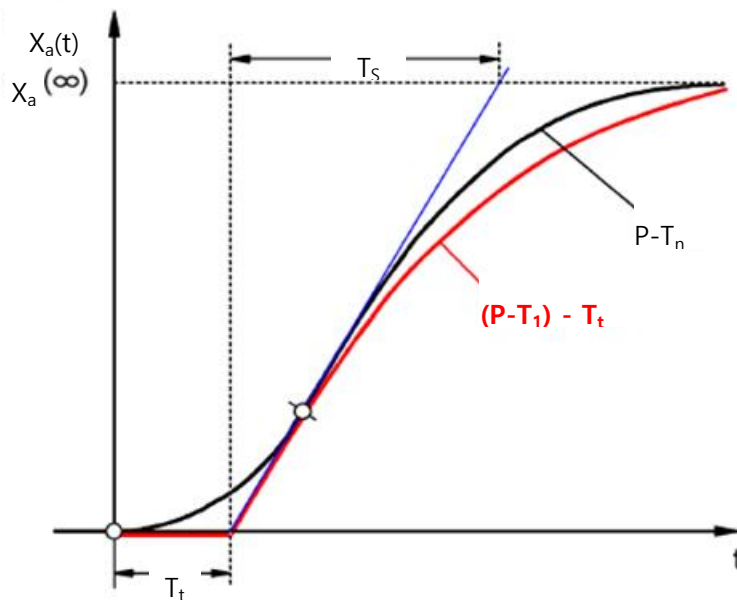
4.4.4 시간 지연이 n 회 발생하는 비례 시스템

이러한 제어 시스템을 간단히 P-Tn 시스템이라고 합니다.

시간 응답은 n 차 미분 방정식으로 설명됩니다. 스텝 응답 특성은 P-T2 시스템의 특성과 비슷합니다. 시간 응답은 T_u 및 T_g 로 설명됩니다.

대체: 지연이 많이 발생하는 시스템의 경우에는 P-T1 시스템을 정지 시간 시스템에 직렬 연결하는 것으로 대체할 수 있습니다.

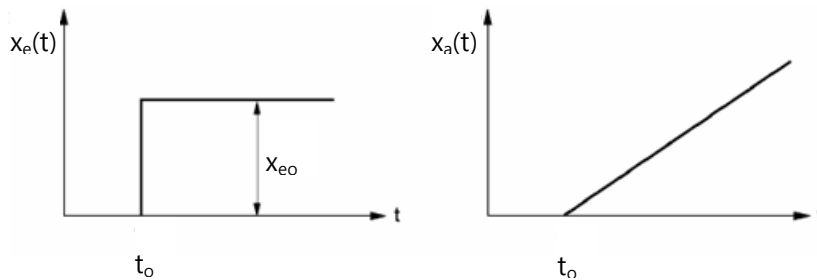
다음이 적용됩니다. $T_t \gg T_u$ 및 $T_S \gg T_g$.



4.5 자기 조절이 되지 않는 시스템

이러한 제어 시스템을 간단히 I 시스템이라고 합니다.

외란 이후에도 제어 변수가 계속 꾸준히 증가하기 때문에 고정된 최종 값을 얻기 위해 노력할 필요가 없습니다.

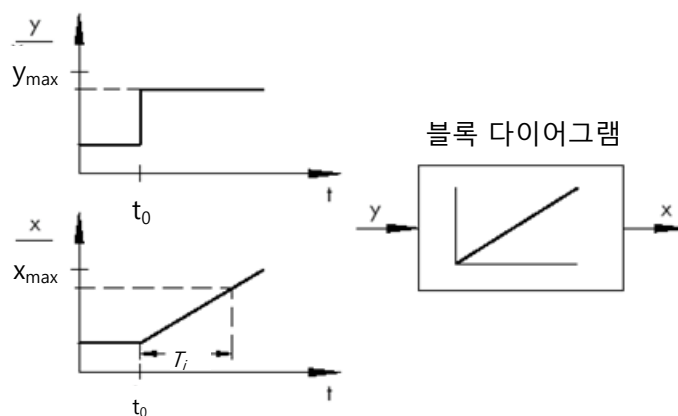


예: 수위 제어

유입 유량과 유출 유량이 동일한 배출구가 있는 탱크의 경우에는 수위가 일정합니다. 유입 유량과 유출 유량에 변화가 생기면 수위가 증가 또는 감소합니다. 유입 유량과 유출 유량 차이가 클수록 수위도 빠르게 변화합니다.

이 예에서는 실질적으로 적분 동작에 한계가 있는 경우가 대부분이라는 것을 분명하게 보여주고 있습니다. 제어 대상 변수는 시스템 고유의 한계 값에 도달할 때까지만 증가 또는 감소합니다. 예를 들어 탱크가 넘치거나 비게 되는 경우, 압력이 시스템 최대 또는 최소 값에 도달하는 경우 등에 해당합니다.

아래 그림에는 입력 변수에서의 스텝 변화에 대한 I 시스템의 시간 응답을 비롯해 도출된 블록 다이어그램이 나와 있습니다.



입력 시 스텝 평선이 $x_e(t)$ 평선으로 변경되는 경우

$$x_a(t) = K_{IS} \int x_e(t) dt \Rightarrow \text{제어 대상 시스템 적산}$$

K_{IS} : 제어 대상 시스템의 적분 계수

* SAMSON 기술 정보 - L102 컨트롤러 및 제어 대상 시스템에서 나온 그림,
버전: 2000 년 8 월 (http://www.samson.de/pdf_en/l102en.pdf)

4.6 연속 컨트롤러(Continuous controllers)의 기본 유형

1 개 또는 2 개의 조작 변수만을 켜고 끄는 작업만 하는 이산(discrete) 컨트롤러는 단순성이라는 이점이 있습니다. 두 컨트롤러 자체는 물론이고 액추에이터 및 최종 제어 요소들이 본질적으로 훨씬 간단하고 따라서 연속 컨트롤러보다 비용도 저렴합니다.

하지만 이산 컨트롤러는 몇 가지 단점도 있습니다. 그 한 가지는 대형 전기 모터나 냉각 장치 같은 부하가 큰 시스템을 작동시켜야 하는 경우 스위치가 켜진 상태일 때 부하 피크가 발생해서 전원 공급에 과부하가 걸릴 수 있다는 점입니다. 이러한 이유로 종종 "꺼짐" 및 "켜짐" 간에 전환을 하기 보다는 최대 전력("전부하")과 전력이 현저하게 낮은 액추에이터 또는 최종 제어 요소("기저 부하") 간에 전환을 하는 경우가 많습니다. 이러한 개선에도 불구하고 여전히 이산 폐쇄 루프 제어가 적합하지 않은 경우가 많습니다. 속도가 이산적으로 제어되는 자동차 엔진을 생각해 봅시다. 이 경우에는 아이들링 상태와 최대 출력 간에 아무것도 없습니다. 갑작스러운 최대 출력으로부터 발생한 힘이 타이어를 통해 노면에 적절하게 전달되는 것은 거의 불가능하다는 사실 외에도, 이러한 차량은 도로 교통에 적합하지 않을 것입니다.

따라서 이러한 용도로는 연속 컨트롤러가 사용됩니다. 이론적으로는 시스템 오류와 컨트롤러 출력 변수 사이에 제어 요소를 설정하는 수학적 관계에는 제한이 거의 없습니다. 그러나 실제로는 3 가지 전형적인 유형으로 구분됩니다. 이에 대해서는 아래에서 보다 자세하게 설명하겠습니다.

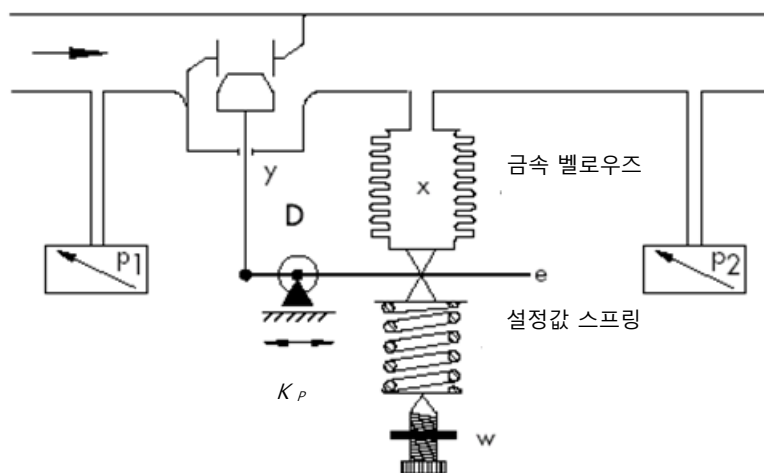
4.6.1 비례 컨트롤러(P 컨트롤러)

P 컨트롤러의 조작 변수 y 는 측정 오류 e 에 비례합니다. 따라서 P 컨트롤러는 편차 발생 시 지체 없이 반응하며 시스템 편차인 경우에만 조작 변수를 생성한다고 추론할 수 있습니다.

그림에 나와 있는 비례 압력 컨트롤러는 설정값 스프링의 FS 힘과 압력 p_2 에 의해 탄성 금속 벨로우즈에서 생성된 FB 힘을 비교합니다. 두 힘이 균형을 이루고 있지 않으면 레버가 지점 D 를 축으로 회전합니다. 이렇게 되면 밸브 플러그의 위치가 바뀌면서 새롭게 힘의 균형이 회복될 때까지 압력 p_2 가 제어됩니다.

오류 변수의 스텝 변화 이후 P 컨트롤러의 동적 작동이 그림에 나와 있습니다. 조작 변수 y 의 진폭은 오류 e 와 비례 조치 계수 K_p 에 의해 결정됩니다.

제어 편차를 가능한 작게 유지하려면 가능한 큰 값의 비례 조치 계수를 선택해야 합니다. 이 요인이 증가하면 컨트롤러의 반응 속도가 빨라지지만 값이 너무 크면 오버슈트가 발생할 위험이 있고 컨트롤러의 "헌팅" 경향성이 커집니다.

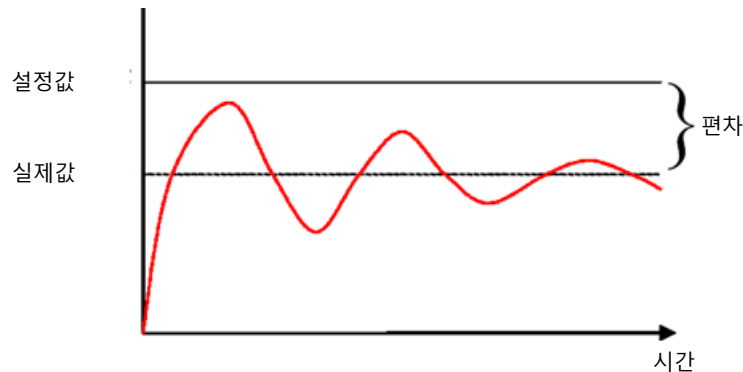


$$y = K_p \cdot e$$

* SAMSON 기술 정보 - L102 컨트롤러 및 제어에서 나온 그림과 텍스트, 시스템, 버전: 2000 년 8 월(http://www.samson.de/pdf_en/l102en.pdf)

다이어그램에서 P 컨트롤러의 응답을 확인할 수 있습니다.

제어대상 변수



이러한 컨트롤러 유형은 단순성이라는 장점이 있지만(가장 간단한 경우로는 저항기만으로도 전자식으로 구현이 가능), 다른 한편으로는 다른 컨트롤러 유형에 비해 반응이 너무 즉각적이라는 단점도 있습니다.

P 컨트롤러의 주요한 단점은 영구적인 시스템 편차입니다. 즉, 장기간에 걸쳐서도 결코 설정값에 완전히 도달할 수 없습니다. 이러한 단점 외에도 응답 속도가 아직 최적화되지 않았기 때문에 만족할만한 수준까지 단점을 최소화하는 것이 불가능합니다. 왜냐하면 이로 인해 컨트롤러에 의한 오버슈트, 즉 과반응이 야기될 수 있기 때문입니다. 최악의 경우, 컨트롤러는 제어 변수가 조작 변수가 아니라 컨트롤러 자체에 의해 제어 대상 변수가 설정값에서 주기적으로 벗어나는 영구적 발산(oscillation)이 될 수 있습니다.

영구적인 제어 편차의 문제는 적분 컨트롤러를 추가하여 해결하는 것이 가장 좋습니다.

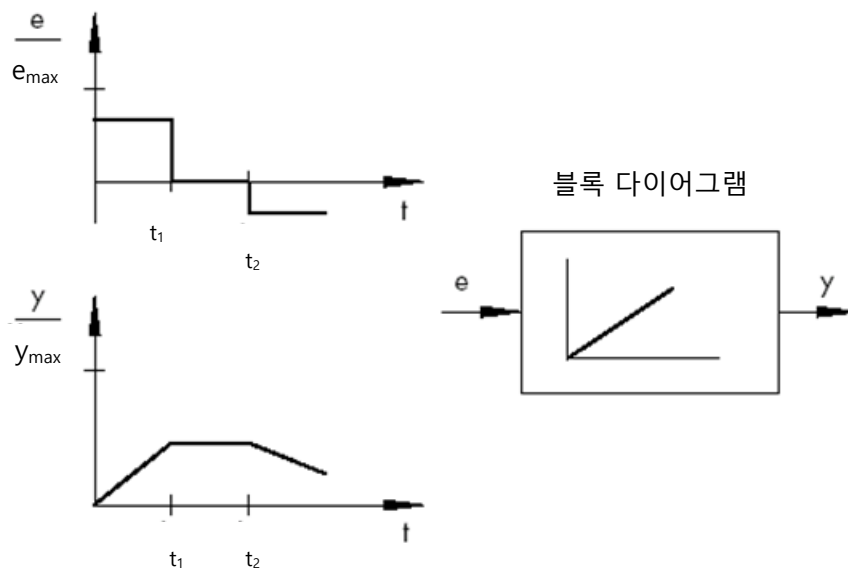
4.6.2 적분 컨트롤러(I 컨트롤러)

적분 제어 조치는 어떤 작동 지점에서는 시스템 편차를 완전히 교정하기 위해 사용됩니다. 오류가 0이 아닌 한, 적분 조치를 하면 조작 변수의 값이 변경됩니다. 참조 변수와 제어 대상 변수의 값이 똑같이 커지면서도 조작 변수가 시스템 고유의 한계 값(U_{max} , p_{max} 등)에 도달할 때만 제어 프로세스가 균형을 이룹니다.

수학에서의 적분 조치는 다음과 같이 조작 변수의 값이 오류 e 의 적분에 비례해 변경되는 것으로 표현됩니다.

$$y = K_i \int e \, dt \quad \text{여기서} \quad K_i = \frac{1}{T_n}$$

조작 변수가 얼마나 빨리 증가/감소하느냐는 오류 및 적분 시간에 따라 다릅니다.

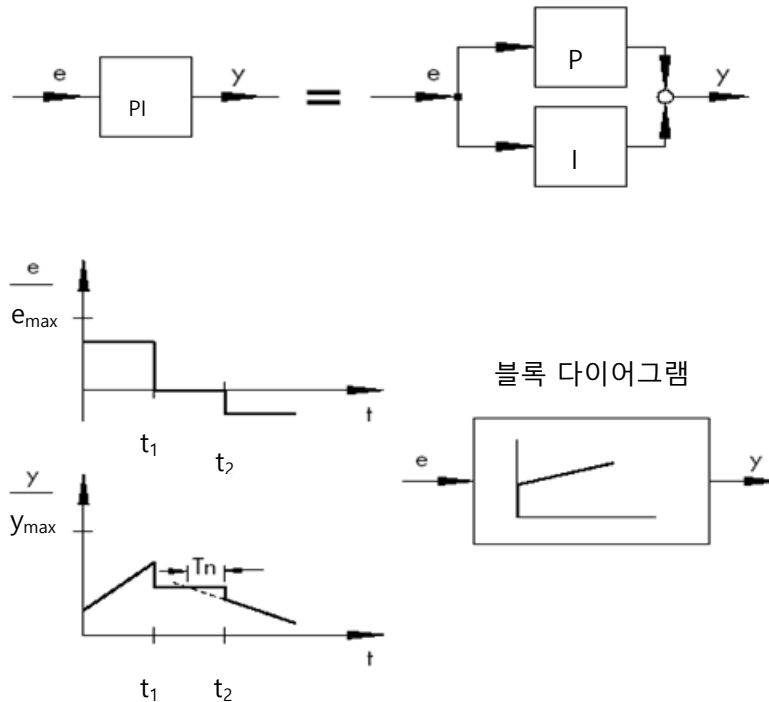


* SAMSON 기술 정보 - L102 컨트롤러 및 제어에서 나온 그림과 텍스트, 시스템, 버전: 2000 년 8 월(http://www.samson.de/pdf_en/l102en.pdf)

4.6.3 PI 컨트롤러

PI 컨트롤러가 실제로 사용되는 경우가 종종 있습니다. 이 조합에서는, 하나의 P 컨트롤러와 하나의 I 컨트롤러가 병렬로 연결됩니다.

제대로 설계될 경우, PI 컨트롤러는 두 컨트롤러 유형의 장점(안정성 및 신속성과 정상 상태 오류 없음)을 하나로 결합하며 동시에 단점을 보완합니다.



동적 작동은 비례 조치 계수 K_p 와 리셋 시간 T_n 으로 표시됩니다. 비례 구성요소 덕분에 조작 변수는 오류 신호 e 에 즉각 반응하지만, 적분 구성 요소는 일정 시간이 지난 이후에만 영향력을 갖게 됩니다. T_n 은 처음에 P 구성요소(K_p)가 생성한 것과 동일한 제어 진폭을 I 구성요소가 생성할 때까지 경과된 시간을 나타냅니다. I 컨트롤러에서와 마찬가지로 적분 조치 구성요소가 증폭될 경우 리셋 시간 T_n 을 줄여야 합니다.

컨트롤러 차원화(Controller dimensioning):

K_p 값과 T_n 값을 조정하면 제어 대상 변수의 발산을 줄일 수 있지만, 그 대신 제어의 역동성이 떨어집니다.

PI 컨트롤러의 용도: 정상 상태 오류가 발생하지 않는 신속한 제어 루프

예: 압력, 온도, 비율 제어 등

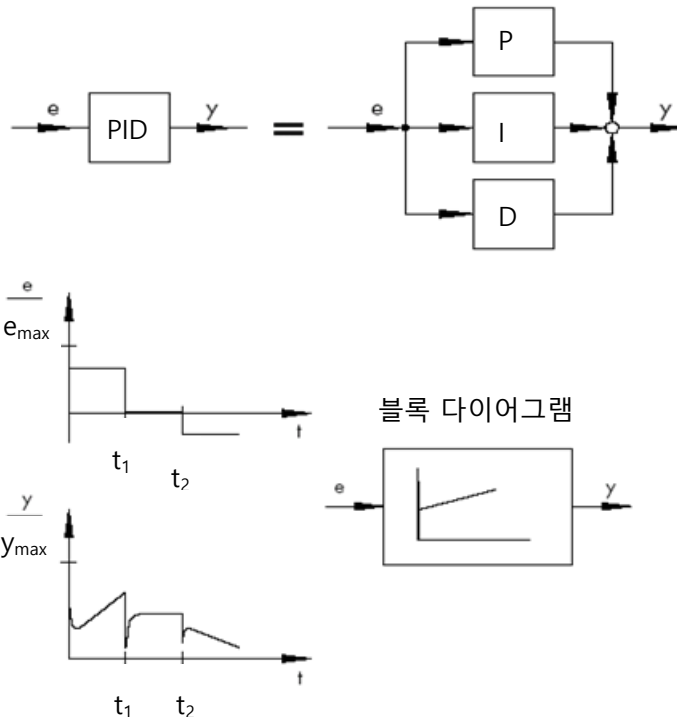
* SAMSON 기술 정보 - L102 컨트롤러 및 제어에서 나온 그림과 텍스트 시스템, 버전: 2000 년 8 월(http://www.samson.de/pdf_en/l102en.pdf)

4.6.4 미분 컨트롤러(D 컨트롤러)

D 컨트롤러는 P 컨트롤러 같이 진폭이 아니라 오류의 변화율을 토대로 조작 변수를 생성합니다. 따라서, P 컨트롤러보다 훨씬 빠르게 반응합니다. 오류가 아무리 작더라도 진폭의 변화가 발생하면 그 즉시 미분 컨트롤러가 예측에 의해 큰 제어 진폭을 생성시킵니다. 그러나 오류의 크기와 관계없이 변화율은 0 이기 때문에 정상 상태 오류 신호를 D 컨트롤러가 인식하지 못합니다. 따라서 미분 전용 컨트롤러는 실제로 거의 사용되지 않습니다. 보통은 다른 제어 요소들과 결합해 사용이 되며, 대부분은 비례 제어와 함께 사용됩니다.

4.6.5 PID 컨트롤러

D 구성요소를 PI 컨트롤러에 추가하면 그 결과 매우 활용도가 뛰어난 PID 컨트롤러가 됩니다. PI 컨트롤러에서 추가된 D 구성요소(적절하게 조정된 경우)는 제어 대상 변수가 설정값에 보다 빨리 도달하여 보다 신속하게 안정 상태가 되도록 해줍니다.



$$y = K_p \cdot e + K_i \int e dt + K_D \frac{de}{dt} \quad \text{여기서 } K_i = \frac{K_p}{T_n}; K_D = K_p \cdot T_v$$

* SAMSON 기술 정보 - L102 컨트롤러 및 제어에서 나온 그림과 텍스트, 시스템, 버전: 2000 년 8 월(http://www.samson.de/pdf_en/l102en.pdf)

4.7 발산 테스트를 사용한 컨트롤러 튜닝

만족할 만한 제어 결과를 얻기 위해서는 적절한 컨트롤러를 선택하는 것이 매우 중요합니다. 제어 파라미터 K_p , T_n 및 T_v 를 시스템 응답에 맞게 적절하게 조정하는 것은 더욱 중요합니다. 대부분의 컨트롤러 파라미터 조정은 매우 안정적이면서도 느린 제어 루프와 역동적이지만 불규칙한 제어 응답 간에 절충을 이루고 있어 발산이 쉽게 발생하고 제어 루프가 불안정해질 수 있습니다.

비선형적 시스템의 경우 항상 동일한 작동 지점에서 작동해야 합니다. 예를 들어, 고정 설정값 제어의 경우 컨트롤러 파라미터를 이 특정 작동 지점에서 시스템 응답에 맞게 조정해야 합니다. 후속 제어(follow-up) 시스템과 같이 고정 작동 범위를 정의할 수 없는 경우 컨트롤러를 조정하여 전체 작동 범위 내에서 충분할 정도로 빠르고 안정적인 제어 결과를 보장할 수 있도록 컨트롤러를 조정해야 합니다.

실제로 컨트롤러는 보통 경험을 통해 얻은 값을 기반으로 튜닝됩니다.

그러나 이러한 값들을 사용할 수 없는 경우 시스템 응답은 적절한 제어 파라미터를 결정하기 위해 몇 가지 이론적 또는 실용적 튜닝 방식을 적용하고 상세하게 분석해야 합니다.

한 가지 접근 방법으로 Ziegler와 Nichols가 처음으로 제안한, 이른바 궁극적인 방법이 있습니다. 이 방법은 다양한 경우에 적용할 수 있는 간단한 튜닝 방식을 제공합니다. 그러나 이 방법은 제어 대상 변수의 지속적인 발산을 허용하는 제어 시스템에만 적용할 수 있습니다.

이 방법은 다음과 같이 진행합니다.

- 컨트롤러에서, K_p 및 T_v 에 가장 낮은 값을 T_n 에 가장 높은 값을 설정합니다(컨트롤러의 영향력이 최소).
- 제어 시스템을 원하는 작동 지점으로 수동으로 조정합니다(제어 루프 스타트업).
- 컨트롤러의 조작 변수를 수동 조정된 값으로 설정하고 자동 작동 모드로 전환합니다.
- 제어 대상 변수가 조화 발산에 이를 때까지 K_p 를 계속 증가시킵니다(X_p 감소). 가능하다면 K_p 조정 중에 제어 루프가 발산하도록 설정값에서 약간의 단계 변화를 수행해야 합니다.
- K_p 값을 중요한 비례 조치 계수 $K_{p,crit}$ 까지 낮춥니다. 필요한 경우 여러 발산 시간을 수집하고 그 평균을 계산해 전체 발산 진폭에 대한 시간 범위를 T_{crit} 로 결정합니다.
- $K_{p,crit}$ 및 T_{crit} 값에 표에 따른 값을 곱하고 컨트롤러에서 K_p , T_n 및 T_v 에 대해 결정된 값을 입력하십시오.

	K_p	T_n	T_v
P	$0.50 \times K_{p, crit.}$	-	-
PI	$0.45 \times K_{p, crit.}$	$0.85 \times T_{crit.}$	-
PID	$0.59 \times K_{p, crit.}$	$0.50 \times T_{crit.}$	$0.12 \times T_{crit.}$

* SAMSON 기술 정보 - L102 컨트롤러 및 제어에서 나온 그림과 텍스트, 시스템, 버전: 2000년 8월(http://www.samson.de/pdf_en/l102en.pdf)

4.8 T_u - T_g 근사치를 통한 컨트롤러 조정

여기에서 P-T2 시스템의 예를 사용하여 제어 대상 시스템 조정이 수행됩니다.

T_u - T_g 근사치

Ziegler-Nichols 방법 및 Chien, Hrones 및 Reswick 방법은 시스템 K_s 의 이동 계수, 지연 시간 T_u 및 균형 시간 T_g 파라미터가 시스템 단계 응답으로부터 결정되는 T_u - T_g 근사치를 기반으로 합니다.

아래 설명된 조정 규칙은 아날로그 컴퓨터 시뮬레이션을 사용한 실험 결과입니다.

P- T_N 시스템은 일명 T_u - T_g 근사치, 즉 P- T_1 - T_L 시스템을 사용한 근사치를 통해 충분히 정확하게 설명할 수 있습니다.

시작 지점은 입력 단계 높이 K 의 시스템 단계 응답입니다. 필요한 파라미터(시스템 K_s 의 전이 계수, 지연 시간 T_u 및 균형 시간 T_g)는 그림과 같이 결정됩니다.

이동 평선은 계산에 필요한 시스템 K_s 의 이동 계수를 결정할 수 있도록 최종 정상 상태 값($K \cdot K_s$)까지 측정해야 합니다.

이 방법의 주요 장점은 시스템에 대한 분석적 설명이 불가능할 때 근사치를 사용할 수 있다는 것입니다.

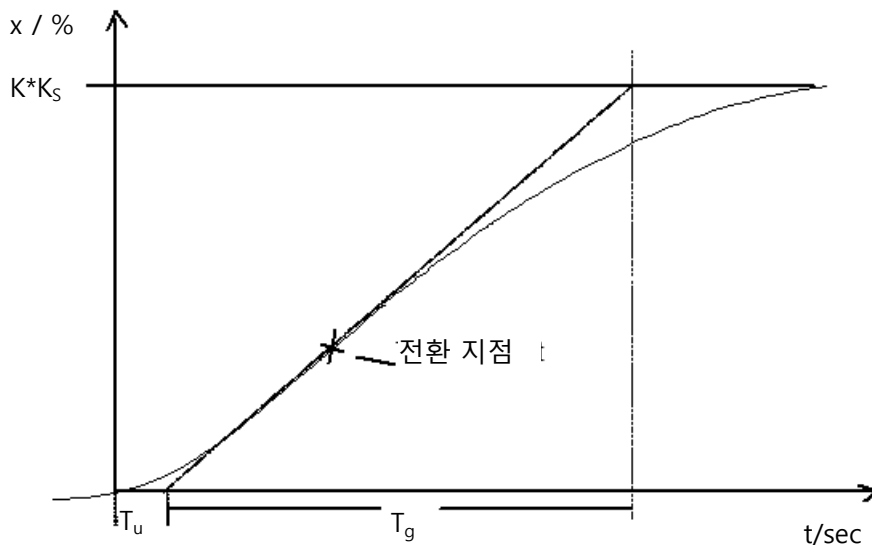


그림: T_u - T_g -근사치

4.8.1 Ziegler-Nichols 방법에 따른 PI 컨트롤러 조정

P-T₁-T_L 시스템에 대한 실험을 바탕으로 Ziegler 및 Nichols 는 고정 설정값 제어에 대해 다음과 같은 최적화된 컨트롤러 조정을 파악했습니다.

$$K_{PR} = 0,9 \frac{T_g}{K_S T_u}$$

$$T_N = 3,33 T_u$$

이러한 조정 값을 사용하면 일반적으로 방해에 확실하게 반응합니다.

4.8.2 Chien, Hrones 및 Reswick 방법에 따른 PI 컨트롤러 조정

가장 바람직한 컨트롤러 파라미터를 확보하기 위해 방해에 대한 응답 및 설정값 변화에 대한 응답을 모두 조사했습니다. 두 경우에 대해 서로 다른 값이 산출됩니다. 뿐만 아니라, 각기 다른 제어 성능 요구 사항을 충족하도록 두 가지 다른 조정이 지정되었습니다.

그 결과, 다음과 같이 조정이 이루어집니다.

- 방해에 대한 응답:

유지 기간이 가장 짧은
비주기적이고 일시적인 반응

20 % 오버슈트 최소 발산
기간

$$K_{PR} = 0,6 \frac{T_g}{K_S T_u}$$

$$K_{PR} = 0,7 \frac{T_d}{K_S T_u}$$

$$T_N = 4 T_u$$

$$T_N = 2,3 T_u$$

- 설정값 변경에 대한 응답:

유지 기간이 가장 짧은
비주기적이고 일시적인 반응

20 % 오버슈트 최소 발산
기간

$$K_{PR} = 0,35 \frac{T_g}{K_S T_u}$$

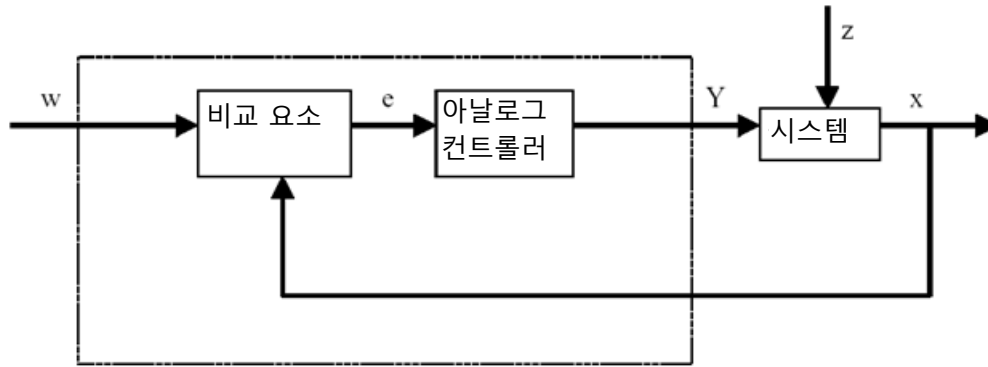
$$K_{PR} = 0,6 \frac{T_g}{K_S T_u}$$

$$T_N = 1,2 T_g$$

$$T_N = T_g$$

4.9 디지털 컨트롤러

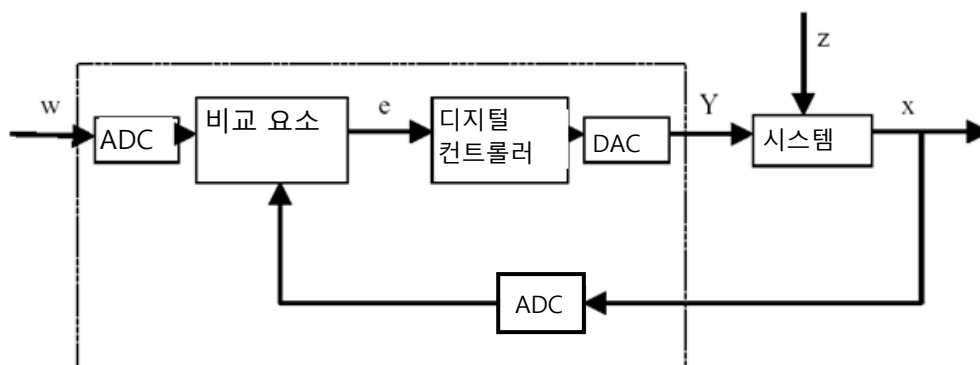
지금까지 중점을 둔 것은 아날로그 컨트롤러, 즉 아날로그 값으로 존재하는 시스템 오류를 사용하는 컨트롤러가 아날로그 방식으로 컨트롤러 출력 변수를 유도했습니다. 이런 유형의 제어 루프 다이어그램은 이제 많이 알려졌습니다.



그러나 종종 실제 시스템 오류를 디지털 방식으로 평가하는 것이 유리합니다. 하나는 아날로그 회로 형태로 구현할 때보다 시스템 오류 및 컨트롤러 출력 변수 간의 관계를 컴퓨터에서 프로그래밍할 때 사용하는 알고리즘 또는 수식으로 정의하는 것이 훨씬 유연하게 처리할 수 있습니다. 다음으로 디지털 기술을 통해 회로 통합 능력을 현저하게 향상시켜 최소한의 공간에 여러 대의 컨트롤러를 수용할 수 있습니다. 마지막으로 계산 용량이 충분할 때 계산 시간을 분할함으로써 개별 컴퓨터를 다중 제어 루프의 컨트롤러로 사용할 수 있습니다.

변수를 디지털로 처리하려면 참조 변수와 피드백 변수를 먼저 아날로그-디지털 변환기(ADC)에서 디지털 값으로 변환해야 합니다. 이렇게 하면 디지털 비교 요소에 따라 서로 차감되고 그 차이가 디지털 제어 요소로 전달됩니다. 해당 컨트롤러 출력 변수는 디지털-아날로그 변환기(DAC)에서 다시 아날로그 값으로 변환됩니다. 외부에서 변환기, 비교 요소 및 제어 요소를 결합한 장치는 아날로그 컨트롤러와 유사합니다.

다음 다이어그램을 기반으로 디지털 컨트롤러 구조를 살펴 보겠습니다.



컨트롤러의 디지털 구현으로 인한 장점에는 여러 가지 문제가 수반됩니다. 이러한 이유로 디지털 컨트롤러와 관련된 일부 변수의 크기는 폐쇄 루프 제어의 정확도가 디지털화로 인해 너무 큰 차이를 보이지 않도록 충분히 크게 선택해야 합니다.

디지털 컴퓨터의 품질 기준은 다음과 같습니다.

– 디지털-아날로그 변환기의 양자화 해상도

연속 값 범위가 디지털로 매핑되는 방법을 지정합니다. 선택한 해상도는 폐쇄 루프 제어에 중요한 세부 지점이 손실되지 않도록 충분히 높아야 합니다.

– 아날로그-디지털 변환기의 샘플링 속도

변환기에서 아날로그 값이 측정되고 디지털화되는 빈도입니다. 이는 컨트롤러가 적시에 제어 변수의 단계 변화에 계속 반응할 수 있도록 충분히 높아야 합니다.

– 사이클 시간

아날로그 폐쇄 루프 컨트롤러와는 달리 각 디지털 컴퓨터는 클록 주기로 동작합니다. 사용하는 컴퓨터 속도는 제어 변수의 중요 변경 사항이 단일 클록 주기(출력 값이 계산되고 입력 값이 조회되지 않음) 도중에 발생할 수 없도록 충분히 높아야 합니다.

디지털 컨트롤러의 성능은 아날로그 컨트롤러만큼 응답 속도가 빠르고 정확해야 합니다.

5 과제

이 챕터에서는 속도 제어를 위한 PID 컨트롤러가 "SCE_EN_031-500 Analog Values_S7-1200" 챕터의 프로그램에 추가됩니다. 이를 위해 "MOTOR_SPEEDCONTROL" [FC10] 함수 호출이 삭제되어야 합니다.

6 계획 수립

PID_Compact 기술 객체는 폐쇄 루프 제어용 TIA 포털에서 사용할 수 있습니다.

모터 속도의 폐쇄 루프 제어의 경우 이 기술 객체가 "MOTOR_SPEEDCONTROL" [FC10] 블록을 대체합니다.

이는 "031-500_Analog_Values_S7-1200" 프로젝트 확장 형태로 수행됩니다. 사전에 미리 이 프로젝트를 아카이브에서 압축을 풀어야 합니다.

"MOTOR_SPEEDCONTROL" [FC10] 함수 호출은 기술 객체가 순환 인터럽트 OB 호출 및 연결 전에 "Main" [OB1] 오거나이제이션 블록에서 삭제해야 합니다.

그런 다음 PID_Compact 기술 객체를 구성하고 시운전해야 합니다.

6.1 PID_Compact 폐쇄 루프 제어 블록

PID_Compact 기술 객체는 비례 동작 최종 제어 요소의 대한 통합 조정 기능이 있는 PID 컨트롤러를 제공합니다.

다음 작동 모드를 사용할 수 있습니다.

- 비활성
- 사전 조정
- 미세 조정
- 자동 모드
- 수동 모드
- 오류 모니터링으로 출력 값 대체

여기에서 이 컨트롤러의 연결, 파라미터 할당 및 시운전은 자동 모드에 대한 것입니다.

시운전 도중 통합 조정 알고리즘을 사용하고 제어 시스템의 제어 응답을 기록합니다.

PID_Compact 기술 객체는 항상 고정 집합 주기 시간이 50ms 인 순환 인터럽트 OB 에서 호출됩니다.

속도 설정값은 분당 회전(범위: +/-50 rpm)으로 "MOTOR_SPEEDCONTROL" [FC10] 평선의 입력에서 속도가 설정됩니다. 데이터 유형은 32 비트 부동 소수점 수(Real)입니다.

실제 속도 값 -B8(모터 +/-10V 의 센서 실제값 속도는 +/-50rpm 에 해당)이 "Input_PER" 입력에 입력됩니다.

그런 다음 컨트롤러 "Output_PER"의 출력이 신호 -U1(2 방향 모터 조작 속도값 +/-10V 는 +/-50 rpm 에 해당)과 직접 연결됩니다.

컨트롤러는 출력 -Q3(컨베이어 모터 -M1 가변 속도)가 설정된 경우에만 활성화됩니다. 이것이 설정되어 있지 않으면 "재설정" 입력 연결로 인해 컨트롤러가 비활성화됩니다.

6.2 기술 다이어그램

여기에는, 과제에 대한 기술 다이어그램이 나와 있습니다.

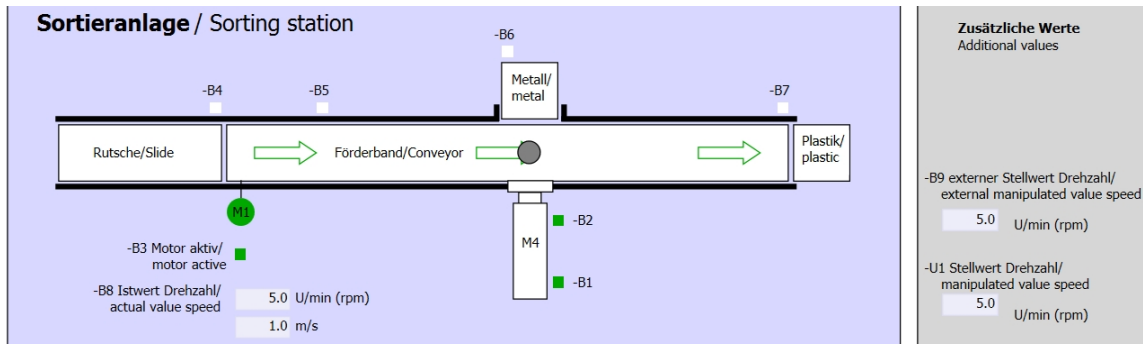


그림 1: 기술 다이어그램



그림 2: 제어 패널

6.3 참조 목록

이 과제를 위한 글로벌 오퍼랜드로서 아래와 같은 신호들이 필요합니다.

DI	유형	식별자	평선	NC/NO
I 0.0	BOOL	-A1	반환 신호 비상 정지 확인	NC
I 0.1	BOOL	-K0	메인 스위치 "ON"	NO
I 0.2	BOOL	-S0	모드 선택 수동(0)/자동(1)	수동 = 0 자동 = 1
I 0.3	BOOL	-S1	푸시버튼(자동 시작)	NO
I 0.4	BOOL	-S2	푸시버튼(자동 정지)	NC
I 0.5	BOOL	-B1	센서 실린더 -M4 복귀	NO
I 1.0	BOOL	-B4	슬라이드의 센서 부분	NO
I 1.3	BOOL	-B7	컨베이어 끝의 센서 부분	NO
IW64	BOOL	-B8	모터의 센서 실제 속도값 +/-10V 는 +/-50 rpm 에 해당	

DO	유형	식별자	평선	
Q 0.2	BOOL	-Q3	컨베이어 모터 -M1 가변 속도	
QW 64	BOOL	-U1	2 방향 모터 조작 속도값 +/-10V 는 +/-50 rpm 에 해당	

참조 목록 범례

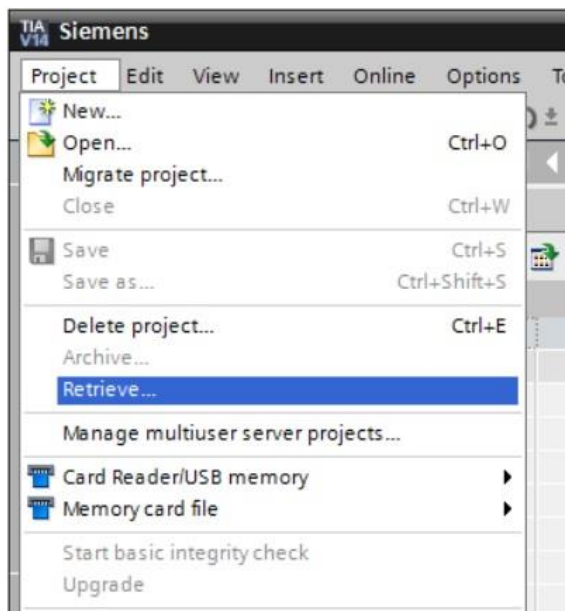
DI	디지털 입력	DO	디지털 출력
AI	아날로그 입력	AO	아날로그 출력
I	입력	Q	출력
NC	상시 닫힘		
NO	상시 열림		

7 단계별 따라 해보기

아래에는 계획을 수립하는 방법에 대한 지침이 나와 있습니다. 모든 내용을 이미 충분히 숙지했다면 숫자가 표시된 단계에만 집중하셔도 좋습니다. 그렇지 않다면, 지침의 상세 단계를 따라가면 됩니다.

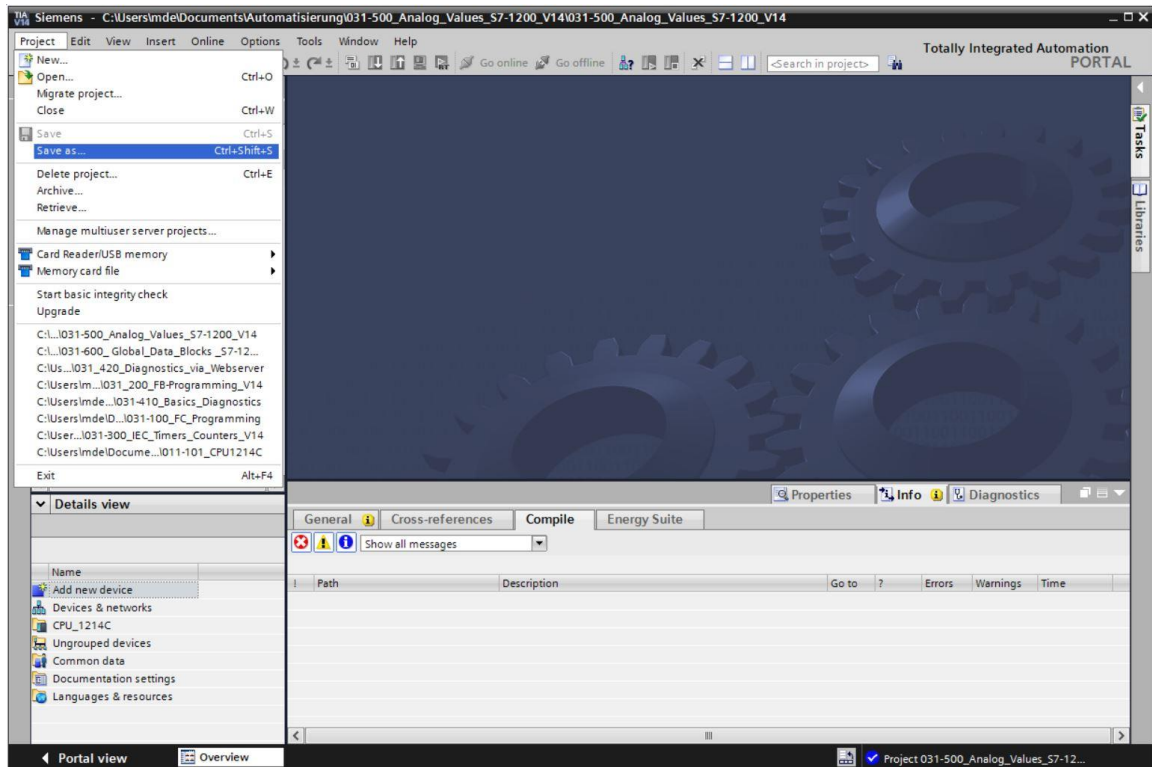
7.1 기존 프로젝트 압축 풀기

→ “SCE_EN_031-500 Analog Values_S7-1200” 장에서 “SCE_EN_031-500_Analog_Values_S7-1200.zip14” 프로젝트를 확장하기 전에 보관 위치에서 해당 프로젝트를 검색해야 합니다. 아카이브된 기존 프로젝트의 압축을 풀려면 프로젝트 뷰에서 → “프로젝트”의 → “압축 풀기”로 가서 해당되는 아카이브를 선택해야 합니다. “열기”로 선택을 확정합니다. (→ Project → Retrieve → .zap 보관 위치 선택 → Open)



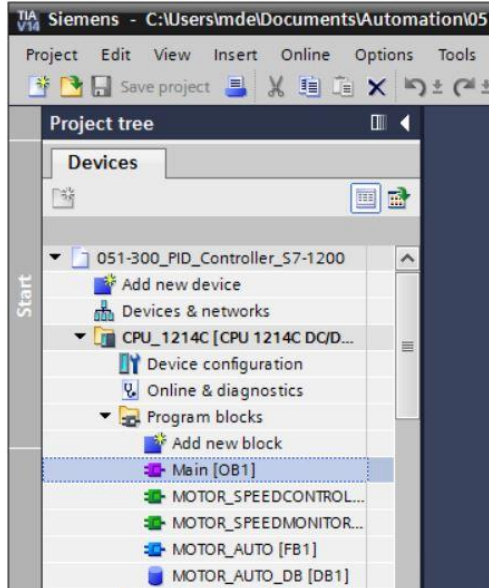
→ 그 다음으로 압축 풀기한 프로젝트가 저장될 대상 디렉토리를 선택합니다. “확인”을 클릭해 선택을 확정합니다. (→ 대상 디렉토리 → 확인)

→ 열려 있는 프로젝트를 051-300_PID_Controller_S7-1200 이름으로 저장합니다.
(→ Project → Save as... → 051-300_PID_Controller_S7-1200 → Save)

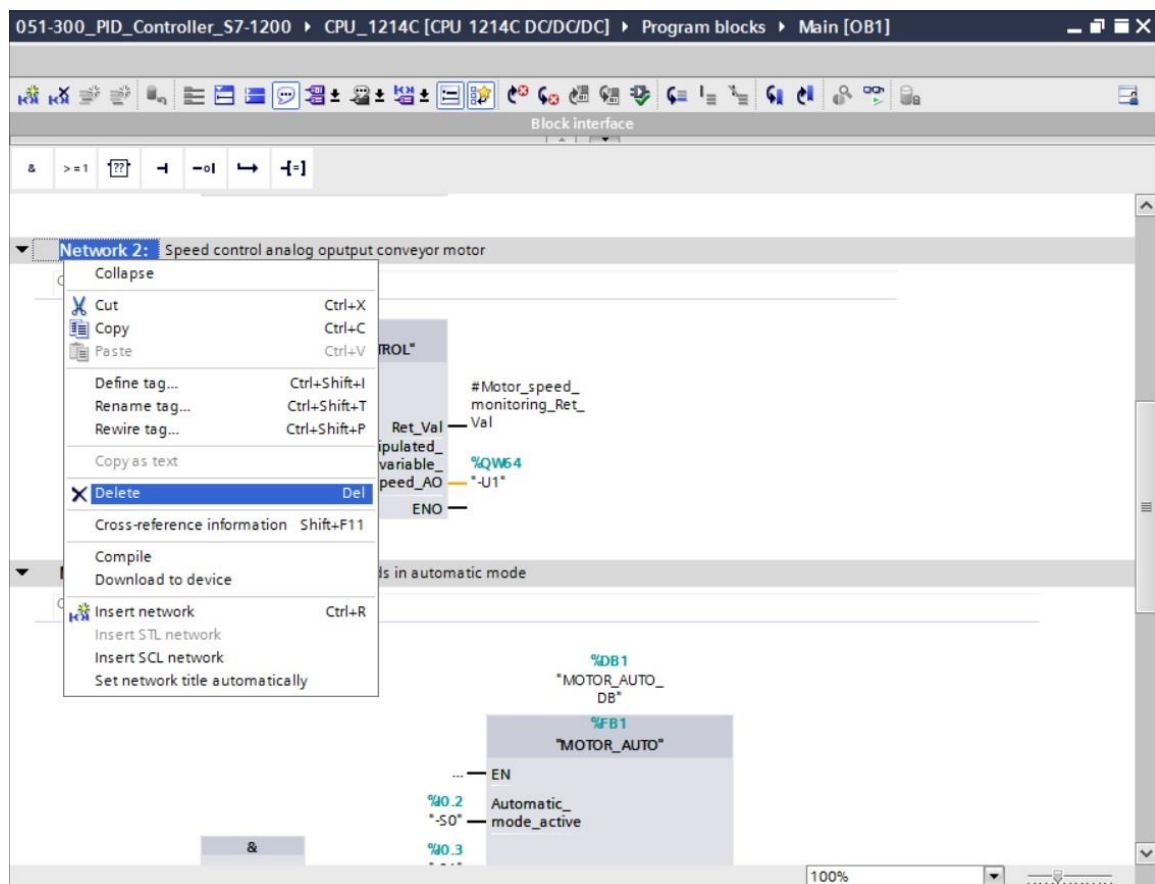


7.2 순환 인터럽트 OB 에서 PID_Compact 컨트롤러 호출

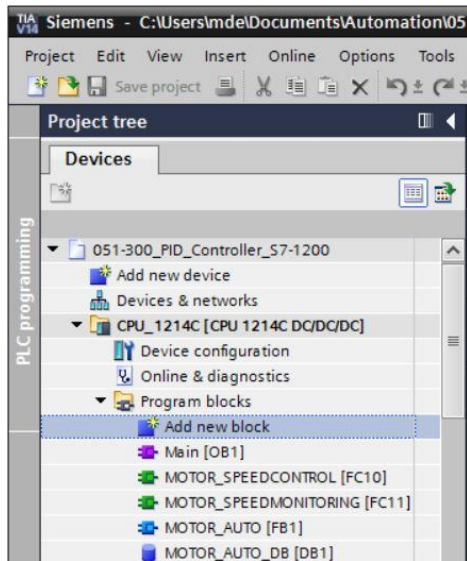
→ "Main [OB1]" 오거나이제이션 블록을 더블클릭해서 엽니다.



→ 더 이상 필요하지 않은 "MOTOR_SPEEDCONTROL" [FC10] 함수 호출로 네트워크 2 를 삭제합니다. (→ Network 2 → Delete)

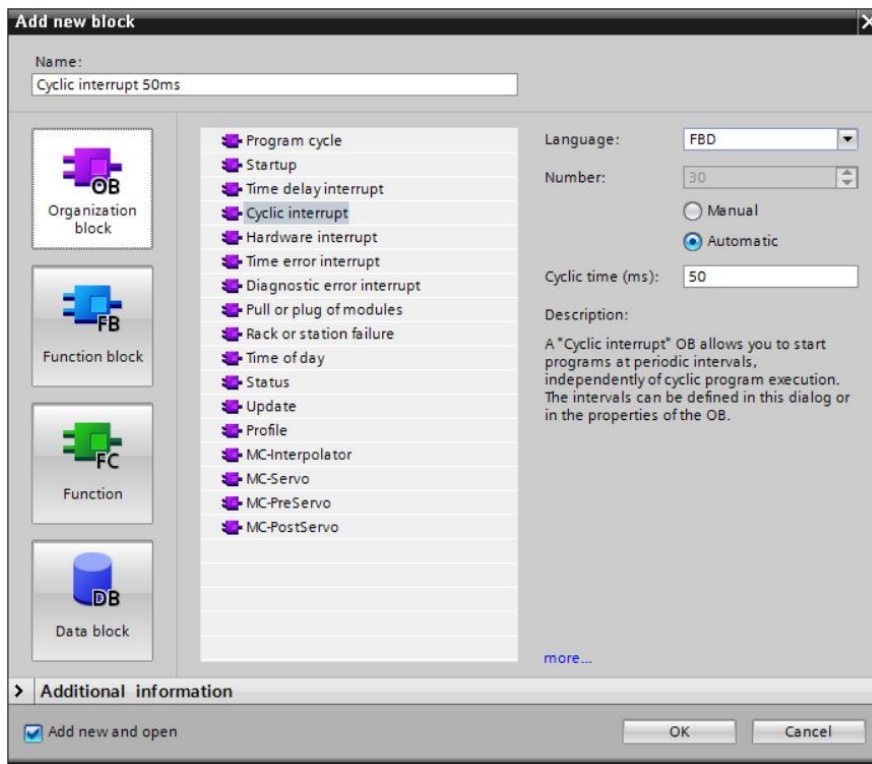


→ PID_Compact 컨트롤러 호출에 대한 순환 인터럽트 OB 가 필요합니다. 그러므로 프로그램 블록 폴더에서 '새 블록 추가' 항목을 선택하십시오. (→ Program block → Add new block)

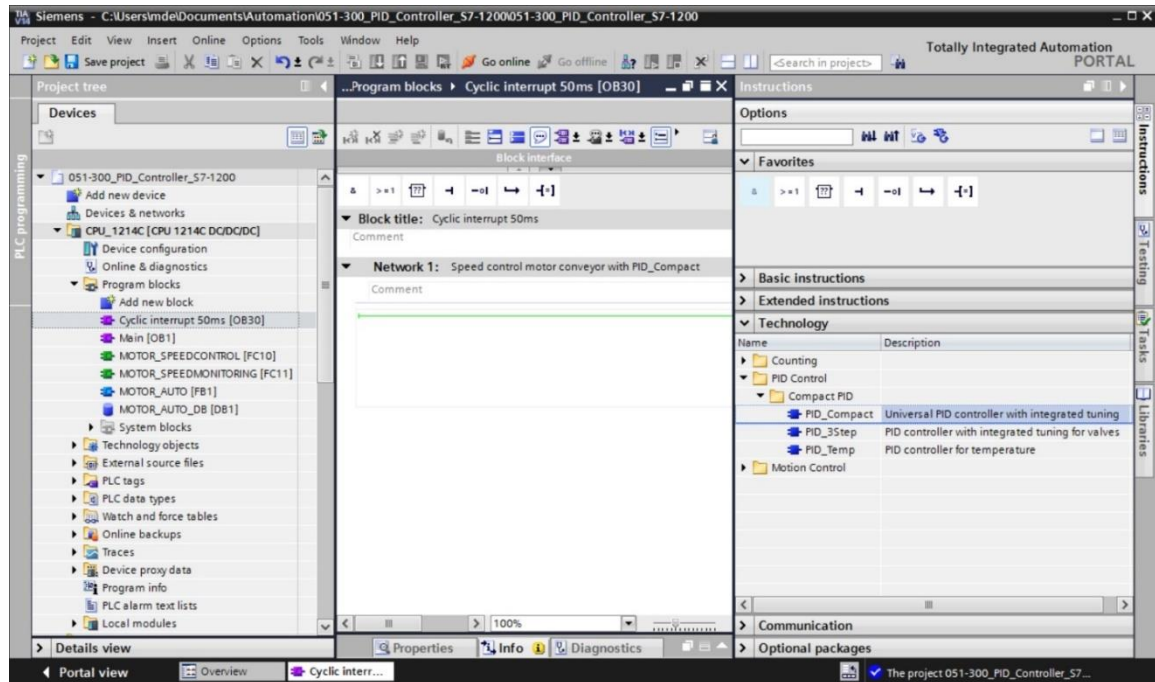


→ 다음 대화 상자에서  을 선택하고 순환 인터럽트 OB 이름을 다음으로 변경합니다. "Cyclic interrupt 50ms" 언어를 FBD 로 설정하고 순환 시간으로 "50ms"를 할당합니다. "Add new and open" 체크박스를 선택합니다. "OK"을 클릭합니다.

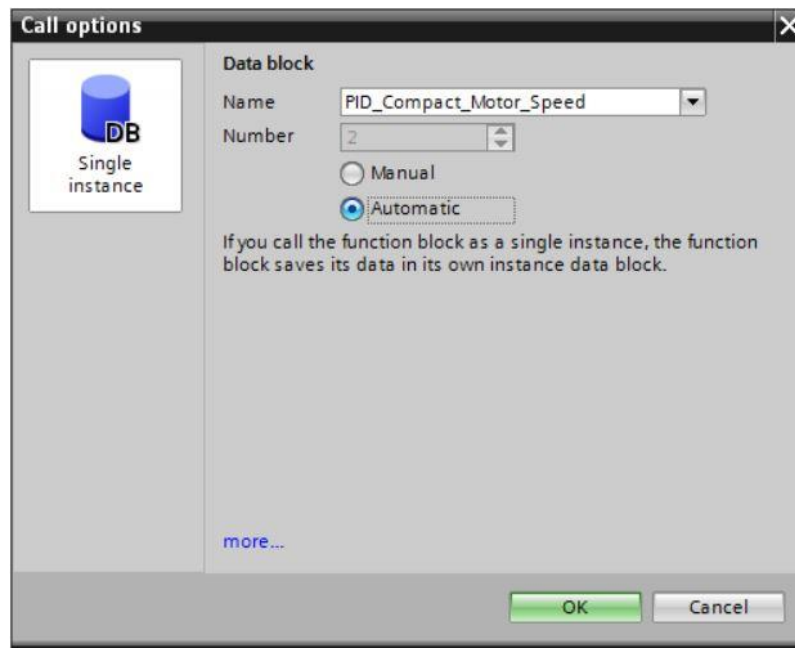
(→  → Name: Cyclic interrupt 50ms → Language: FBD → Cyclic time(ms): 50 → 
Add new and open → OK)



→ 그림 블록이 바로 열립니다. 의미 있는 설명을 입력하고 끌어다 놓기로 'PID_Compact' 기술 객체를 네트워크 1 로 이동시킵니다. (→ Technology → PID control → Compact PID → PID_Compact)

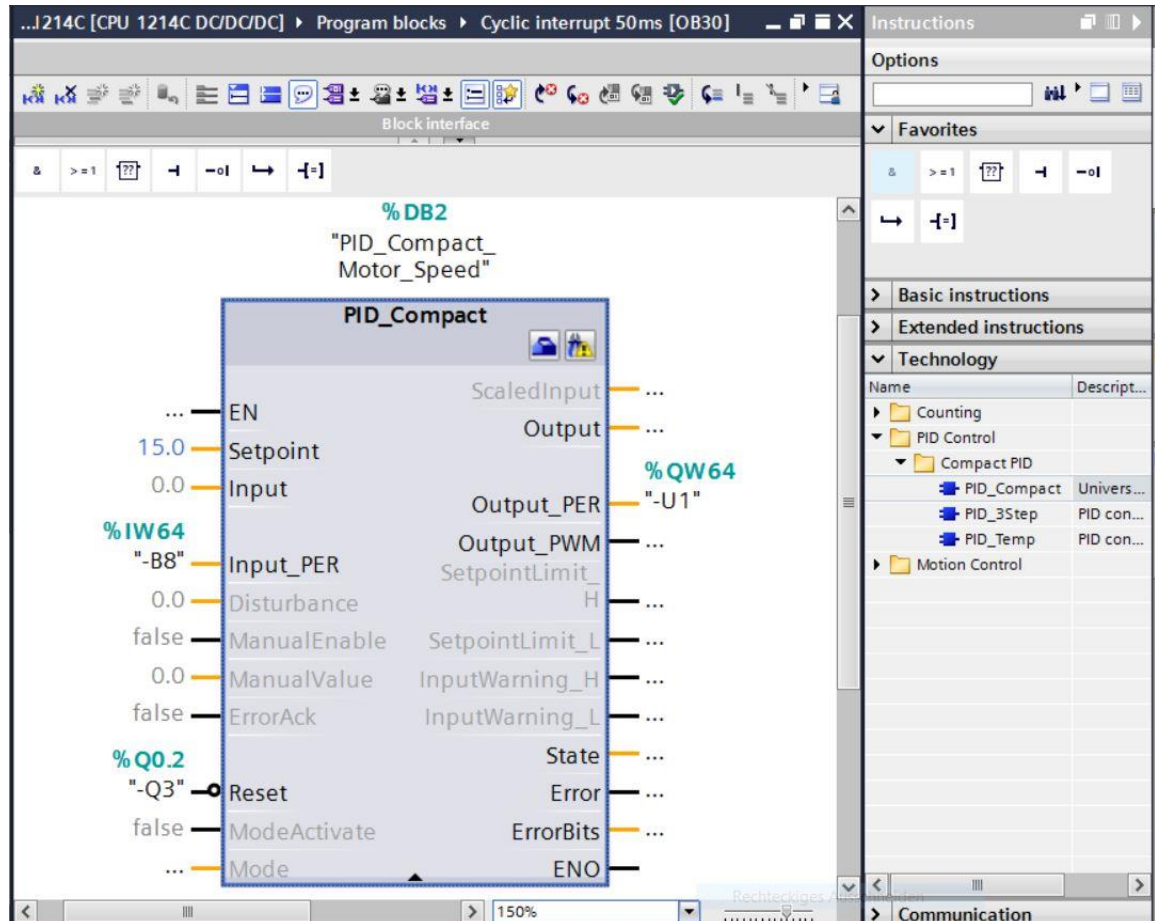


→ 인스턴스 데이터 블록에 대한 이름을 할당하고 확인으로 적용합니다. (→ PID_Compact_Motor_Speed → OK)



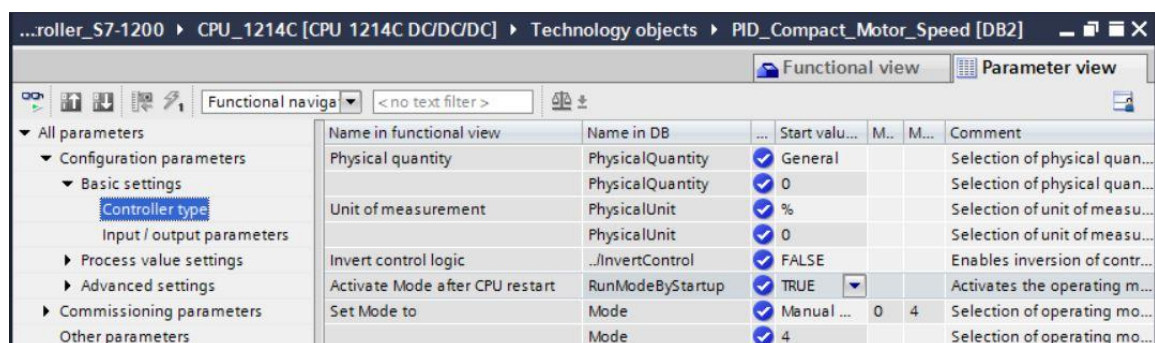
→ '▲' 화살표를 클릭하여 블록 뷰를 확장합니다. 설정값(상수: 15.0), 실제값(글로벌 태그 "-B8"), 조작 변수(글로벌 태그 "-U1") 및 컨트롤러 비활성화를 위한 재설정 입력(글로벌 태그 "-Q3")과 함께 이 블록을 상호 연결합니다. '재설정' 입력을 무효화합니다. 그런 다음 컨트롤러의 '구성 마스크'를 열 수 있습니다.

(→ ▲ → 15.0 → "-B8" → "-U1" → -Q3 → -이 →)



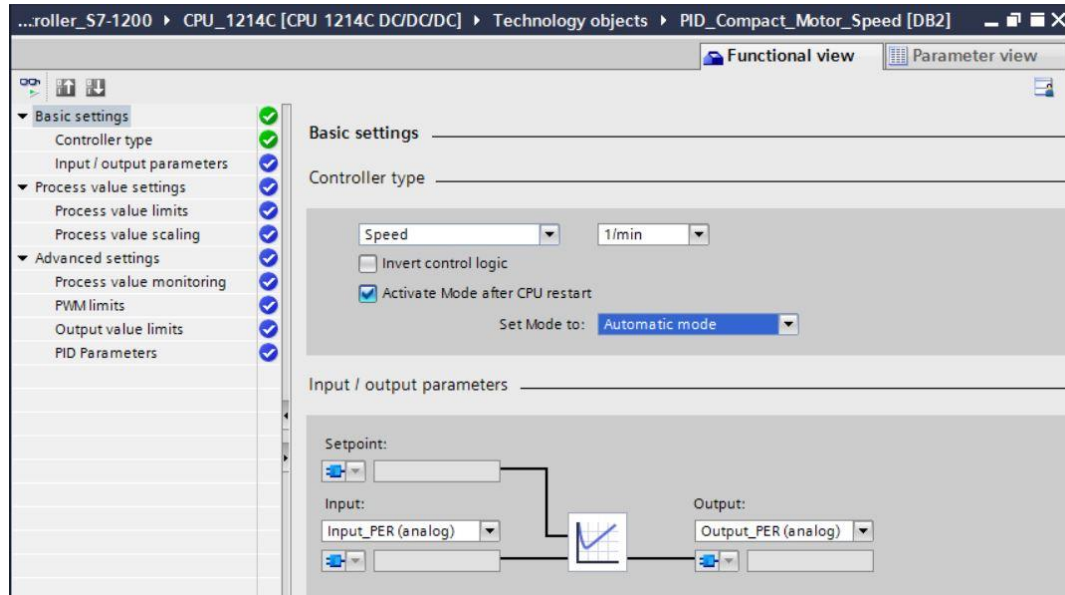
→ 컨트롤러 구성에는 두 가지 뷰, 즉 파라미터 뷰와 기능별 뷰가 있습니다. 여기서는 더 이해하기 쉬운 '기능별 뷰'를 사용하겠습니다.

(→ 기능별 뷰)



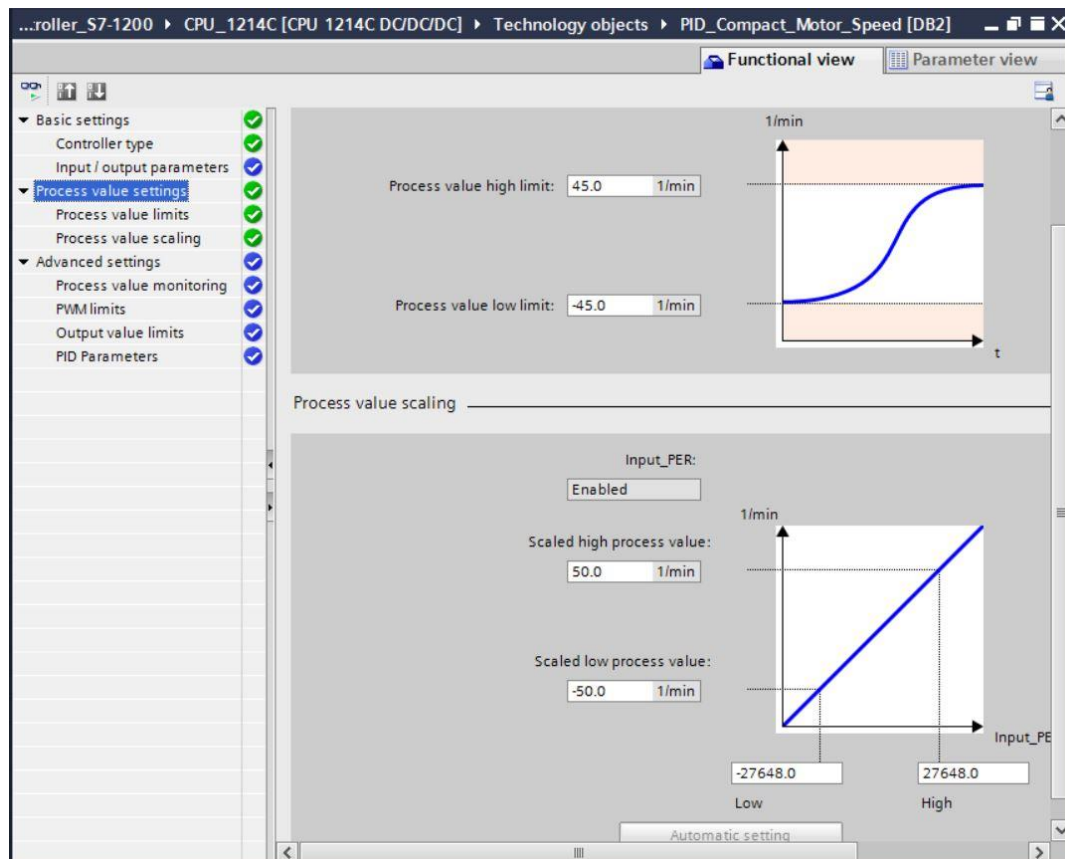
→ 'Basic settings'에서 'Controller type' 및 'Input/output parameters'의 상호 연결이 입력됩니다. 여기 표시된 대로 값을 설정합니다.

(→Basic settings→ Controller type → Input/output parameters)

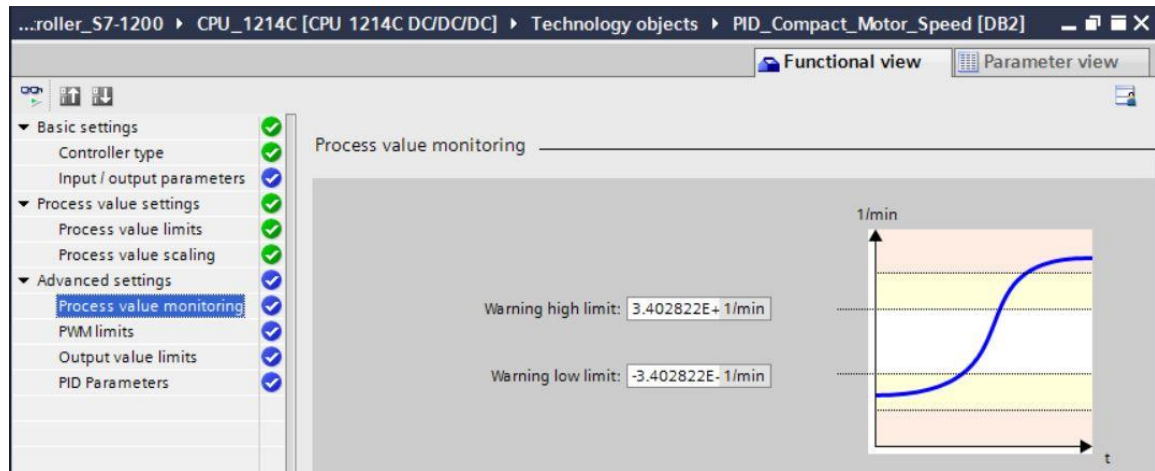


→ 'Process value setting'에서 +/-50rpm 범위로 확장하고 +/-45rpm의 '프로세스 값 제한'을 정의합니다.

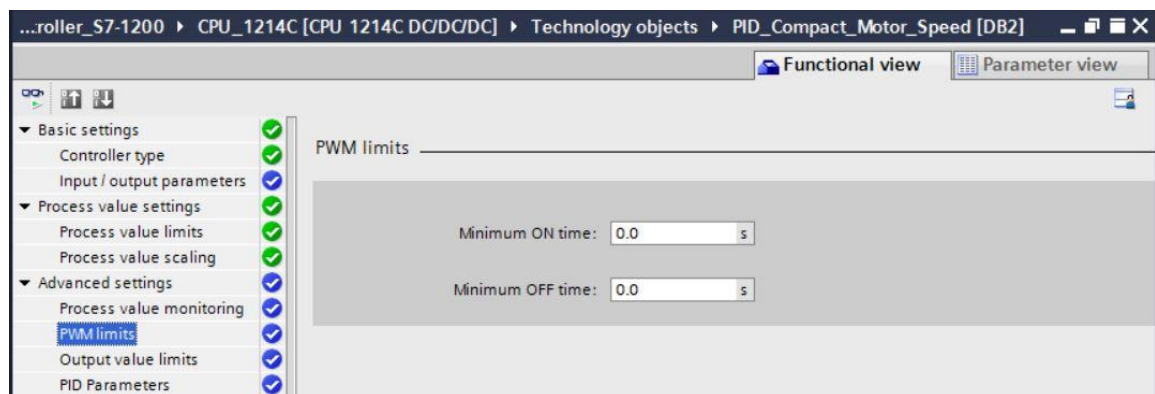
(→Process value setting → Process value limits → Process value scaling)



→ 'Advanced setting'에서 프로세스 값 모니터링이 가능하지만 여기에서는 다루지 않습니다.
(→Advanced setting→ Process value monitoring)

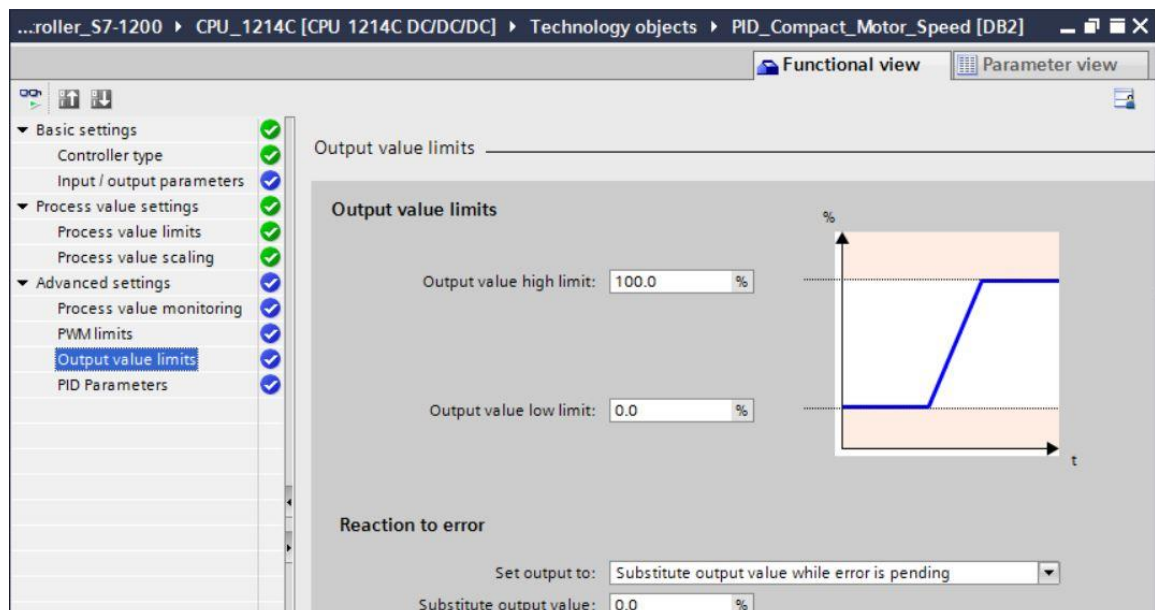


→ 'PWM'(Pulse Width Modulation)에 대한 'Advanced setting'에서 해당 출력은 프로젝트에 필요하지 않으므로 기본값을 그대로 사용합니다. (→ Advanced setting → PWM)



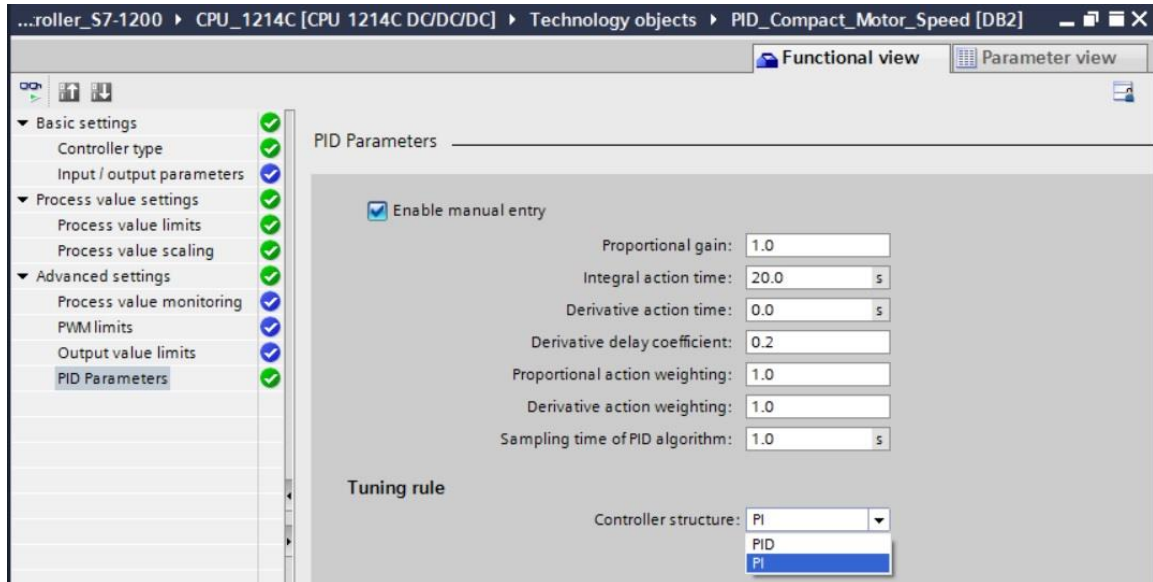
→ 'Advanced setting'에서 0.0%~100.0%의 '출력 값 제한'을 정의합니다.

(→Advanced setting → Output value limits)



→ 'Advanced setting'에서 이제 'PID Parameters'의 수동 설정을 찾을 수 있습니다. 컨트롤러 구조를 'PI'로 변경한 후 을 클릭해 구성 창을 닫으면 기능적 PID 컨트롤러가 포함된 완제품을 받게 됩니다. 이 제품은 작동 중 온라인으로 계속 시운전 및 조정 작업을 수행해야 합니다.

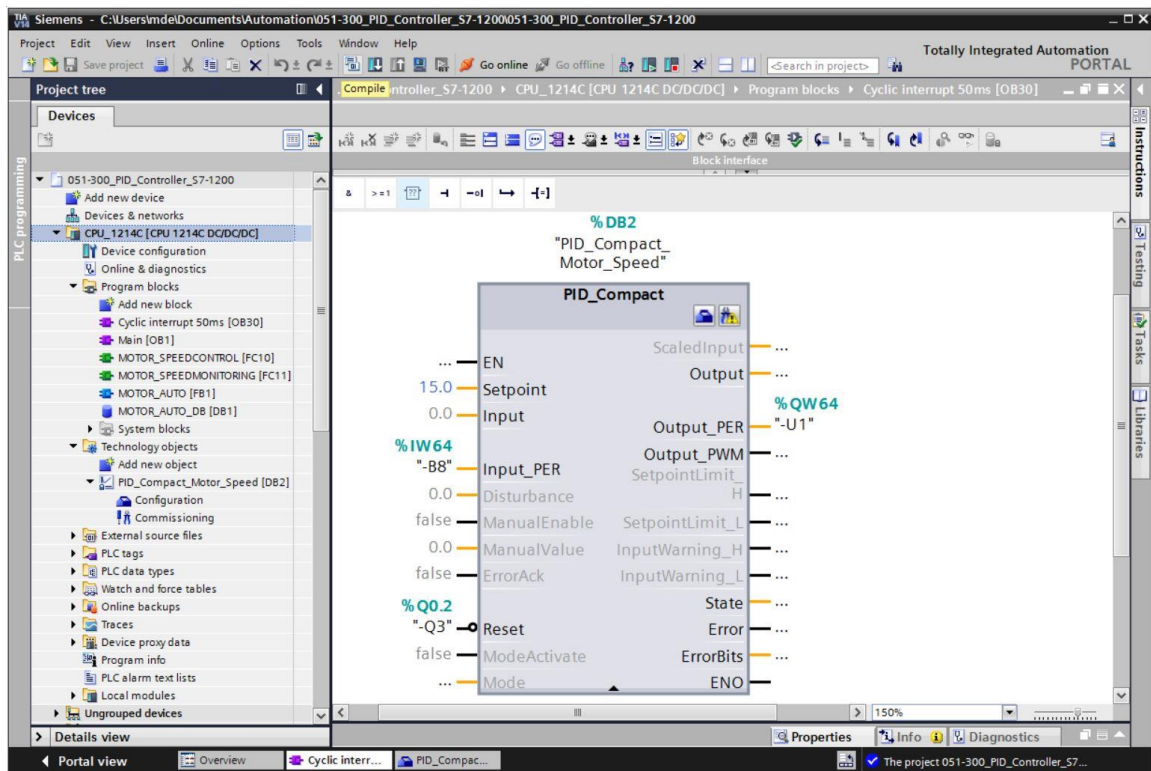
(→ Advanced setting → PID Parameters → Controller structure: PI → )



7.3 프로그램 저장 및 컴파일

→ 프로젝트를 저장하려면 메뉴에서  **Save project** 버튼을 선택합니다. 모든 블록을 컴파일하려면 “프로그램 블록” 폴더를 클릭하고 메뉴에서 컴파일을 위한 아이콘  을 선택합니다.

(→  **Save project** → Program blocks → )

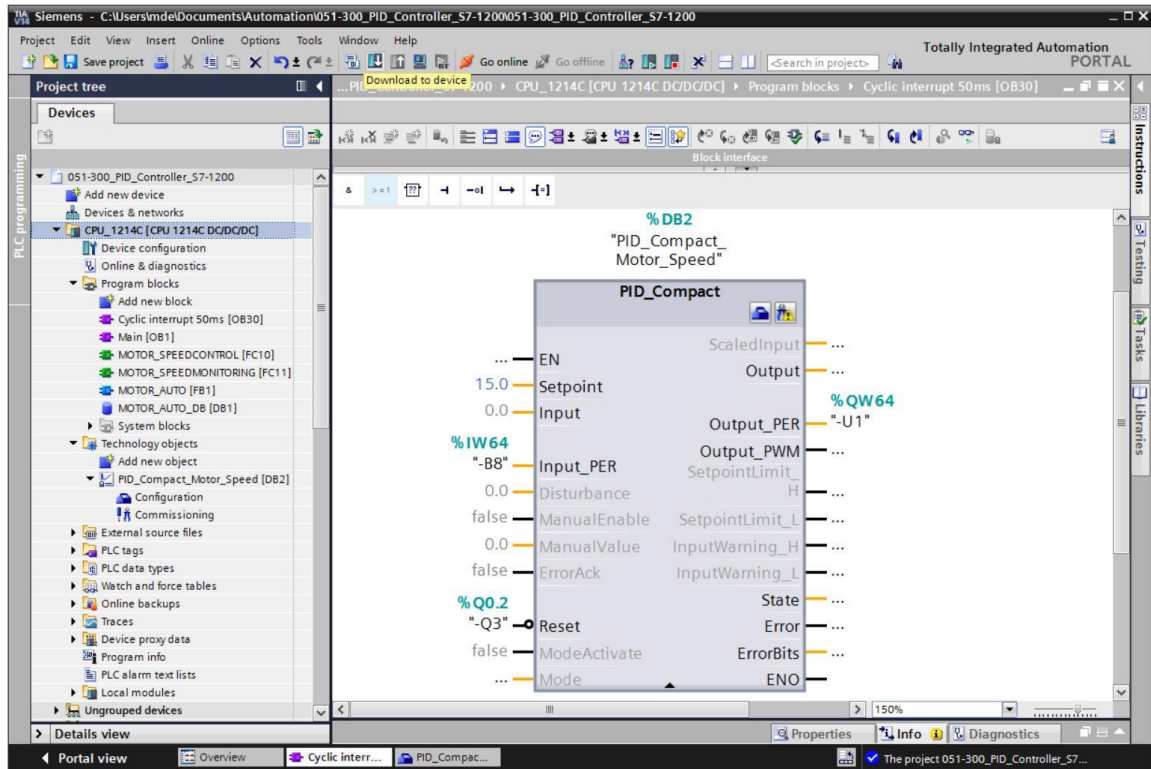


→ “Info” 아래의 “Compile” 영역에 어떤 블록이 성공적으로 컴파일되었는지가 나타납니다.

Properties Info Diagnostics							
General Cross-references Compile Energy Suite Syntax							
Show all messages							
Compiling finished (errors: 0; warnings: 2)							
Path	Description	Go to	?	Errors	Warnings	Time	
Tuning	Tuning has not been started yet.	→				2:33:09 PM	^
Program blocks	Block was successfully compiled.	→		0	0	2:33:09 PM	
Cyclic interrupt 50ms (OB30)	Block was successfully compiled.	→				2:33:09 PM	
Main (OB1)	Block was successfully compiled.	→				2:33:11 PM	
	Compiling finished (errors: 0; warnings: 2)					2:33:12 PM	

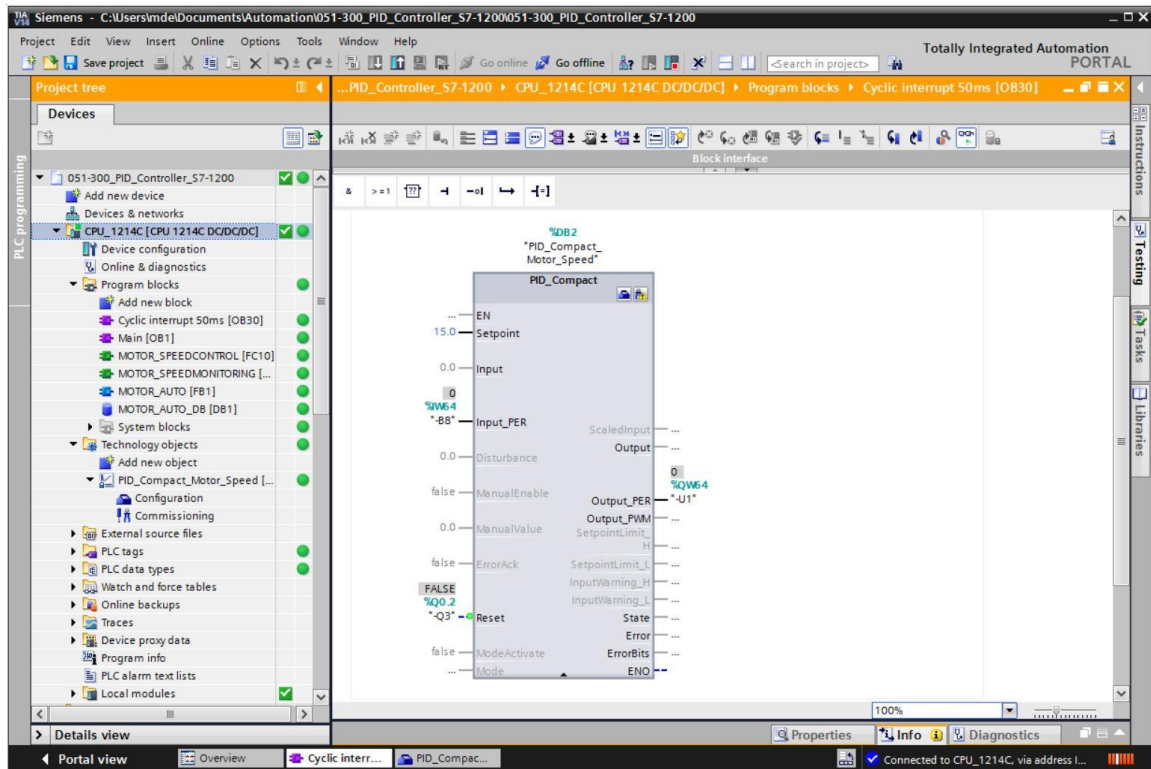
7.4 프로그램 다운로드

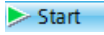
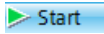
→ 컴파일이 성공적으로 완료되고 나면 앞서 모듈에서 설명한 바와 같이 하드웨어 구성을 포함하여 생성된 프로그램과 함께 전체 컨트롤러를 다운로드할 수 있습니다. (→ )

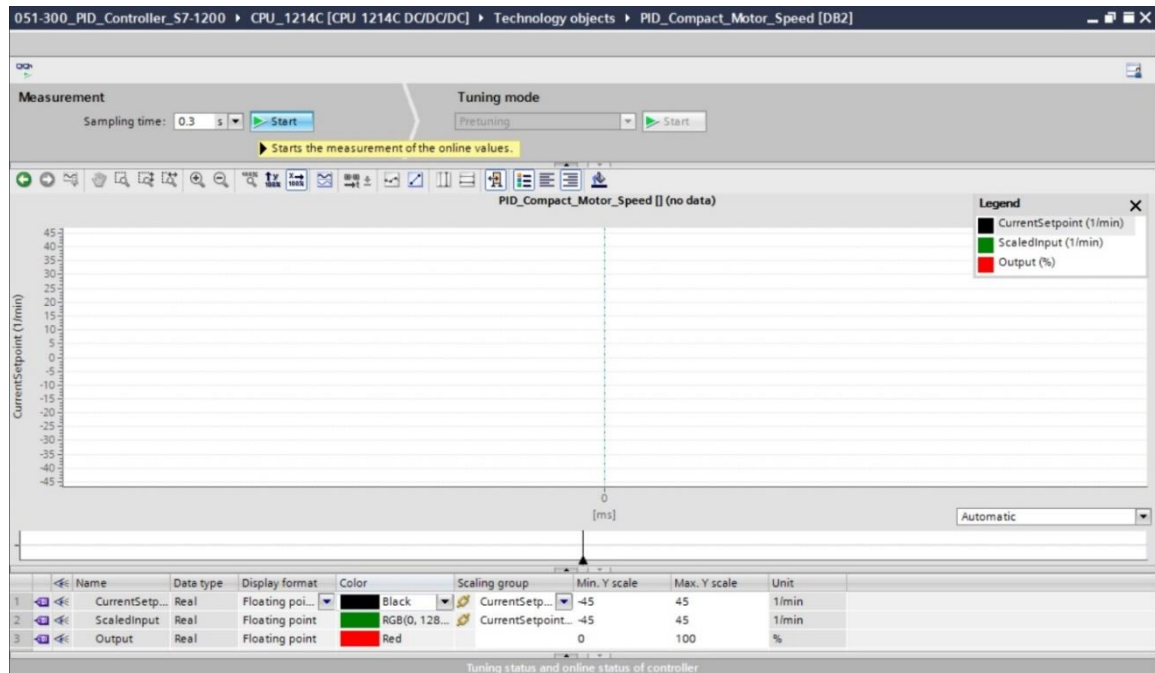


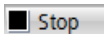
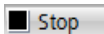
7.5 PID_Compact 모니터링

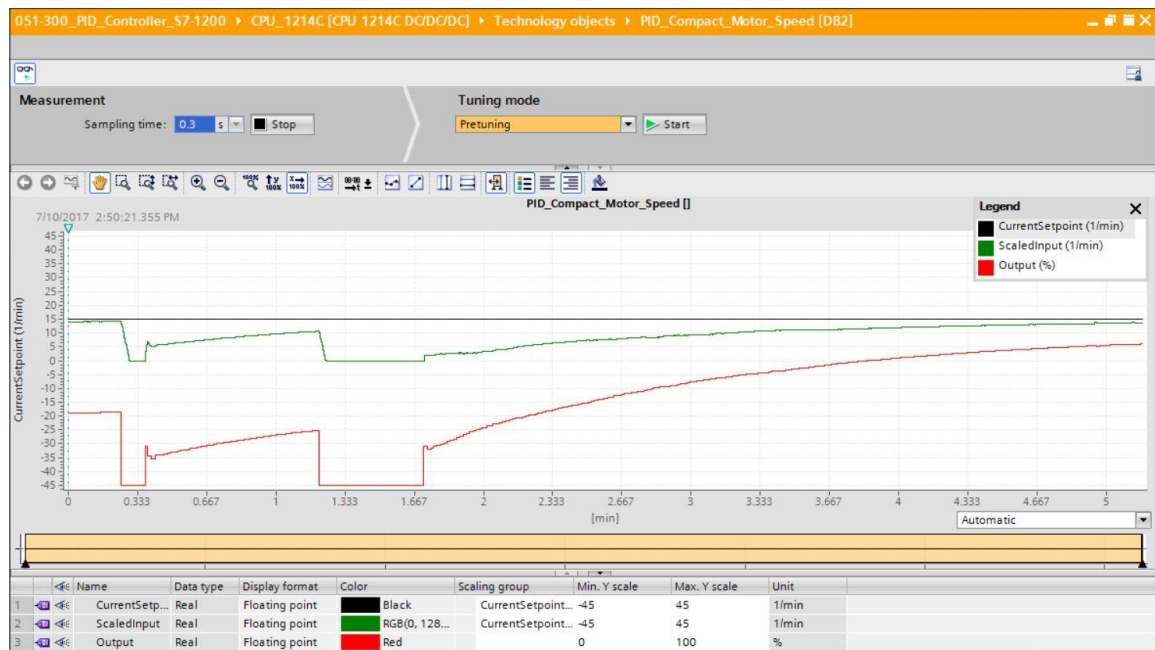
→ 프로그램을 테스트할 때 블록 및 태그 상태를 모니터링하려면 모니터링 켜기/끄기 아이콘  을 클릭합니다. 그러나 CPU 를 처음 시작할 때 'PID_Compact' 컨트롤러가 아직 조정되지 않았습니다. '  Commissioning ' 아이콘을 클릭하여 조정을 시작해야 합니다.
(→ Cyclic interrupt 50ms [OB30] →  → PID_Compact →  Commissioning)



→ 'Measurement'에서  을 클릭하면 설정값(Setpoint), 실제값(ScaledInput) 및 조작 변수(Output) 값을 다이어그램에 표시하고 모니터링할 수 있습니다. (→ )



→ 이 측정은 " Stop"을 다시 클릭하여 중단시킬 수 있습니다. (→ )



7.6 PID_Compact 사전 조정

사전 조정은 출력 값의 단계 변화에 대한 프로세스 응답을 결정하고 전환 지점을 검색합니다. PID 파라미터는 제어 시스템의 최대 기울기 및 불감 시간으로부터 계산됩니다. 사전 조정 및 미세 조정을 수행하여 최적의 PID 파라미터를 얻을 수 있습니다.

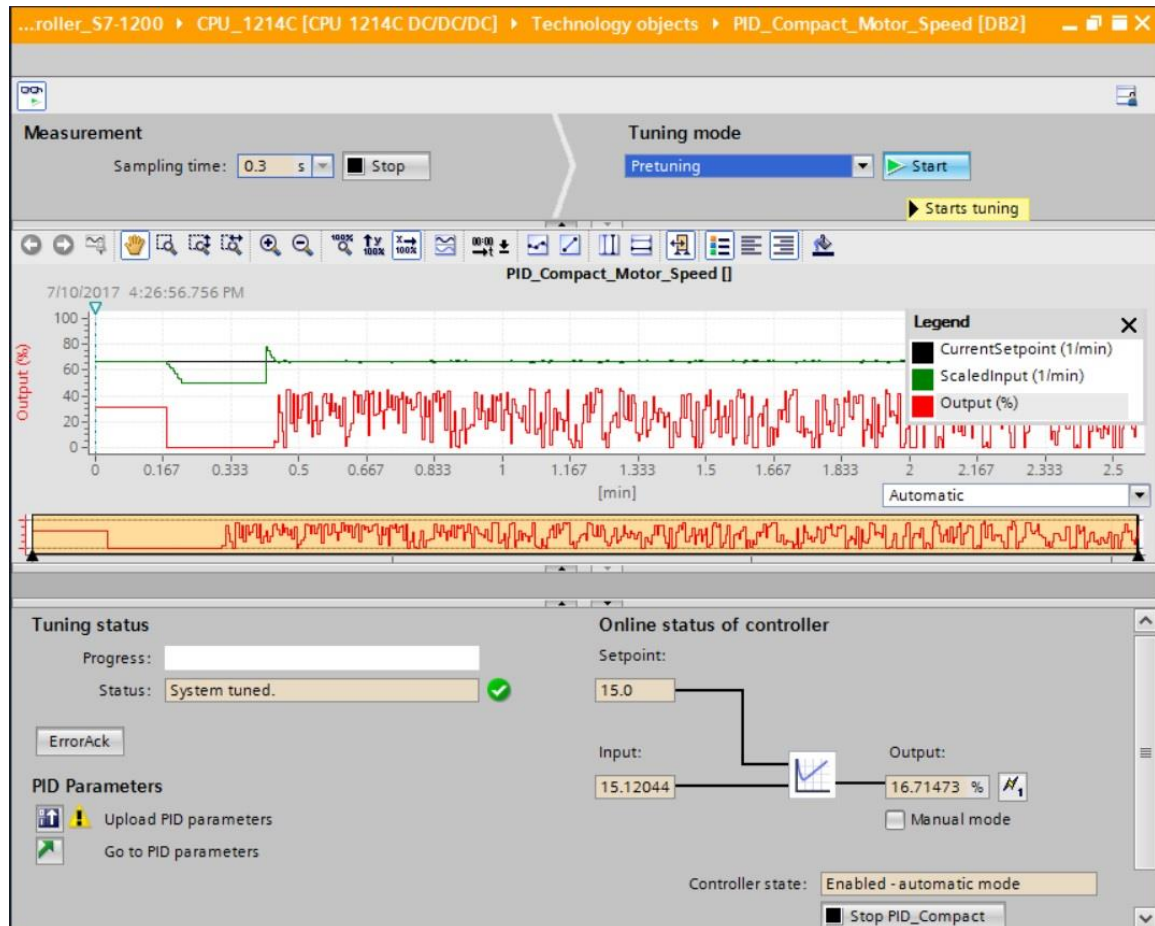
실제값의 안정성이 높아질수록 더욱 쉽고 정확하게 PID 파라미터를 결정할 수 있습니다. 실제값 상승이 잡음보다 현저히 크면 실제값 잡음이 허용됩니다. 이는 "비활성" 또는 "수동 모드" 작동 모드일 가능성이 큼니다. PID 파라미터는 다시 계산되기 전에 백업됩니다.

다음 요구 사항을 충족시켜야 합니다.

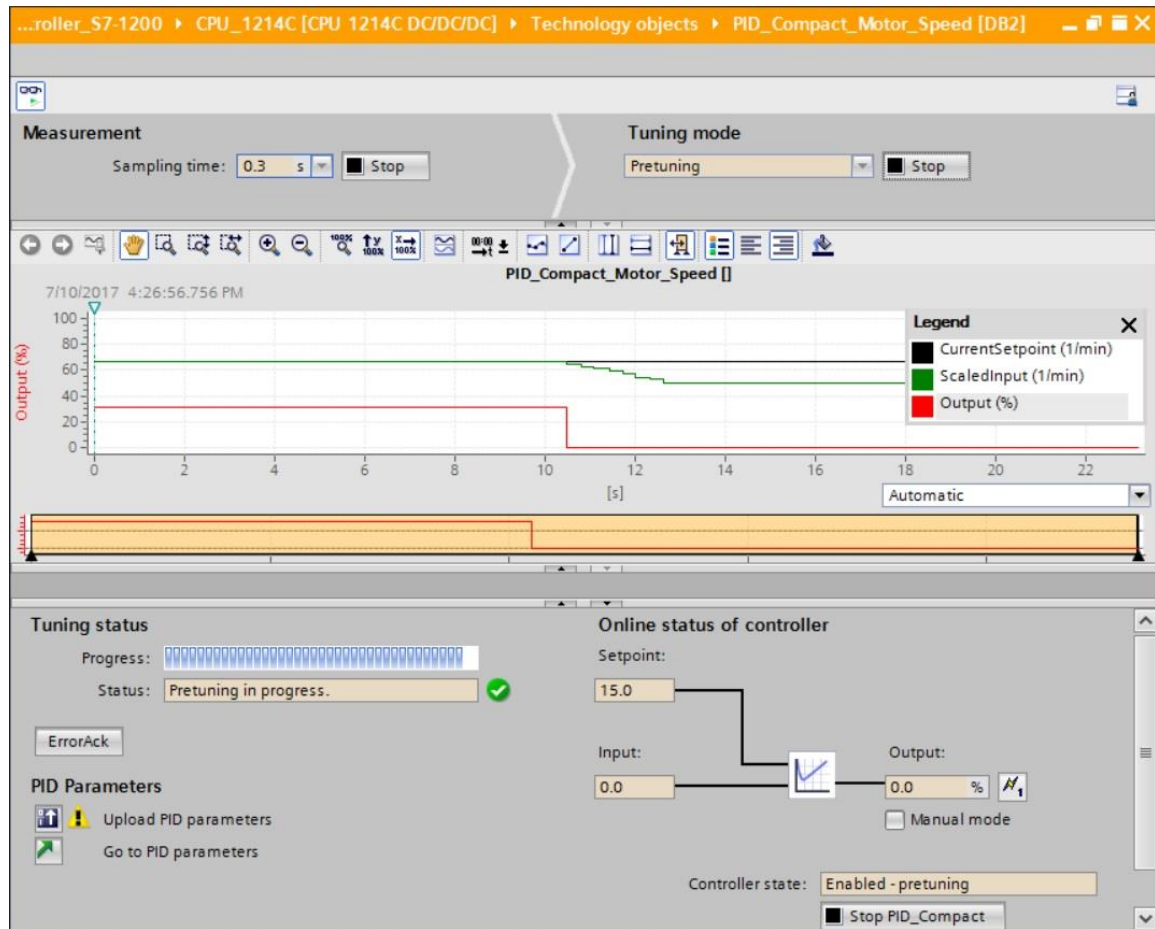
- "PID_Compact" 지침이 순환 인터럽트 OB 에서 호출됩니다.
- ManualEnable = FALSE
- Reset = FALSE
- PID_Compact 는 "수동 모드", "비활성" 또는 "자동 모드" 작동 모드에 있습니다.
- 설정값 및 실제값은 구성된 제한 내에 있습니다("프로세스 값 모니터링" 구성 참조).
- 설정값 및 실제값 간의 차이는 프로세스 값 상한 및 하한 간 차이의 30%보다 큼니다.
- 설정값 및 실제값 간의 차이는 설정값의 50% 이상입니다.

→ 'Pretuning'이 'Tuning mode'로 설정된 후에 시작됩니다.

(→ 조정 모드 → 사전 조정 → )



→ 사전 조정이 시작. 현재 작업 단계 및 발생하는 모든 오류는 "Tuning status" 필드에 표시됩니다. 진행 표시줄에 현재 작업 단계의 진행률이 표시됩니다.



7.7 PID_Compact 미세 조정

미세 조정은 실제값에 대해 일정하면서 제한된 발산을 생성합니다. PID 파라미터는 이러한 발산의 진폭 및 주파수를 기반으로 작동 지점에 맞게 최적화됩니다. 모든 PID 파라미터는 결과에서 다시 계산됩니다. 미세 조정으로 인한 PID 파라미터는 일반적으로 사전 설정의 PID 파라미터보다 설정값 변화 및 방해에 대해 더 나은 응답을 제공합니다. 사전 조정 및 미세 조정을 수행하여 최적의 PID 파라미터를 얻을 수 있습니다.

PID_Compact 는 자동으로 실제값 잡음보다 큰 발산 생성을 시도합니다. 미세 조정은 실제값 안정성에 따라 약간 영향을 받습니다. PID 파라미터는 다시 계산되기 전에 백업됩니다.

다음 요구 사항을 충족시켜야 합니다.

- "PID_Compact" 지침이 순환 인터럽트 OB 에서 호출됩니다.
- ManualEnable = FALSE
- Reset = FALSE
- 설정값 및 실제값은 구성된 제한 내에 있습니다.
- 제어 루프는 작동 지점에서 안정적입니다. 실제값이 설정값과 같을 때 작동 지점에 도달합니다.
- 방해가 예상되지 않습니다.
- PID_Compact 는 "수동 모드", "비활성" 또는 "자동 모드" 작동 모드에 있습니다.

미세 조정은 자동 모드에서 시작할 때 다음과 같이 실행됩니다.

기존 PID 파라미터를 조정하여 파라미터를 향상시키려면 자동 모드에서 미세 조정을 시작합니다.

PID_Compact 는 제어 루프가 안정적이고 미세 조정에 대한 요구 사항이 충족될 때까지 제어용으로 기존 PID 파라미터를 사용합니다. 그 이후에만 미세 조정이 시작됩니다.

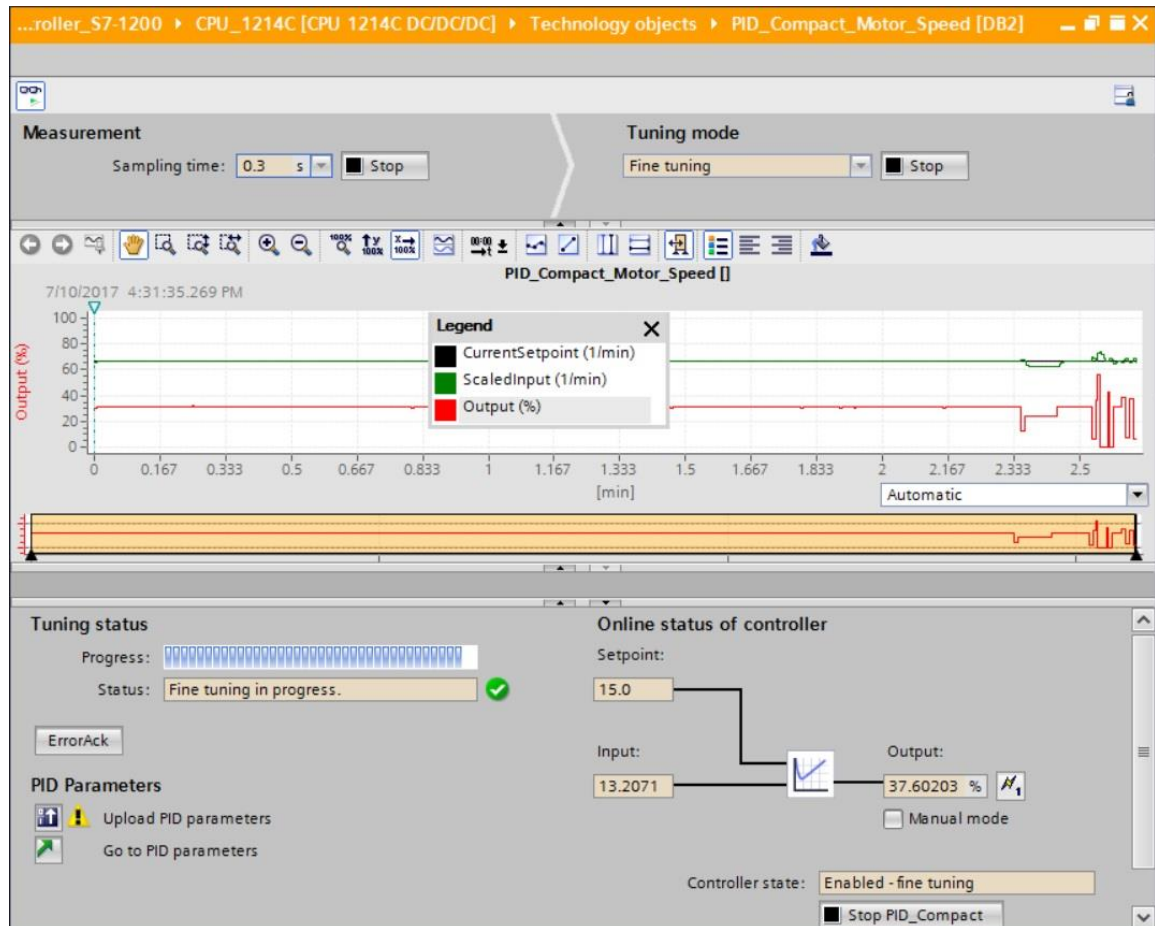
미세 조정은 비활성 또는 수동 모드에서 시작할 때 다음과 같이 실행됩니다.

사전 조정을 위한 요구 사항이 충족되면 사전 조정이 시작됩니다. PID_Compact 는 제어 루프가 안정적이고 미세 조정에 대한 요구 사항이 충족될 때까지 제어용으로 결정된 PID 파라미터를 사용합니다. 그 이후에만 미세 조정이 시작됩니다. 사전 조정이 불가능한 경우 PID_Compact 는 오류 응답에서 구성된 대로 응답합니다.

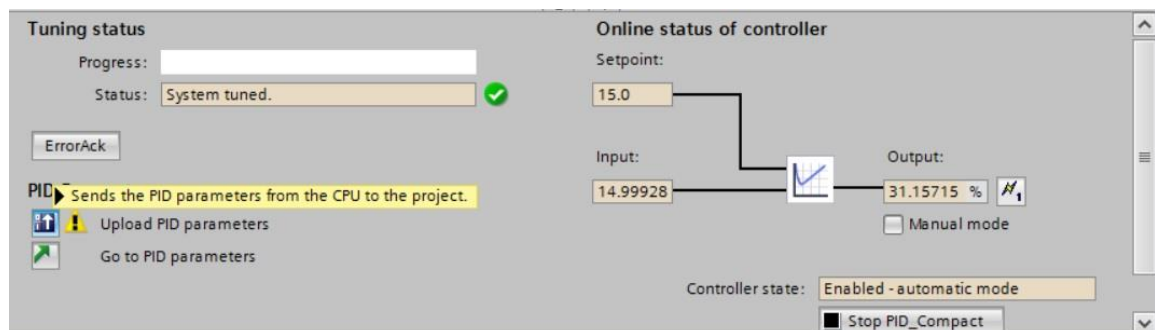
실제값이 이미 사전 조정을 위한 설정값에 너무 근접한 경우 최소 또는 최대 출력 값으로 설정값에 도달하려고 시도합니다. 이로 인해 오버슈트가 증가할 수 있습니다.

→ 'Tuning mode'에서 'Fine tuning'로 설정된 후에 시작됩니다.

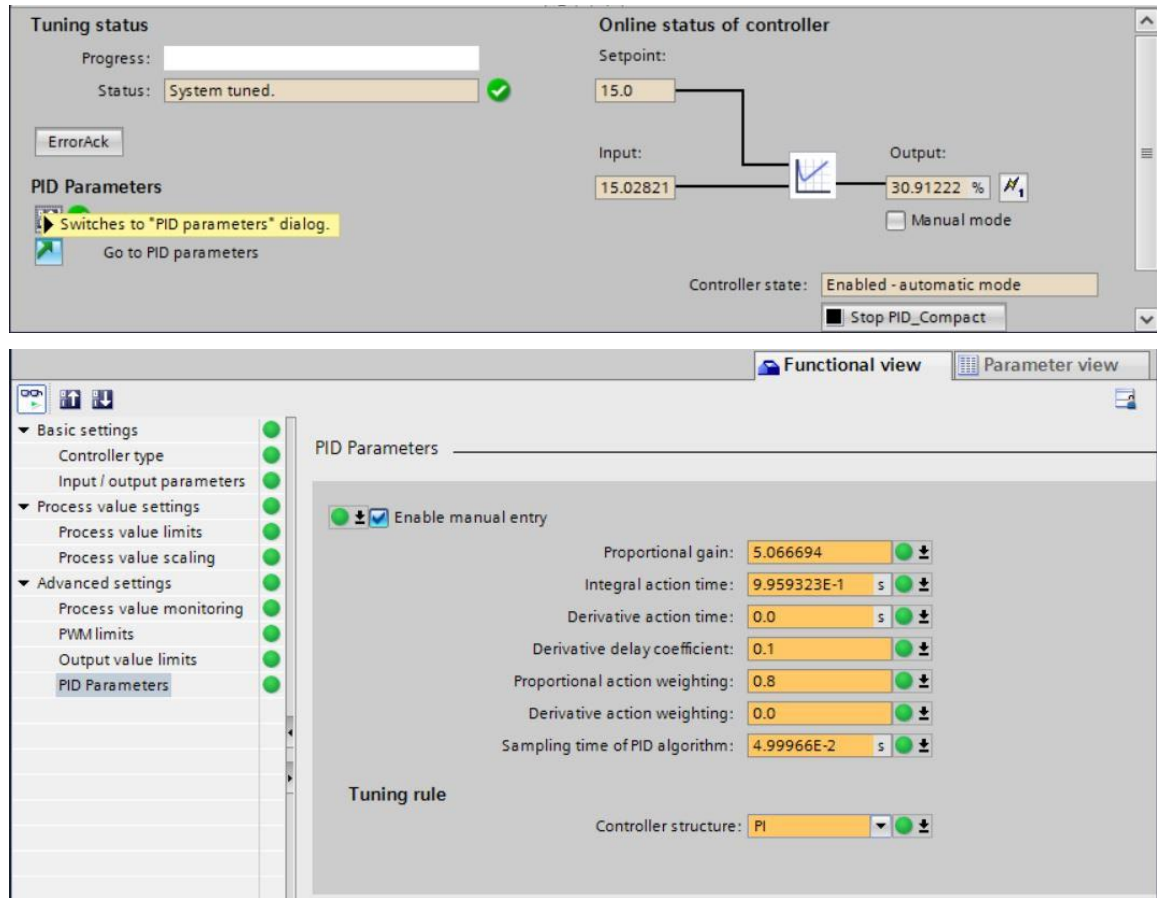
(→ 조정 모드 → 미세 조정 → )



→ 미세 조정이 시작. 현재 작업 단계 및 발생하는 모든 오류는 "Tuning status" 필드에 표시됩니다. 오류 메시지 없이 자체 조정이 완료된 경우 PID 파라미터가 조정됩니다. PID 컨트롤러가 자동 모드로 전환되고 조정된 파라미터를 사용합니다. 조정된 PID 파라미터는 전원을 켜고 CPU 를 다시 시작할 때 그대로 유지됩니다.  버튼을 사용하여 PID 파라미터를 CPU 에서 프로젝트로 업로드할 수 있습니다. (→ )



→ 구성의 PID Parameters 는  을 클릭하여 표시할 수 있습니다. (→ )

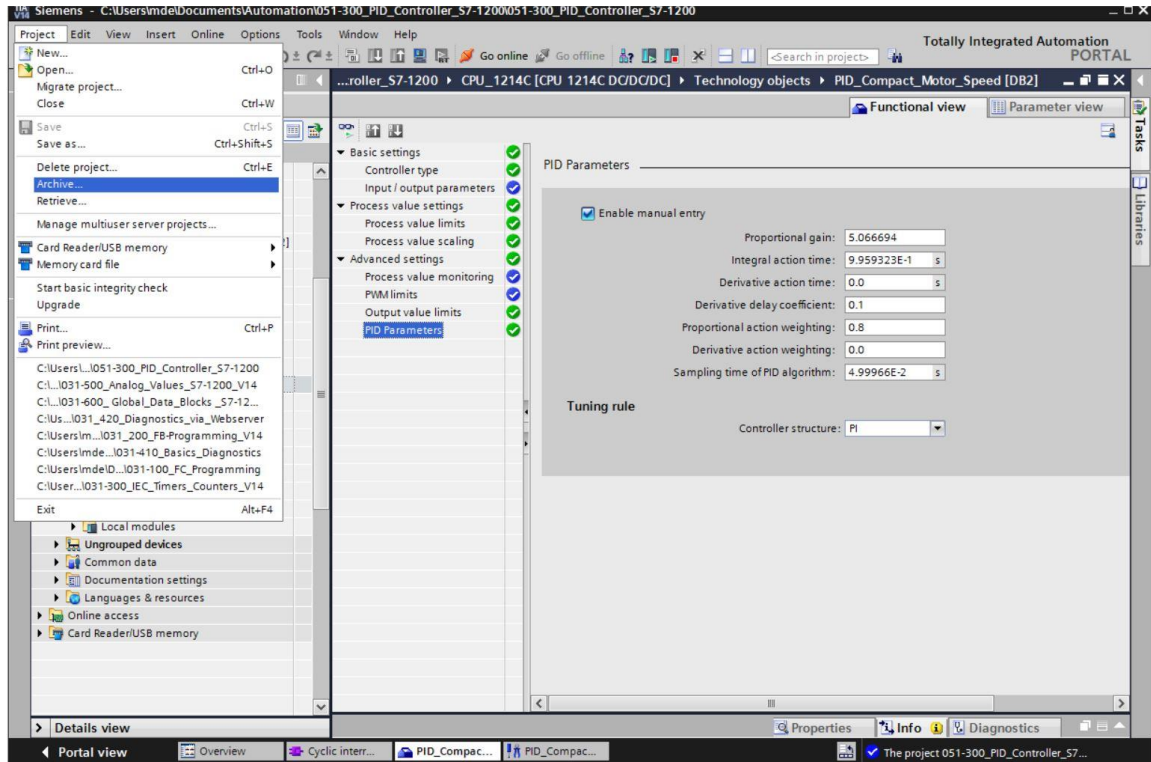


→ 최종 단계로 온라인 연결을 끊고 전체 프로젝트를 저장해야 합니다.
(→  Go offline →  Save project)

7.8 프로젝트 아카이브

→ 이제 전체 프로젝트를 아카이브하려고 합니다. → “Project” 메뉴에서 → “Archive...” 항목을 선택합니다. 프로젝트를 아카이브하고자 하는 폴더를 선택하고 “TIA Portal 프로젝트 아카이브” 파일 유형으로 이를 저장합니다.

(→ Project → Archive... → TIA Portal project archive → 051-300_PID_Control_S7-1200.... → Save)



8 체크리스트

번호	설명	완료
1	순환 인터럽트 OB Cyclic interrupt 50ms [OB30]이 성공적으로 생성됨	
2	순환 인터럽트 OB Cyclic interrupt 50ms [OB30]의 PID_Compact 컨트롤러가 호출되고 연결됨	
3	PID_Compact 컨트롤러 구성 수행	
4	오류 메시지 없이 성공적으로 컴파일	
5	오류 메시지 없이 성공적으로 다운로드	
6	오류 메시지 없이 사전 조정 성공	
7	오류 메시지 없이 미세 조정 성공	
8	스테이션 전원 켜기(-K0 = 1) 실린더 복귀 / 피드백 활성화(-B1 = 1) 비상 정지 오프(-A1 = 1)가 활성화되지 않음 자동 모드(-S0 = 1) 푸시버튼 자동 정지가 구동되지 않음(-S2 = 1) 자동 시작 푸시버튼을 짧게 누르기(-S1 = 1) 슬라이드의 센서 부분이 활성화되고(-B4 = 1) 이후 컨베이어 모터 M1 가변 속도(-Q3 = 1) 스위치를 켜고 그 상태를 유지합니다. 속도는 +/-50 rpm 범위의 속도 설정값에 해당	
9	컨베이어 끝의 센서 부분이 활성화(-B7 = 1) → -Q3 = 0(2 초 후)	
10	푸시버튼(자동 정지)을 짧게 누르기(-S2 = 0) → -Q3 = 0	
11	비상 정지를 활성화(-A1 = 0) → -Q3 = 0	
12	수동 모드(-S0 = 0) → -Q3 = 0	
13	스테이션 전원 끄기(-K0 = 0) → -Q3 = 0	
14	실린더가 복귀되지 않음(-B1 = 0) → -Q3 = 0	
15	속도 > Motor_speed_monitoring_error_max → -Q3 = 0	
16	속도 < Motor_speed_monitoring_error_min → -Q3 = 0	
17	프로젝트가 성공적으로 아카이브됨	

9 추가 정보

초기 및 심화 교육에 방향을 제시하는 도우미로서 예를 들어 시작하기, 동영상, 교재, 앱, 매뉴얼, 프로그래밍 지침, 체험용 소프트웨어/펌웨어와 같은 추가 정보를 아래 링크에서 찾아보실 수 있습니다.

www.siemens.com/sce/s7-1200

"추가 정보" 미리보기

□ Getting Started, Videos, Tutorials, Apps, Manuals, Trial-SW/Firmware

- TIA Portal Videos
- TIA Portal Tutorial Center
- Getting Started
- Programming Guideline
- Easy Entry in SIMATIC S7-1200
- Download Trial Software/Firmware
- Technical Documentation SIMATIC Controller
- Industry Online Support App
- TIA Portal, SIMATIC S7-1200/1500 Overview
- TIA Portal Website
- SIMATIC S7-1200 Website
- SIMATIC S7-1500 Website

추가 정보

Siemens Automation Cooperates with Education
siemens.com/sce

SCE 교육 커리큘럼
siemens.com/sce/documents

SCE 교육 담당자 패키지
siemens.com/sce/tp

SCE 담당 파트너
siemens.com/sce/contact

Digital Enterprise
siemens.com/digital-enterprise

Industrie 4.0
siemens.com/future-of-manufacturing

완전히 통합된 자동화 시스템 (TIA)
siemens.com/tia

TIA Portal
siemens.com/tia-portal

SIMATIC 컨트롤러
siemens.com/controller

SIMATIC 기술 문서
siemens.com/simatic-docu

산업 온라인 지원
support.industry.siemens.com

제품 카탈로그 및 온라인 주문 시스템 산업 물
mall.industry.siemens.com

Siemens AG
Digital Factory
P.O. Box 4848
90026 뉘른베르크
독일

오류는 제외되며 사전 통보없이 변경될 수 있습니다.
© Siemens AG 2018

siemens.com/sce