



TM242 Sequential Function Chart (SFC)

I 버전 정보

버전	날짜	수정내역	번역	검수
1.0	2017.12.14	첫번째 버전 TM242TRE.00 ENG (v1.0.1) AS3090	-	Eun Im

Table 1: Versions

선행 및 필요 조건

교육 자료	TM210 – The Basics of Automation Studio TM246 – Structured Text (ST)
소프트웨어	Automation Studio 3.0.90
하드웨어	-



II 목차

1	소개	1
1.1	학습 목표	1
2	Sequential Function Chart	2
2.1	일반 정보	2
2.2	SFC의 기본 기능	3
2.3	SFC 편집기 기능	4
3	단계(Steps)	6
3.1	초기화 단계	6
3.2	진입 및 퇴출 동작	6
3.3	단계 타이밍 감시	7
3.4	동작 단계(Action steps)	7
3.4.1	동작 작성과 관련시키기	9
3.4.2	동작의 '최종 스캔(final scan)' 평가하기	10
3.5	복합 단계	10
4	변환(Transitions)	11
4.1	프로그램 코드를 이용한 변환	11
4.1.1	래더 다이어그램의 변환	12
4.1.2	Structured Text의 변환	12
4.1.3	FBD과 CFC의 변환	13
4.1.4	Instruction List의 변환	13
5	대체 및 병렬 분기	14
5.1	대체 분기 (Alternative branches)	14
5.2	병렬 분기 (Parallel branches)	15
6	점프	16
7	예제	17
7.1	애플리케이션 예제: 충전 제어 (Fill control)	17
7.2	애플리케이션 예제: 혼합기 제어 (Mixer control)	18
8	시스템 변수 이용하기	20
8.1	애플리케이션에서 사용	20
9	진단 기능	22
9.1	SFC 모니터 모드	22
9.2	SFC 변수	23
9.3	디버거	24
10	요약	25
11	부록: 예제 솔루션	26
11.1	예제 솔루션: 충전 제어 (Fill control)	26

11.2 예제 솔루션: 혼합기 제어 (Mixer control).....	27
Automation Academy 에서 제공하는 것	28

1 소개

Sequential Function Chart 는 줄임말, SFC 로 불린다. SFC 는 일련의 단계를 중심으로 배열된 시각적 프로그래밍 언어이다. 명확성을 개선할 뿐만 아니라 프로그램 문제 해결을 용이하게 한다.

SFC 는 소프트웨어 설계 과정에서 발생하는 순차 및 병렬 시퀀스를 작성하고 구현하는 것이 가능하다.

교육 자료는 Sequential Function Chart 프로그래밍 언어에서 이용되는 요소를 다룬다.

심화 이해 수준을 제공하기 위한 예제도 포함되어 있다. 단계와 변환에서 일어나는 프로그래밍은 래더 다이어그램 (Ladder Diagram)과 Structured Text 를 이용하여 작성한다.

여기에서 다루는 각 장에 관한 추가 정보는 Automation Studio 도움말에서 찾아볼 수 있다.



그림 1 Sequential Function Chart (SFC)



Programming W Programs W Sequential Function Chart(SFC)

Programming W Editors W Graphic editors W Sequential Function Chart editor

1.1 학습 목표

이 교육 자료에서는 다음을 학습한다.:

- ... Sequential Function Chart 언어 사용법
- ... 단계와 변환을 올바르게 이용하는 방법
- ... 동작 단계를 이용하는 방법
- ... 프로그래밍 규칙
- ... 문제 해결 옵션

2 Sequential Function Chart

2.1 일반 정보

Sequential Function Chart 는 여러 가지 프로그램 섹션과 순차적 단계 실행 조정이 가능한 시각적 프로그래밍 언어이다. 이 프로그래밍 언어는 프로그램 단계와 개별 단계에 정의된 변환이 필요한 경우에 특히 유용하다. 단일 프로그램에서 병렬 시퀀스를 실행하는 것도 가능하다.



그림 2 Sequential Function Chart 속성

Sequential Function Chart 는 기계 상태(state machines)에 따라 함께 사용하는 곳에 적합하다. 병렬 처리를 지원하기 때문에 ("[병렬 분기](#)"), 복잡한 설계를 명확한 구현과 관리가 가능하다. 프로그램의 시각적 구조는 시퀀스의 문서 작성과 문제 해결에도 유리하다.

SFC 는 언제 이용되는가?

시퀀스를 직접 기술할 수 있는 경우에는 언제나 시퀀스를 구현하기 위해 SFC 프로그래밍 언어를 이용하는 것이 가능하다. 기계 상태는 SFC 로 "재작성(redrafted)"되고 구현될 수 있으나; 그 반대는 일반적으로 가능하지 않다.

유용한 정보

EN 60848 에 정의된 GRAFCET(어원은 프랑스어 "**GRA**phe **Fonctionnel** de **Com**mande **Eta**-**pes/Transitions**"의 두문자)는 시퀀스를 표시하기 위한 표준화된 언어이다. 주로 자동화 분야에서 이용되지만 프로세스 엔지니어링에서도 찾아볼 수 있다.

IEC 61131-3 에 규정된, Sequential Function Chart 는 GRAFCET 플랜을 구현하기 위한 한 가지 방법으로써 GRAFCET 에 구체적으로 언급되어 있다 (de.wikipedia.org/wiki/GRAFCET).

대규모 애플리케이션 작성하기

보다 큰 애플리케이션을 디자인할 때는 SFC 가 다루도록 설계된 특성, 즉 시퀀스 제어를 실제로 이용할 수 있도록 하기 위해 미리 모든 것을 고려할 필요가 있다.

실제 시퀀스는 SFC 프로그램으로 구현할 것을 권장한다. 입출력 데이터의 평가, 수치 스케일링 또는 액추에이터 제어와 같은 기계 기능은 추가적인 프로그램 모듈에서 다루어야 한다.

SFC 프로그램은 정의된 인터페이스를 통해 하위 기능을 호출하고 각 기능의 상태를 검색할 수 있다.

¹ IEC 61131-3 표준은 프로그래밍 할 수 있는 로직 제어기에 사용할 수 있는 프로그래밍 언어에 대한 유일한 합법적인 국제 표준이다. 이 표준에는 래더 다이어그램, Instruction List, Structured Text 및 Function Block Diagram 이 포함되어 있다.

이 구조 유형이 유리한 것은 기계 하위 기능은 변경될 수 있지만 (예, 새로운 센서나 상이한 드라이브) 시퀀스 그 자체는 변동 없이 유지되기 때문이다. 이러한 종류 구분은 시퀀스 체인과 기계의 하위 기능을 모두 시험할 수 있는 좋은 방법이다.

 이 문서의 나머지 부분에서 Sequential Function Chart 는 줄임말인 SFC 라고 부르기로 한다.

2.2 SFC 의 기본 기능

SFC 의 기본적 컴포넌트는 단계(step)와 변환(transition)이다. 단계는 여러 가지 유형의 경계를 갖는 직사각형으로 표시된다. 변환은 단계 사이의 연결과 교차하는 수평선으로 표시된다.

모든 단계와 변환은 번갈아 가면서 서로에게 연결된다. 다시 말해서, 변환은 언제나 단계를 뒤따른다. 하나의 단계와 그 변환은 하나의 유닛(unit)을 형성한다. 예를 들어서 두 단계 또는 두 변환이 연속적으로 일어나는 것은 불가능하다.

여기에 제시된 예시는 단계 세 개, 관련 변환 및 점프 한 개를 갖는 프로그램을 보여준다.

SFC 프로그램이 처음 호출되면 “첫 번째(FIRST)” 단계가 스캔된다. 이 단계는 이중 경계에 의해 식별되고 프로그램이 초기화될 때 언제나 호출된다. 연관된 변환은 참(TRUE) 값을 가지며, 이것은 초기화 단계가 한 주기에 대해서만 호출될 것임을 의미한다.

그 다음 프로그램은 “가열(HEAT)” 단계에 머무르고 이 단계는 “과열(tooHot)” 변환이 참(TRUE)이 될 때까지 지속된다. 그 다음 “가열(HEAT)” 단계는 비활성화되고, 프로그램은 “냉각(COOL)” 단계로 이동한다. “과냉(tooCool)” 변환이 참이 되는 순간, 점프가 실행되고 다시 한번 “가열(HEAT)” 단계가 활성화된다.

프로그램 주기 당 한 개 단계만 실행된다.

단계 명칭은 상징적이고 반드시 선언할 필요는 없다. 반면에 변환은 불(BOOL) 또는 논리적 표현의 데이터 유형 변수이고 다양한 프로그래밍 언어로 결합될 수 있다.

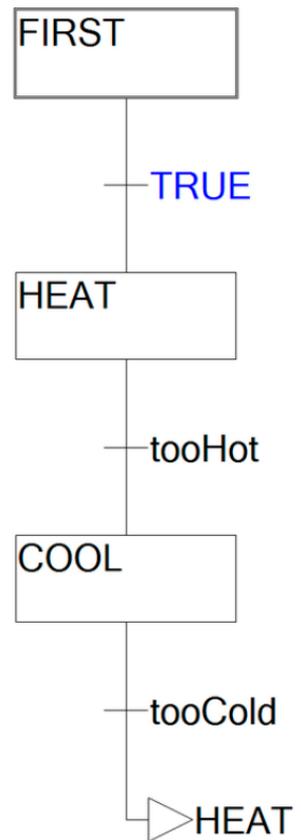


그림 3 단계 2 개, 점프 1 개, 초기화 단계 1 개를 갖는 시퀀스

 한 단계에 한 개 변환은 한 개 유닛을 형성하며, 변환이 각 단계를 뒤따른다. 두 개 이상의 변환이나 단계가 연속적으로 나타나는 것은 불가능하다.

한 단계에서 다음 단계로 이동할 때, 다음 단계는 다음 프로그램 주기가 시작될 때만 수행된다.

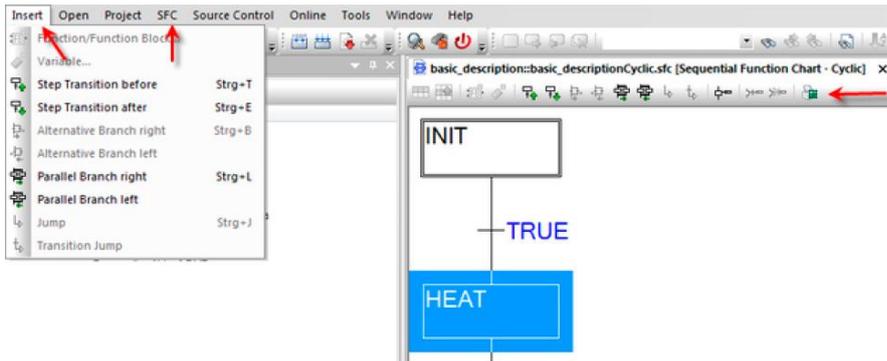
 Programming W Programs W Sequential Function Chart (SFC)

2.3 SFC 편집기 기능

SFC 로 프로그래밍할 때는 반드시 단계, 변환 및 점프를 삽입해야 한다.

Automation Studio 의 SFC 편집기를 운영할 수 있는 장소는 여러 곳이다:

- 주 메뉴: Insert
- 주 메뉴: SFC
- 편집기 툴바(Editor toolbar)
- SFC 프로그램, 단계 및 변환에 대한 단축 메뉴



객체는 주 메뉴와 툴바를 이용하여 삽입된다. 객체 속성은 주 메뉴로 또는 객체의 단축 메뉴를 이용하여 정의할 수 있다. 선택된 요소는 더블 클릭하여 열 수 있다.

줌 툴바(Zoom toolbar) 또는 메뉴 바의 “뷰(View)” 메뉴를 이용하여 SFC 편집기를 줌인 또는 줌아웃할 수 있다.

 Programming W Editors W Graphic editors W Sequential Function Chart editor

Programming W Editors W Graphic editors W Sequential Function Chart editor W General editor settings

Project management W the workspace W Toolbars W Zoom

첫 SFC 프로그램 작성하기

예제에서 SFC 의 기본 기능에서 설명한 간단한 프로그램을 작성한다.

- 1) 로지컬 뷰(logical view)에 새로운 SFC 프로그램을 삽입한다.
- 2) 프로그램 이름은 “sfc_basic”으로 한다.
- 3) 단계 “HEAT”을 삽입한다.
- 4) 단계 “COOL”을 삽입한다.
- 5) 변환 “tooHot” 및 “tooCold” 명칭을 부여하고 선언한다.
- 6) 프로그램을 컴파일한다.
- 7) 모니터 모드를 활성화한다.
- 8) 변수 감시 창(variable watch window)에서 프로그램을 시험한다.
변수 감시 창에서 변환 값을 수동으로 설정함으로써 프로그램을 시험할 수 있다. 변환 중 하나가 설정되면, SFC 프로그램은 다음 활성화 단계로 이동한다. 모니터 모드가 활성화된 상태에서 “Powerflow”를 활성화하면 활성화 단계와 전환된 변환은 채색된다.

파워 플로우(Powerflow)가 활성화 되면, 활성화 단계와 변환은 색깔로 표시된다. 변환은 변수 감시 창 혹은 SFC 프로그램을 클릭하여 설정할 수 있다.

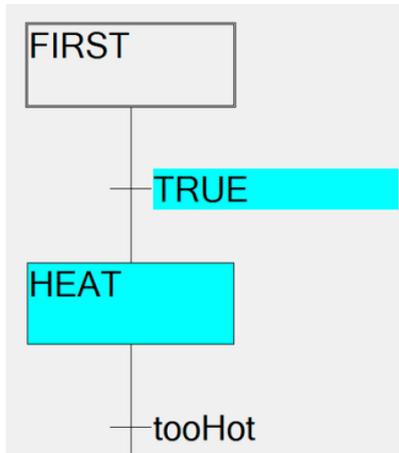


그림 4 “FIRST”이후 “HEAT”과 변환이 활성화된다.



Diagnostics and service W Diagnostic tool W Monitors W Sequential Function Chart editor in monitor mode

3 단계(Steps)

단계는 실행 가능한 프로그램 코드나 동작(action)을 호출할 수 있다. 이 코드는 단계가 활성화되면 언제나 실행된다. 단계는 활성화, 실행, 비활성화 될 수 있다. 단계에 대한 타이밍도 감시할 수 있다.

단계는 직사각형으로 표시된다. 단계가 사이클릭 프로그램 코드(cyclic program code)에 포함되는 경우는 우측 상단 모서리에 흑색 삼각형이 표시된다.

단계 내용물은 IEC 프로그래밍 언어 중 하나로 작성될 수 있는 프로그램으로 구성된다.



그림 5 실행 가능한 코드를 갖는 단계

 직접 실행 가능한 코드를 포함하는 단계는 IEC 표준의 확장이다. 사이클릭 동작은 일반적으로 IEC 동작으로 구성된다 ([“동작 단계\(Action steps\)”](#) 참조).

 Programming W Programs W Sequential Function Chart (SFC) W Step
Programming W Editors W Graphic editors W Sequential Function Chart editor W Defining SFC cyclic actions

3.1 초기화 단계

초기화 단계(Initialization steps)는 사이클릭 프로그램의 첫 번째 단계로 프로그램이 시작된 직후에 실행된다. 초기화 단계를 초기화 하위 프로그램과 혼동하지 않는 것이 중요하다. 초기화 단계는 또한 복귀(return) 후 또는 SFCReset 이나 SFCInit 가 수행된 후에도 실행된다. 복합 단계가 활성화 될 때, SFC 프로그램 밑에 있는 초기화 단계가 활성화된다.

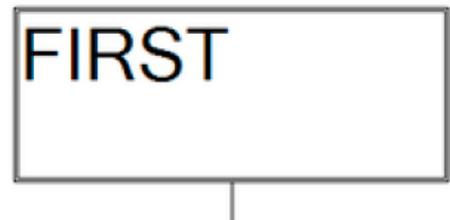


그림 6 초기화 단계

초기화 단계는 이중 경계로 표시된다.

다음 사항 참조:

- [“복합 단계”](#)
- [“시스템 변수 이용하기”](#)

3.2 진입 및 퇴출 동작

초기화 단계를 제외한 모든 단계는 옵션의 진입(entry) 및 퇴출(exit) 동작을 가질 수 있다. 진입 동작은 문자 “E”, 퇴출 동작은 문자 “X”로 표시된다. 초기화 단계는 퇴출 동작만을 가질 수 있다.

일단 단계가 활성화되면 진입 동작이 (E) 호출된다. 단계가 비활성화되면 퇴출 동작이 (X) 한 번 호출된다.

이 동작은 단계의 단축 메뉴로 추가되거나 삭제될 수 있다.



그림 7 진입과 퇴출 동작을 갖는 단계

 Programming W Editors W Graphic editors W Sequential Function Chart editor W Defining SFC entry actions

Programming W Editors W Graphic editors W Sequential Function Chart editor W Defining SFC exit actions

3.3 단계 타이밍 감시

각 단계의 타이밍을 감시할 수 있다 (옵션). 최소 또는 최대 시간이 초과되었는지 여부를 확인할 수 있다. 설정된 시간 위반을 차단하기 위해 시스템 변수를 이용할 수 있다 ([“시스템 변수 이용하기”](#) 참조).

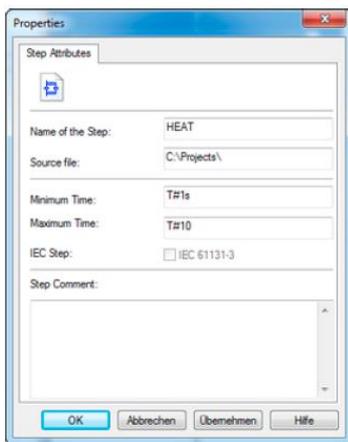


그림 8 단계 속성 다이얼로그 박스, 최소 최대 실행 시간

 시간 감시외에도, 특성 대화 상자에는 단계에 대한 설명(comment)을 추가할 수 있는 공간이 있다.
이 설명과 시간 값은 단계의 툴팁(tooltip)에 표시된다.

3.4 동작 단계(Action steps)

동작 단계(Action steps)는 IEC 단계라고도 한다. 동작 단계 자체는 어떠한 프로그램 코드도 포함되지 않는다. 다만 동작 단계와 관련된 동작만 포함할 수 있다. 동작은 단계가 활성화될 때 실행된다.

동작 블록에는 두 가지 유형이 있다:

- 불 대수 동작 (Boolean actions)
- 프로그램 코드를 갖는 동작

동작이 호출되는 방법을 결정하기 위해 한정자(qualifier)를 정의할 수 있다. 예를 들어, 단계가 활성화될 때 동작이 특정한 시간 만큼 지연되는 것을 규정할 수 있다.

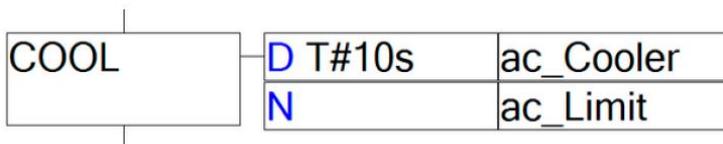


그림 9 지연 동작과 사이클릭 동작을 갖는 단계

예를 들어 프로그램 코드를 갖는 IEC 동작 내용은 structured text 와 같은 별도 프로그램 모듈에서 프로그램한다. IEC 동작은 여러 번 이용될 수 있기 때문에, 상이한 한정자를 갖는 여러 단계와 관련시킬 수 있다.

불 대수(Boolean) 동작은, 한정자 이외에 불(BOOL) 데이터 유형의 변수와 연결된다. 동작이 실행되면 언제나 변수 값은 참(TRUE)으로 설정된다.

사용 가능한 한정자 목록은 도움말 문서를 참조한다.

중요한 한정자(qualifiers) 개요:

- 사이클릭으로 동작 호출(Calls on action cycilcally) (N)
- 동작을 일시적으로 제한(Limits an action temporally) (L)
- 동작 지연(Delays an action) (D)
- 조건부 동작 설정(Sets an action conditionally) (S)
- 동작 리셋(Resets an action) (R)

IEC 단계 활성화하기

툴바에서 설정하면 사이클릭 프로그램 코드를 갖는 단계나 IEC 동작을 갖는 IEC 단계의 삼입 여부를 설정하는 것이 가능해진다. 이미 배치된 단계들은 단계의 특성 대화 상자에서 재설정될 수 있다.



그림 10 “IEC 단계 사용하기” 디폴트 설정

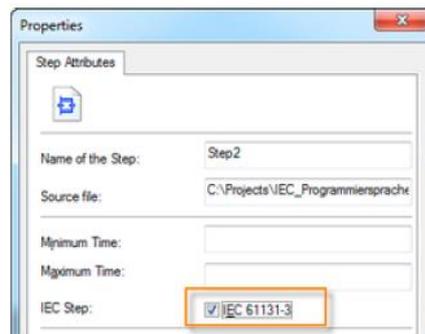


그림 11 “IEC 61131-3” 단계 속성

단일 동작 단계에 각종 한정자를 10 개까지 갖는 여러 가지 동작을 관련시킬 수 있다.

SFC 에서 동작 블록은 “최종 스캔(final scan)” 논리에 따라 실행된다.

이러한 이유로 한정자 “P” 동작은 언제나 두 번 호출된다.

현재 동작이 처리되고 있는 스캔을 동작 인스턴스 구조와 함께 볼 수 있다 ([“동작의 ‘최종 스캔\(final scan\)’ 평가하기”](#)).

게다가, 비활성화된 동작들은 프로그램 말미에 알파벳 순서로 호출된다. 그 다음 새로운 활성화 동작들 또한 알파벳 순서로 호출된다. 시퀀셜 프로그램을 구성할 때는 이 점을 고려해야 한다.

Programming W Programs W Sequential Function Chart (SFC) W Action step

Programming W Actions

Programming W Actions W Action block – Qualifiers

Programming W Actions W Using actions in the various programming languages

3.4.1 동작 작성과 관련시키기

프로그램 코드를 이용한 동작 작성

프로그램 코드를 포함하는 동작을 단계와 관련시키기 전에 반드시 먼저 작성해야 한다. 새로운 동작(Action)은 로지컬 뷰(logical view)의 위저드를 이용하여 추가한다.

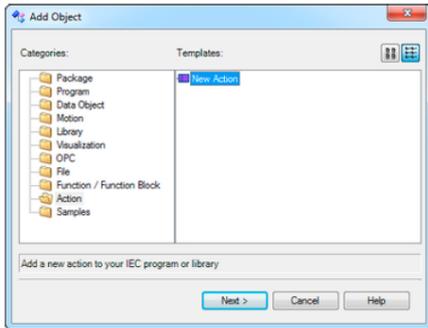


그림 12 로지컬 뷰에서 새로운 동작 추가하기

Project Management W Logical view W Wizards in the logical view

불 대수(Boolean) 동작 작성하기

불 대수(Boolean) 동작에 필요한 것은 데이터 유형 불(BOOL)의 로컬 또는 글로벌 변수뿐이다. 이것은 반드시 동작이 단계와 관련되기 전에 변수 선언 창에 선언되어야 한다.

동작을 단계와 관련시키기

“동작 관련(Associate action)” 툴바 아이콘을 이용하여 동작을 단계와 관련시킬 수 있다. 반드시 단계가 먼저 선택되어야 한다.



그림 13 "Associate action" 아이콘

실제 동작과 한정자를 (N, D, P 등) 단계와 관련된 동작에 지정할 수 있다. 가능한 동작 목록은 (코드 또는 불 대수(Boolean)를 갖는) 다음 키 조합으로 열 수 있다 “<CTRL> + <space bar>”.

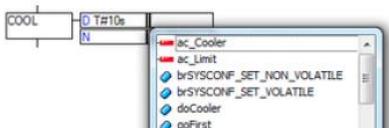


그림 14 <CTRL> + <space bar>를 이용하여 모든 동작과 불 대수(Boolean) 변수 표시하기

상이한 단계에 상이한 한정자를 갖는 동작을 사용할 수 있다. 그러나 이것이 언제나 좋은 아이디어는 아니다.

예를 들어 동작이 “S” 한정자로 설정된 경우에, 동일한 동작에 영향을 미치는 한정자 “R”을 가정할 수 있다.

그러나 이 동작이 한정자 “N”과 함께 호출된다면, 그 동작은 단계가 비활성화될 때 이미 리셋된다. 따라서 “R” 한정자는 아무런 영향도 미치지 않는다. 동작이 이미 리셋되었기 때문이다.



Programming W Editors W Graphic editors W Sequential Function Chart editor W Using action blocks
 Programming W Editors W General operations W SmartEdit

3.4.2 동작의 '최종 스캔(final scan)' 평가하기

프로그램 배열에 따라 단 한 번만 동작을 호출할 필요가 있을 수 있다. 이런 경우에, 마지막 스캔은 액션의 인스턴스 구조를 사용하여 프로그래밍 코드로 액션에서 쿼리(queried) 할 수 있습니다.



이름 “ac_Once”와 한정자 “P”를 갖는 동작은 한 번만 호출해야 한다. 이로 인해 최종 스캔은 동작에서 조회된다. 여기에 이용되는 프로그래밍 언어는 Structured Text(ST)이다.

프로그램 코드	<pre> ACTION ac_Once: IF ac_Once.x AND NOT ac_Once._x THEN (* last scan *) ELSE OnlyOnce := TRUE; END_IF; END_ACTION </pre>
---------	---

3.5 복합 단계

복합 단계(Compound step)는 사이클릭 프로그램 코드로써 다른 SFC 프로그램을 포함하는 SFC 단계이다. 이 단계는 우측 상단 모서리의 흑색 삼각형과 회색 음영으로 표시된다.

예를 들어 더 작은 시퀀스를 더 크고 더 복잡한 시퀀스에 통합하는 것이 가능하다. 하위 SFC 프로그램은 초기화 단계뿐만 아니라 주 프로그램에서 이용 가능한 그 밖의 모든 가능성이 포함된다.

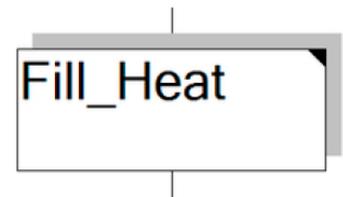


그림 15 복합 단계



Programming W Programs W Sequential Function Chart (SFC) W Compound step
 Programming W Editors W General editor W Sequential Function Chart editor W Using several SFC layers

4 변환(Transitions)

각 단계는 그 단계를 비활성화하는 데 이용될 수 있는 변환(transition)를 갖는다. 변환이 활성화 되면 관련 단계는 비활성화되고 다음 주기의 다음 단계가 활성화된다.

변환은 “참(TRUE)” 또는 “거짓(FALSE)” 중 한가지 값을 갖는다. 값이 “참(TRUE)”이면, 변환이 활성화된다.

변환 값은 상수, 변수 혹은 불 대수(Boolean) 값을 되돌려 보내는 프로그램 코드로부터 수신될 수 있다.

기본 프로그램 코드를 기반으로 하는 변환은 흑색 삼각형으로 표시된다. 변환 텍스트는 녹색으로 표시되고 설명(comment) 역할을 한다.

두 개의 변환이 연이어 배치되는 것은 불가능하다.

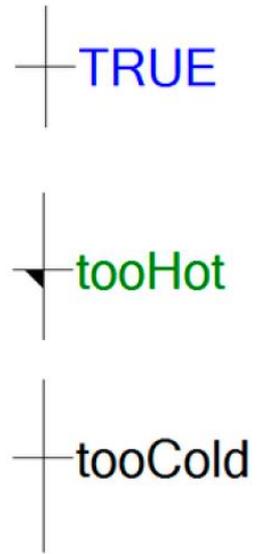


그림 16 사우, 프로그램 코드, 변수를 갖는 변환



변환의 전환으로 다른 단계로 이동하는 것은 동일한 프로그램 주기에서는 일어나지 않는다. 새로운 활성화 단계는 언제나 다음 주기에서 활성화된다. 단계의 연쇄에서 여러 변환이 활성화되는 경우에 SFC 프로그램은 안정된 상태를 찾는다. 이후에 비활성화 변환을 갖는 단계는 사이클릭으로 실행된다. 스캔되는 단계에 대한 진입과 퇴출 동작이 호출된다.



Programming W Programs W Sequential Function Chart (SFC) W Transition W Trasistion codition

4.1 프로그램 코드를 이용한 변환

변환은 프로그램 코드를 포함할 수 있다. 유일한 조건은 프로그램 코드가 호출되면 결과를 반환한다는 것이다. 반환 값은 반드시 불(BOOL) 데이터 유형이어야 한다. 이것은 변환의 전환 상태를 나타낸다.

논리 연산자를 (AND, OR, XOR 등) 조합하는 것이 가능하다. 기능을 호출할 수 있다.

프로그램 코드를 이용한 변환에서 다음 프로그래밍 언어를 이용할 수 있다:

- Instruction List
- Ladder Diagram
- Structured Text
- Function chart
- Continuous Function Chart
- B&R Automation Basic

변환 프로그램 코드는 간단히 변환을 클릭하거나 단축 메뉴로 삽입할 수 있다. 프로그래밍 언어를 선택하기 위한 대화 상자가 열린다.



주의(WARNING)!

변환에서는 기능만 이용할 수 있다. 변환에서 어떤 표현이 사용되든 최초 호출에서 반드시 명확히 정의된 결과가 반환되어야 한다. 평선 블록의 호출을 시도하면 컴파일러는 오류 메시지를 전달한다.:

Error 1453: A transition is not allowed to cause side effects (assignments, function block calls, etc.).

(해석) 오류 1453: 부작용을 (할당, 평선 블록 호출 등) 야기하는 변환은 허용되지 않는다.

4.1.1 래더 다이어그램의 변환

변환은 래더 다이어그램의 네트워크로 프로그램 될 수 있다. 변환은 네트워크의 종말에 주어진다. 래더 다이어그램에 변환이 만들어지면, 출력 위치에 배치된 변환과 함께 자동적으로 네트워크가 만들어진다.

“변환(Transition)” 식별자가 사용되고 시스템에 미리 선언되어야 한다.

래더 다이어그램의 변환은 다음과 같다.

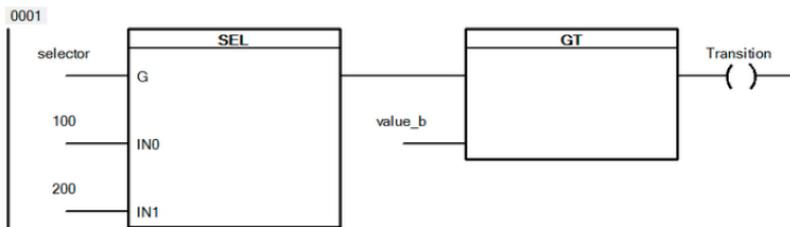


그림 17 두 평선 변환. “selector” 변수는 후속 비교를 위해 사전 선택 값을 결정합니다.



Programming W Programs W Ladder Diagram (LD)

4.1.2 Structured Text 의 변환

Structured Text 에서 변환은 할당 연산자 없는 표현으로 정의된다. 논리적 연산자를 이용하는 비교 연산자 기능이 허용된다.

비교되는 값이 같지 않고 두 값의 최댓 값이 100 을 초과하지 않으면 다음 선언은 참이다.

`(value_a <> value_b) AND (MAX(value_a, value_b) < 100)`



Programming W Programs W Structured Text (ST)

4.1.3 FBD 과 CFC 의 변환

평선 블록 다이어그램(Function Block Diagram: FBD)과 Continuous Function Char(CFC)에서, 변환은 평선 차트 네트워크에 배치될 수 있다. 결과는 “변환(Transition)” 변수에 할당된다. 이 변수는 이미 시스템에 존재하고 선언한 필요가 없다.

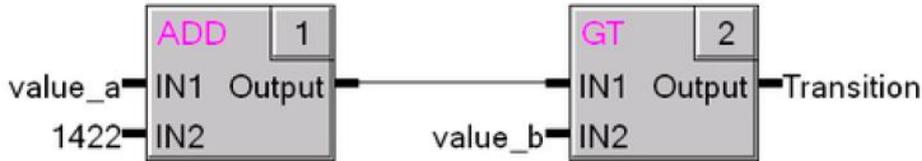


그림 18 변환에 “greater than” 비겨 결과 더하기

 Programming W Programs W Function Block Diagram (FBD)

4.1.4 Instruction List 의 변환

변환은 할당 연산자 없이 Instruction List(IL) 언어에서 실행될 수 있다. 표현이 호출되는 경우에는 반드시 불 대수(Boolean) 결과를 반환해야 한다.

아래 변환은 “value_a”가 200 보다 큰 경우에 참으로 설정된다.

```
(* Transition is TRUE if *)
(* value_a > 200      *)
LD      value_a
GT      200
```

 Programming W Programs W Instruction List (IL)

5 대체 및 병렬 분기

SFC 에서 개별 단계들은 반드시 한 단계를 실행한 이후 바로 다음 단계를 실행할 필요는 없다. 더 많은 유연성을 제공하기 위해서 병렬 및 대체 분기가 허용된다.

5.1 대체 분기 (Alternative branches)

시퀀스가 여러 개 상이한 경로로 분할되어야 하는 경우에 SFC 는 한 개 또는 그 이상의 대체 분기가 필요하다. 대체 분기에서 각 단계는 관련된 변환과 함께 활성화된다. 이로 인해 경로는 대체 자체보다 먼저 분리된다. 이 프로세스의 변환은 좌측에서 우측으로 평가된다.

대체 분기는 “main” 경로로 되돌아가거나 점프로 끝날 수 있다.

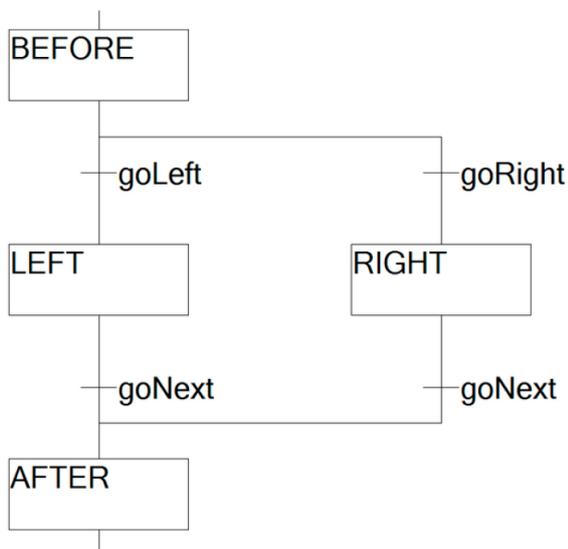


그림 19 “goNext” 변환을 통해 귀환된 대체 분기



Programming W Programs W Sequential Function Chart (SFC) W Alternative branche

5.2 병렬 분기 (Parallel branches)

프로그램을 몇 개 단계로 분할하면 명확성이 상당히 개선된다. 동시에 단계들을 실행해야 하는 경우에, SFC 에 병렬 분기를 만들 수 있다. 병렬 단계는 공통적 변환과 함께 활성화되고 비활성화된다.

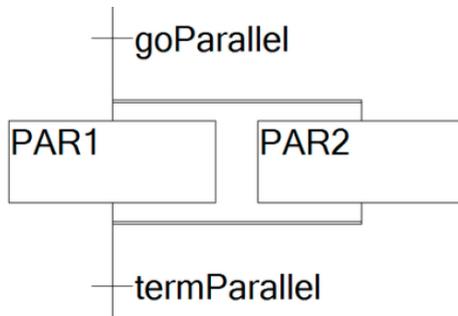


그림 20 두 병렬 스텝



병렬 단계는 좌측에서 우측으로 실행된다.



Programming W Programs W Sequential Function Chart (SFC) W Parallel branche

6 점프

점프(Jump)는 단계 시퀀스를 조금 더 유연하게 만드는 방법이다. 점프는 변환에 의해 활성화된다. 점프가 활성화되면 제어 프로그램을 다른 단계로 전송할 수 있다.

SFC 프로그램 시퀀스는 언제나 점프로 끝나고, 일반적으로 다시 단계 시퀀스로 이어진다. 대체 분기와 병렬 분기도 점프로 끝낼 수 있다.

전환으로 점프가 허용되지 않는다.

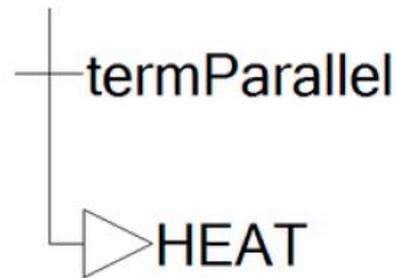


그림 21



Programming W Programs W Sequential Function Chart (SFC) W Jump
 Programming W Editors W Graphic editors W Sequential Function Chart editor W Inserting
 SFC elements

7 예제

7.1 애플리케이션 예제: 충전 제어 (Fill control)

수족관 물의 수위와 온도를 제어할 필요가 있다. 이 예제에서 상호 개별적으로 다루어질 수 있는 프로세스가 두 개 있다.

온도가 설정점에 도달할 때까지 온도를 높이기 위해 가열 요소가 사용된다. 일단 설정점에 도달되면 가열 요소는 다시 꺼진다.

충전 수위 제어는 펌프와 플로팅(floating) 스위치를 사용하여 다루어진다. 플로팅 스위치가 충전 수위가 너무 낮다는 것을 검출하면, 펌프는 적절한 충전 수위가 달성될 때까지 용기에 물을 채운다. 물결이 발생하는 경우에 펌프 전환 빈도를 줄이도록 완충 시간이 추가된다.

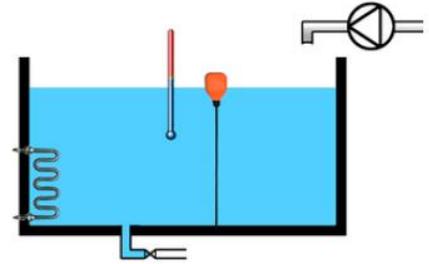


그림 22 예제 스키마

SFC 로 구현

두 가지 독립적인 시퀀스를 다루기 때문에, 동시에 실행되는 병렬 분기를 이용하는 솔루션을 개발할 수 있다. 병렬 분기의 첫 번째 부분은 가열을 제어하고, 두 번째 분기는 충전 수위 조절을 담당한다.

충전 수위 제어 구현

예제는 병렬 분기를 이용하는 SFC 프로그램으로 해결되어야 한다. 각 분기에는 상태 다이어그램에 기술된 상태가 포함된다.

- 1) SFC 프로그램을 작성한다.
- 2) 병렬 분기를 작성한다.
- 3) 단계와 점프를 작성한다.
- 4) 변환을 선언한다.
변환은 설정점 온도와 실제 온도를 평가하는데 이용될 수 있다. 그 결과는 반드시 불(BOOL) 데이터 유형이어야 (GT / LT 평선 블록, 래더 다이어그램 프로그래밍 언어) 한다.
- 5) 동작(Action)과 전환 명령을 프로그램한다.
가열 요소와 펌프의 스위치 켜짐과 스위치 꺼짐 명령은 단계 진입과 퇴출 동작을 이용하여 구현되어야 한다.
- 6) 시간 지연에 대한 IEC 동작을 이용한다.
대기 단계를 비활성화하는 변환을 "D" 한정자(qualifier)를 갖는 동작으로 전환해야 한다. 지연 시간은 5 초로 설정한다.
- 7) 변수 감시 창(variable watch window)에 변환을 설정하여 시퀀스를 시험한다.

한 가지 솔루션은 부록에서 찾아 볼 수 있다 ([“예제 솔루션: 충전 제어”](#) 참조).

7.2 애플리케이션 예제: 혼합기 제어 (Mixer control)

여기에서는 물과 페인트를 분사시켜 혼합하는 혼합 시스템을 설정한다.

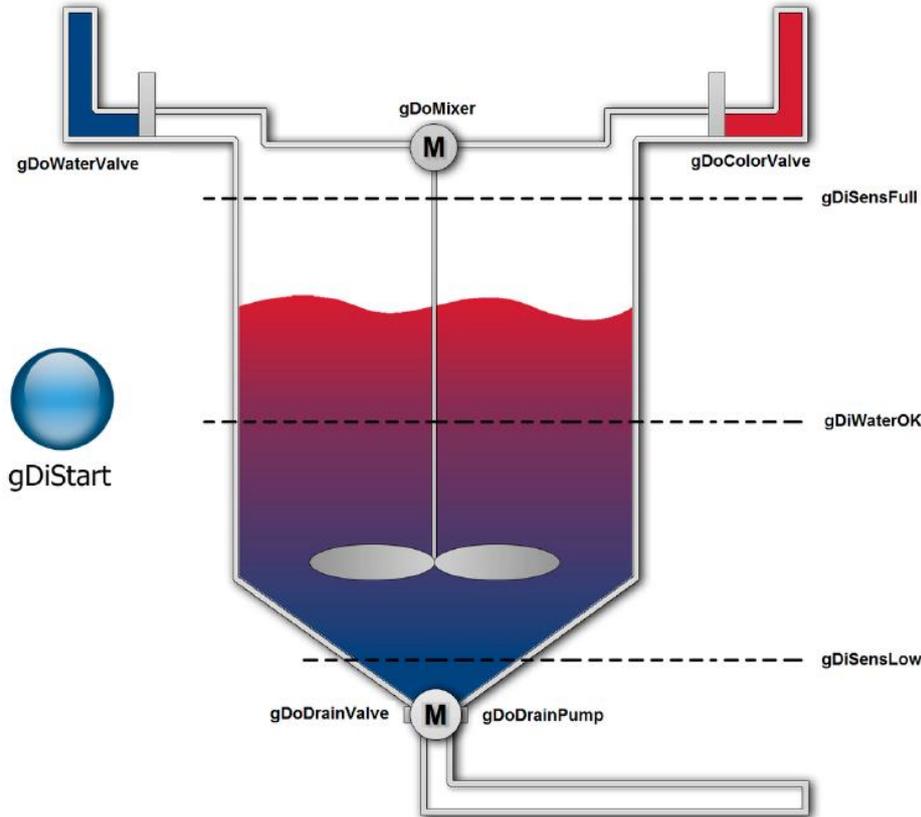


그림 23 혼합기 제어 시스템의 스키마

혼합 프로그램은 다음 절차에 따라 실행된다:

- 혼합 프로그램은 시작 버튼이 (gDiStart) 눌릴 때까지 대기한다.
- “gDiWaterOK” 센서가 반응할 때까지 용기에 물을 추가한다.
- 교반 장치 (gDoMixer) 시작되고 “gDiSensFull” 센서가 반응할 때까지 페인트가 추가된다.
- 혼합 시간은 30 초가 소요된다.
- 용기를 비우기 위해 배수구(gDoDrainValve)와 배수 펌프(gDoDrainPump)가 켜진다.
- “gDiSensLow” 신호가 반응할 때 배수 프로세스가 완료된다.
- 시작 상황이 복원된다.

혼합기 제어 구현하기

이번에는 위에 기술된 혼합 절차를 프로그램한다. 이를 위해 다음 절차를 수행한다:

- 1) 필요한 SFC 의 스케치를 작성한다.
 - 단계를 정의한다.
 - 변환을 정의한다.
 - 동작을 정의한다.
- 2) 어느 단계에 IEC 동작이 필요한지를 결정한다.

3) 이 요건들을 SFC 프로그램으로 전환한다.



교반 장치는 여러 단계에 걸쳐 활성화된다.

그 기능은 여러 가지 방식으로 구현될 수 있다:

- 교반 장치를 단계 진입/퇴출 동작으로 켜고 끈다.
- 교반 장치는 다른 단계들에 대한 병렬 단계이다.
- 교반 장치는 “S” 한정자를 갖는 동작으로 켜지고 시퀀스의 종단에서 “R” 한정자를 갖는 동작으로 꺼진다.

한 가지 솔루션은 부록에서 찾아 볼 수 있다 ([“혼합기 제어“](#) 참조).

8 시스템 변수 이용하기

SFC 시스템 변수는 시스템이 이미 “알고 있고” 이름이 주어진 변수이다. 이 변수는 프로그램에서 선언되고 이용될 수 있다. 시스템 변수는 SFC 프로그램의 현재 상태에 관한 정보를 포함하고 프로그램이 실행되는 방법에 영향을 미칠 수 있다.

예를 들어 현재 프로그램이 실행하고 있는 단계를 결정하고 프로그램을 정지시키거나 리셋하기 위해 시스템 변수를 이용할 수 있다. 시스템 변수의 시간 한도를 위반하는 단계를 알 수 있다.

시스템 변수의 전체 목록은 Automation Studio 도움말에서 찾아볼 수 있다.

시스템 변수 이용을 위한 절차:

- 도움말 시스템에서 필요한 변수를 선택한다.
- 주어진 데이터 유형으로 시스템 변수를 선언한다.
- 프로그램에서 시스템 변수를 이용한다.
- 변수 감시 창에 변수를 표시한다.



시스템 변수가 프로그램 코드에 이용되지 않으면, 컴파일러는 이들을 메모리에 저장하지 않는다. 그러면 시스템 변수를 변수 감시 창에 표시하는 것이 불가능하다.



Programming W Programs W Sequential Function Chart (SFC) W Sequential Function Chart System variables

8.1 애플리케이션에서 사용

다음은 애플리케이션에서 SFC 시스템 변수가 이용될 수 있는 몇 가지 사례를 설명한다. 예제의 일부는 SFC 프로그램에서 다양한 진단 도구 사용 방법에 대한 이해를 향상시키게 될 것이다.

시스템 변수를 이용하기 위해서는 반드시 먼저 이들을 선언하고 SFC 프로그램 코드의 어디든 추가해야 한다. 시스템 변수를 시험 목적을 위해 단순히 변수 감시 창에서 추적하거나 수정한다면, 시스템 변수를 다음 명령문으로 초기화 단계에 (Structure Text) 통합할 수 있다:

```
SFCInit;
```

SFC 프로그램 리셋하기

두 시스템 변수 “SFCInit”과 “SFCReset”는 SFC 프로그램을 리셋하기 위해 사용할 수 있다. “SFCInit” 변수가 프로그램에 존재하면 “SFCReset”은 평가되지 않는다. 두 가지 경우 모두 프로그램이 리셋되고 초기화 단계가 호출된다.

“SFCReset”을 이용하면 변수가 참(TRUE)으로 유지되는 한 초기화 단계가 사이클릭으로 호출된다.

“SFCInit”을 이용하면 초기화 단계는 변수에 하강 엣지(falling edge)가 있을 때만 한 번 호출된다.



변수 “SFCInit”과 “SFCReset”을 설정할 때는 활성 단계의 되출 동작을 호출하지 않는다. 이 변수를 설정하면 본질적으로 프로그램이 중단된다.

작업: SFC 프로그램 리셋하기

다음 예제는 혼합기 제어 예제 프로그램에 적용될 수 있다.

- 1) 불(BOOL) 데이터 유형으로 “SFCInit” 변수를 선언한다.
- 2) 프로그램 코드의 변수를 이용한다.
- 3) 프로그램 시퀀스를 시작한다.
- 4) 변수 감시 창에 시스템 변수를 설정한다.
- 5) 프로그램의 응답을 감시한다.
- 6) 프로그램으로부터 “SFCInit” 변수를 제거한다.
- 7) “SFCReset”을 선언하고 이용한다.
- 8) 다른 프로그램 반응을 비교한다.

현재의 프로그램 단계 결정하기

현재 실행되고 있는 프로그램 단계는 “SFCCurrentStep” 시스템 변수를 이용하여 읽을 수 있다. 데이터 유형은 STRING 이다.

병렬 단계를 사용하여 가장 우측 분기 이름이 저장된다.

예제: 현재 프로그램 단계 결정하기

프로그램에서 현재 실행되고 있는 단계를 결정한다.

시간 감시 이용하기

SFC 에서 단계들은 최소와 최대 “완충 시간(buffer time)”으로 설정될 수 있다. 완충 시간은 단계가 활성화되면 즉시 시작된다. 이때 시간은 단계가 비활성화되면 리셋된다. 디폴트에 의해 시간 감시는 단계에 대해 비활성화된다.

System variable	Data type	기능(Function)
SFCEnableLimit	BOOL	단계에 대한 시간 감시를 활성화한다.
SFCError	BOOL	오류의 존재 여부를 표시한다
SFCErrorPOU	STRING	오류가 발생한 프로그램을 표시한다.
SFCCurrentStep	STRING	오류를 야기한 단계를 표시한다.
SFCQuitError	BOOL	오류를 인지하고 문자열 변수를 소거한다.

그림 24 시간 위반 감지를 위한 시스템 변수

예제: 시간 감시 이용하기

프로그램 단계의 일부가 최대 1 분의 시간을 갖도록 사용자의 프로그램을 확장한다. 변수 감시창에서 시간 위반을 평가한다.

- 1) 일부 단계에 대해 최대 시간을 설정한다.
- 2) 필요한 시스템 변수를 선언한다.
- 3) 시간 감시를 활성화한다.
- 4) 시간 위반을 강제한다.
- 5) 오류를 야기하는 단계를 평가한다.



감시 모드가 활성화되면, 단계의 실행 시간이 툴팁(tooltip)에 표시된다.

9 진단 기능

프로그래밍 언어가 효율적으로 사용됨을 보장하기 위해 진단 툴이 필요하다. 이 장에서는 SFC 프로그램에 이용될 수 있는 문제 해결 툴의 간단한 목록을 제공한다.

9.1 SFC 모니터 모드

모니터 모드는 핵심 컴포넌트로서 많은 진단 툴 사용을 가능하게 한다. 이것은 툴바(toolbar)의 모니터 모드 아이콘으로 활성화 될 수 있다.

모니터 모드가 활성화 되면 다음 진단 툴이 사용가능하다:

- 파워 플로우(Powerflow)
- 변수 감시(Variable watch)
- 디버거(Debugger) ("디버거")



그림 25 모니터 모드 활성화

파워 플로우(Powerflow)

모니터 모드가 활성화 될 때마다 파워 플로우도 활성화 할 수 있다. 모든 활성 단계, 변환 및 동작을 색상으로 보여준다. 이것은 SFC 프로그램에서 현재 실행되고 있는 단계와 어느 변환으로 전환되고 있는지 손쉽게 결정할 수 있다.

여기에서 보이는 그림은 활성화된 상태의 파워 플로우이다. 활성 단계와 변환은 색상으로 표시된다. 단계의 툴 팁은 단계 상태와 함께 표시되며, 시간 감시가 활성화된 경우에는 경과된 시간과 같은 정보를 표시한다.

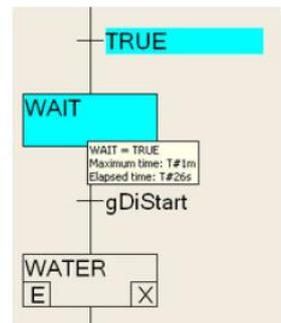


그림 26 단계를 위한 시간 감시가 활성화된 상태에서 모니터 모드의 파워 플로우

Diagnostics and service W Diagnostic tool W Monitors W Sequential Function Chart editor in monitor mode

변수 감시(Variable watch)

변수 감시 기능은 프로그램에 사용되는 변수들뿐만 아니라 SFC 시스템 변수, 단계, 동작의 인스턴스 구조를 감시할 수 있도록 한다. 또한 여기에서 변수를 설정 및 수정할 수 있다.

Diagnostics and service W Diagnostic tool W Variable watch

Name	Typ	Wert
WAIT	BOOL	FALSE
SFCenableInit	BOOL	FALSE
SFCErrorPOU	STRING(80)	"
SFCErrorStep	STRING(80)	"
MDI_TIME	SFCStepType	
x	BOOL	FALSE
t	TIME	T#0ms
_x	BOOL	FALSE
_t	TIME	T#06m_46s_480ms
bTmElapsed_action	SFCActionType	
x	BOOL	FALSE
_x	BOOL	FALSE
t	TIME	T#0ms
_t	TIME	T#0ms
AC	SFCActionControl	
N	BOOL	FALSE
R0	BOOL	FALSE
S0	BOOL	FALSE
L	BOOL	FALSE
D	BOOL	FALSE
P	BOOL	FALSE
SD	BOOL	FALSE
DS	BOOL	FALSE
SL	BOOL	FALSE
T	TIME	T#30s
Q	BOOL	FALSE
S_FF.SET	USINT	0

그림 27 변수 감시 창에서 변수 모니터링

9.2 SFC 변수

시스템 변수(System variables)

시스템 변수는 ("[시스템 변수 이용하기](#)") 프로그램 상태를 결정하는 데 이용될 수 있다. 예를 들어, SFC 프로그램을 일시 정지시키고 "스텝 인투(Step into)" 모드로 계속하거나, 시험 목적을 위해 SFC 프로그램을 리셋할 수 있다.

다음 가능성이 포함된다:

- 활성 단계 결정하기
- 변환이 활성화되어 있는지 결정하기
- 시간 감시 활성화/ 비활성화
- 시간 감시가 촉발되었는지 결정하기
- SFC 프로그램을 일시 정지시키기
- SFC 프로그램을 "스텝 인투(Step into)" 모드로 계속하기
- SFC 프로그램 리셋하기



Programming W Programs W Sequential Function Chart (SFC) W Sequential Function Chart System variables

단계, 동작 및 변환의 인스턴스 변수

컴파일러는 내부적으로 각 단계와 동작에 대한 인스턴스 변수를 생성한다. 인스턴스 변수를 변수 감시 창에 삽입하고 관찰 할 수 있다.

예를 들어, 단계 변수를 이용하여 단계가 현재 활성 상태인지 여부를 결정할 수 있다. 단계 변수 이름은 정확히 단계 자체와 동일하고 단계 유형에 따라서 불(BOOL) 데이터 유형이나 SFCStepType 데이터 유형이다. 단계가 활성 상태이면 단계 변수는 참(TRUE)으로 설정된다.

예를 들어, 동작에 대한 인스턴스 변수는 최종 스캔을 결정하는 데 ("[동작의 '최종 스캔\(final scan\)' 평가하기](#)") 이용될 수 있다. 데이터 유형은 SFCActionType 이다.



주의(WARNING)!

단계와 불 대수(Boolean) 동작에 대한 인스턴스 변수는 프로그램이 컴파일될 때 Automation Studio 에 의해 자동으로 생성된다. 시퀀스에 영향을 미치기 위해 이들 인스턴스를 조작하지 않을 것을 권장한다. 이것은 프로그램을 불명확하게 하고 오류를 발생시키는 경향이 있기 때문이다.

9.3 디버거

텍스트 프로그래밍 언어와 마찬가지로 SFC 에서 디버거도 지원된다. 디버거는 단계에 중단점 (breakpoint) 지정을 허용한다. SFC 프로그램이 중단점을 갖는 단계를 스캔하면, SFC 프로그램이 일시 정지된다.

디버거가 닫히면 SFC 는 사이클릭 실행을 계속한다.

모든 동작은 디버거 툴바와 메시지 창을 이용하여 제어될 수 있다.



그림 28 디버거 툴바

중단점(breakpoint) 설정하기

모니터 모드가 활성화되면 디버거 툴바를 이용하여 여러 개 중단점을 단계에 설정할 수 있다. 단계 중에는 디버거가 활성화되지 않는다.

이미 설정된 중단점은 단계에서 녹색 삼각형으로 표시된다.



그림 29 중단점 설정

디버거 활성화하기

디버거 툴바에서 해당 아이콘을 클릭함으로써 디버거를 가능화 할 수 있다. 프로그램이 실행되고 중단점 중 하나에 도달하면 프로그램이 일시 정지된다.

프로그램이 일시 정지되었던 단계는 황색 삼각형으로 표시된다.

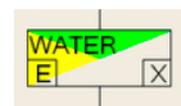


그림 30 프로그램에 도달한 중단점



Diagnostics and service W Diagnostic tool W Debugger

예제: 디버거 시험하기

최근 SFC 프로그램에 중단점을 설정하고 디버거 기능을 시험한다.

- 1) 프로그램을 연다.
- 2) 모니터 모드를 시작한다.
- 3) 중단점을 설정한다.
- 4) 디버거를 활성화한다.
- 5) 메시지 창의 출력을 관찰한다.

10 요약

Sequential Function Chart 프로그래밍 언어는 시각적 인터페이스와 함께 매우 잘 구성된 직선적인 언어이다. 이 언어는 단계와 변환의 형태로 상태 다이어그램과 기계 상태 프로그램을 표시하는 탁월한 방법을 제공한다.

프로그램이 순차적으로 실행되고 다양한 진단 툴이 제공되기 때문에, SFC 프로그램 유지와 문제 해결은 매우 용이하다.

SFC 프로그램으로 정의된 인터페이스를 통해 다른 제어 애플리케이션의 프로세스를 호출하는 것도 가능하다. 이것은 시퀀스와 호출된 하위 기능을 명확하고 용이하게 시험할 수 있도록 한다.

SFC 는, 래더 다이어그램이나 Structured text 와 같은 다른 IEC 프로그래밍 언어와 조합될 때, 매우 강력한 언어가 된다.

IEC 동작을 이용하는 동작 블록은 다양한 한정자와 함께 SFC 에서 이용 할 수 있도록 다양한 기능 범위를 제공한다.

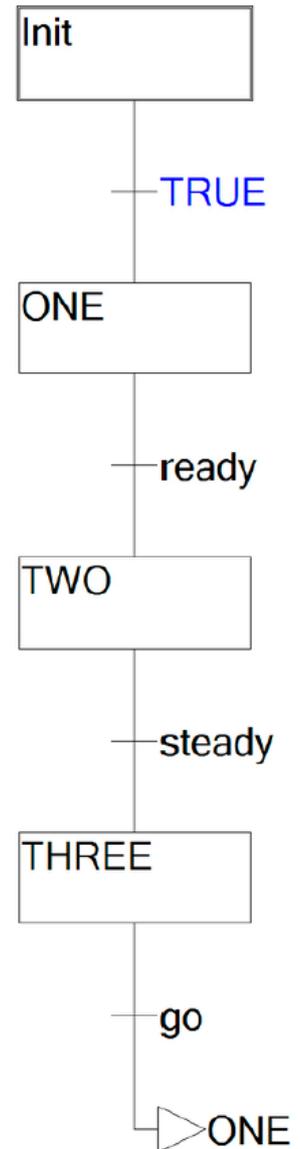


그림 31 SFC 프로그래밍 언어

11 부록: 예제 솔루션

11.1 예제 솔루션: 충전 제어 (Fill control)

“애플리케이션 예제: 충전 제어”는 다음과 같이 해결될 수 있다:

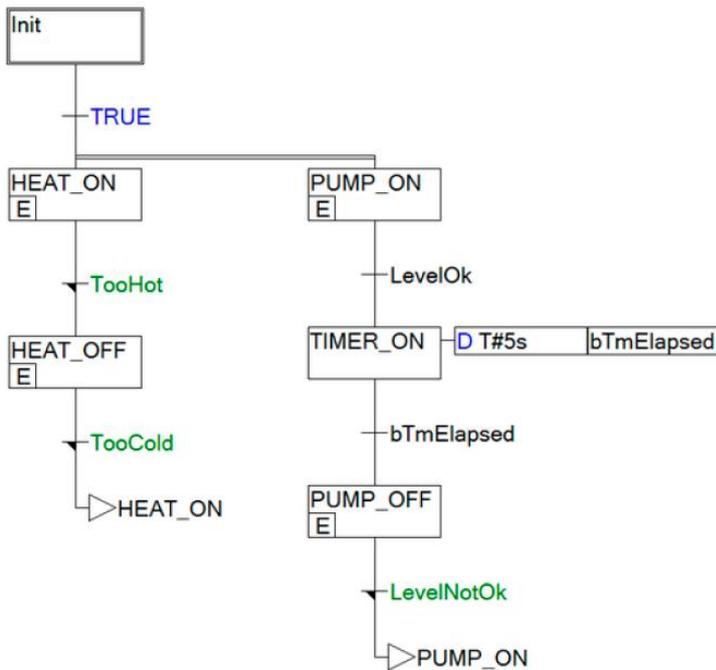


그림 32 가능한 해결책 - 병렬 분기 실행

변환의 내용

“TooHot” 변환은 설정점과 실제 온도 값을 비교하는 래더 다이어그램과 같은 것을 포함할 수 있다. 실제 온도가 설정 온도보다 높으면 변환이 전환된다.

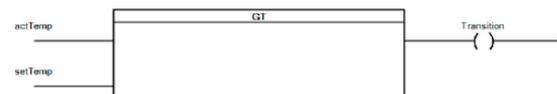


그림 33 래더 다이어그램에 “TooHot” 변환

주요 병렬 분기는 상태 다이어그램 두 개에 대응된다.

“HEAT_ON”과 “HEAT_OFF” 사이 변환에는 설정 온도와 실제 온도를 비교하는 프로그램 코드가 포함된다.

가열 장치를 켜고 끄기 위한 전환 명령은 가열 단계의 진입 동작으로 구현된다.

펌프 분기는 지연 시간에 대한 “D” 한정자만 갖는 IEC 동작을 이용하고 동작의 다음 단계에서 변환을 전환한다.

“PUMP_OFF”의 입력 동작은 펌프 출력 뿐만 아니라 경과 시간의 전환을 리셋한다.

11.2 예제 솔루션: 혼합기 제어 (Mixer control)

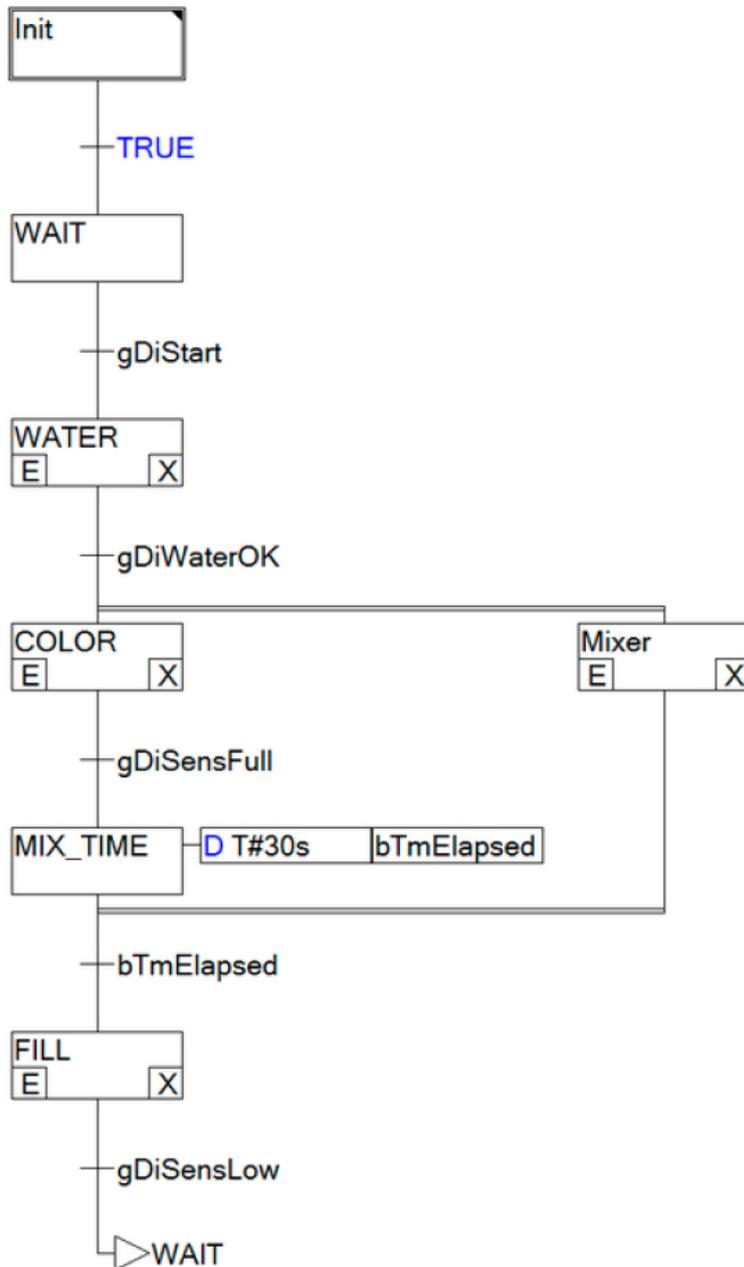


그림 34 혼합기 제어의 가능한 해결책

혼합기 제어 시스템 시퀀스는 (“[애플리케이션 예제: 혼합기 제어](#)”) 여기에서 제공된 그림과 같이 해결될 수 있다.

혼합기 제어를 위한 단계 하나에서 페인트를 충전하고 혼합 시간을 다루기 위한 병렬 분기로 이용한다.

진입과 퇴출 동작은 각각 밸브와 모터와 같은 액추에이터를 전환하기 위해 사용될 수 있다.

시간 감시를 위해, Boolean 동작 “bTmElapsed”를 한정자 “D T#30s”와 함께 호출하고 다음 단계에 대한 변환으로 이용한다.

Automation Academy 에서 제공하는 것

우리는 고객뿐만 아니라 직원을 대상으로 한 교육 과정을 제공합니다.

Automation Academy 에서, 당신은 필요로 하는 능력을 즉시 향상시킬 수 있습니다.

자동화 엔지니어링 분야에서 필요로 하는 지식증진을 위해 세미나가 준비되어 있습니다. 한번 이수하면, 당신은 B&R 기술을 이용하여 능률적인 자동화 솔루션을 개발하는 위치에 있을 것입니다.

이를 통해 귀하와 귀사는 끊임없이 변화하는 시장 수요에 보다 빠르게 대응할 수 있게 됨으로써 결정적인 경쟁 우위를 확보 할 수 있습니다.



세미나



품질 및 관련성은 세미나의 필수 구성 요소입니다. 특정 세미나의 페이스는 엄격하게 코스 참가자가 직면한 요구 사항과 경험에 근거합니다. 그룹 스터디와 자율 학습에 조합은 학습 경험을 극대화하는데 필요한 높은 수준의 유연성을 제공합니다. 각 세미나는 숙련된 경험이 풍부한 강사 중 한 명이 진행합니다.

교육 자료(Training module)

교육 자료는 세미나뿐만 아니라 자율 학습을 위한 기초를 제공합니다. 컴팩트 모듈은 일관된 교육 개념에 의존합니다. 상황식 구조는 복잡하고 상호 연관된 주제를 효율적이고 효과적으로 배울 수 있습니다. 광범위한 도움말 시스템이 가장 좋은 보완책입니다. 교육 자료는 다운받을 수 있으며 인쇄된 버전으로 주문할 수 있습니다.

카테고리 주제:

- 제어 기술(Control technology)
- 모션 제어(Motion control)
- 세이프티 기술(Safety technology)
- 화면작화(HMI)
- 프로세스 컨트롤(Process control)
- 진단 및 서비스(Diagnostics and service)
- 파워링크와 오픈세이프티 (POWERLINK and openSAFETY)

ETA 시스템(ETA system)



ETA 시스템(ETA system)은 훈련, 교육 및 실험실에서 사용하기 위해 실제와 같은 구조를 제공합니다. 두가지 이상의 다른 기구 구조가 선택될 수 있습니다. ETA light system 은 높은 자유도, 공간 절약 및 연구소 작업에 적합합니다. ETA standard system 은 튼튼한 기구 구조와 사전에 와이어링된 센서와 액추에이터를 포함합니다.

더 알아보기!

추가적인 교육이 필요하시나요? B&R Automation Academy 가 제공하는 것에 흥미가 있으신가요? 맞게 찾아오셨습니다.

상세한 정보는 아래 링크에서 확인하실 수 있습니다:

www.br-autoation.com/academy

