

TPT The logo graphic for TPT consists of a thick grey wave-like shape that curves upwards and then downwards. A red curved line is positioned above the grey wave, following its upper contour.

Time Partition Testing

Beginner Training

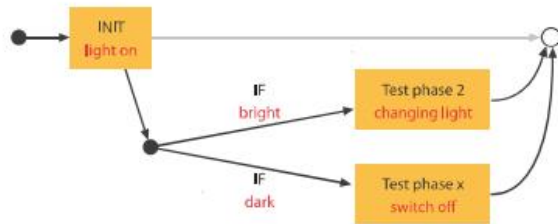
목차

- I. TPT Overview
- II. TPT Platform 구성
- III. TPT Test Modeling
- IV. TPT Test Execution

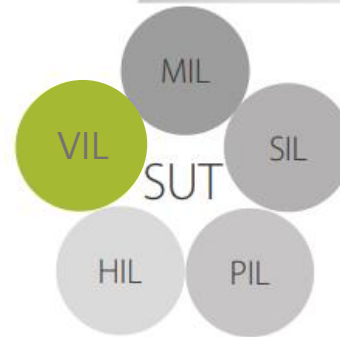
TPT Overview

TPT Overview

1. Test Modeling



2. Test Execution



Test execution:

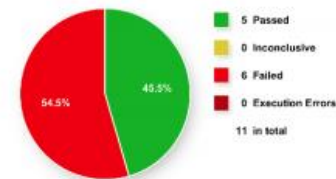
- MATLAB Simulink/ASCET
- C/C++, Silver
- AUTOSAR
- dSPACE / ETAS LABCAR HiL
- ECU via Vector CAN/CANape/INCA
- FUSION

3. Test Assessment

Specify interval for check by	
Condition	light_switch == 1
Ignore interval if	
Check is fulfilled	<input type="radio"/> Once <input checked="" type="radio"/> Always
Then check	#1: headlight == 1
Else check	#1: true

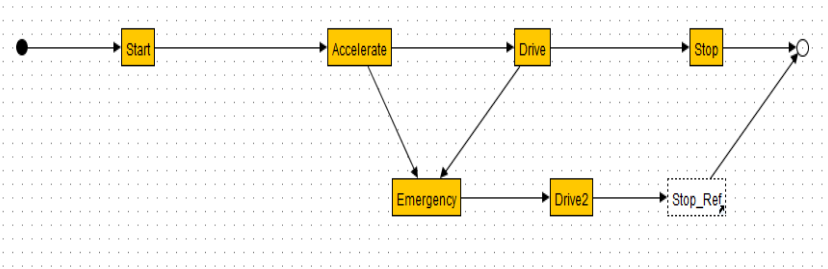


4. Test Reporting



TPT Overview

◆ 시나리오 기반 모델링



Test Cases [135]	
TC1:	Region :0 varia, Accelerate_Spee, Driving_light_o, Iginition_On_On [IC
TC2:	Region :0 varia, Accelerate_Spee, Driving_light_o, Iginition_On_On [IC
TC3:	Region :0 varia, Accelerate_Spee, Driving_light_o, Iginition_On_On [IC
TC4:	Region :0 varia, Accelerate_Spee, Driving_light_o, Iginition_On_On [IC
TC5:	Region :0 varia, Accelerate_Spee, Driving_light_o, Iginition_On_On [IC
TC6:	Region :0 varia, Accelerate_Spee, Driving_light_o, Iginition_On_On [IC
TC7:	Region :0 varia, Accelerate_Spee, Driving_light_A, Iginition_On_On [IC
TC8:	Region :0 varia, Accelerate_Spee, Driving_light_A, Iginition_On_On [IC
TC9:	Region :0 varia, Accelerate_Spee, Driving_light_A, Iginition_On_On [IC
TC10:	Region :0 varia, Accelerate_Spee, Driving_light_o, Iginition_On_AU [
TC11:	Region :0 varia, Accelerate_Spee, Driving_light_o, Iginition_On_AU [
TC12:	Region :0 varia, Accelerate_Spee, Driving_light_o, Iginition_On_AU [
TC13:	Region :0 varia, Accelerate_Spee, Driving_light_o, Iginition_On_AU [
TC14:	Region :0 varia, Accelerate_Spee, Driving_light_o, Iginition_On_AU [
TC15:	Region :0 varia, Accelerate_Spee, Driving_light_o, Iginition_On_AU [

◆ Step List

Step	Channel	Value	Condition	Frequency	Options
1	Channel	light_switch	:= AUTO_MODE	Always	
2	Channel	light_Intensity	:= 0	Always	
3	Wait	10ms			
4	While ...	light_Intensity <= 100			
5	If ...	light_Intensity < 60		Once	
6	Compare	headlight	== 1	with tolerance +/- 0	from scaling
7	End (If)				
8	If ...	light_Intensity > 70		Once	
9	Compare	headlight	== 0	with tolerance +/- 0	from scaling
10	End (If)				
11	Channel	light_Intensity	:= light_Intensity + 1	Always	
12	Wait	1			
13	End (While)				

TPT Platform 구성

TPT Platform 구성(C Platform)

◆ Compiler 환경 설정(MinGW)

① 아래 사이트에서 PC에 설치된 OS 환경에 따라 다운 받아 Compiler 설치

<https://sourceforge.net/projects/mingw-w64/>

② 설치된 Compiler의 실행 파일 경로를 환경 변수의 path에 추가

시스템 속성

컴퓨터 이름 하드웨어 고급 시스템 보호 원격

이 내용을 변경하려면 관리자로 로그인해야 합니다.

성능
시각 효과, 프로세서 일정, 메모리 사용 및 가상 메모리
설정(S)...

사용자 프로필
사용자 로그인에 관련된 바탕 화면 설정
설정(E)...

시작 및 복구
시스템 시작, 시스템 오류 및 디버깅 정보
설정(T)...

환경 변수(N)...

확인 취소 적용(A)

환경 변수 편집

C:\Program Files\Microsoft SQL Server\120\Tools\Binn#\
C:\Program Files (x86)\Microsoft Emulator Manager\1.0#\
C:\Program Files (x86)\nodejs#\
C:\Python36\Scripts
C:\Users\JungSub.Kuk\cpp-2019-12\ eclipse
C:\Program Files\TPT-15u2
C:\Program Files (x86)\IVI Foundation\VISA\WinNT\Bin#\
C:\Program Files\IVI Foundation\VISA\Win64\Bin#\
C:\Program Files (x86)\IVI Foundation\VISA\WinNT\Bin#\
C:\Program Files\TortoiseSVN\bin
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\140\WT...
C:\Program Files (x86)\Microsoft SQL Server\120\Tools\Binn#\
C:\Program Files\Microsoft SQL Server\120\DT\Binn#\
C:\Program Files\Microsoft SQL Server\130\Tools\Binn#\
C:\Program Files\Microsoft SQL Server\130\Tools\Binn#\
C:\Program Files (x86)\Microsoft SQL Server\150\DT\Binn#\
C:\Program Files (x86)\Vector CANdb++ 3.1\Exec32
C:\Program Files\MATLAB\R2018a\bin#\
C:\dev\apache-maven-3.6.3\bin
C:\Program Files\Git\cmd
C:\mingw64\bin

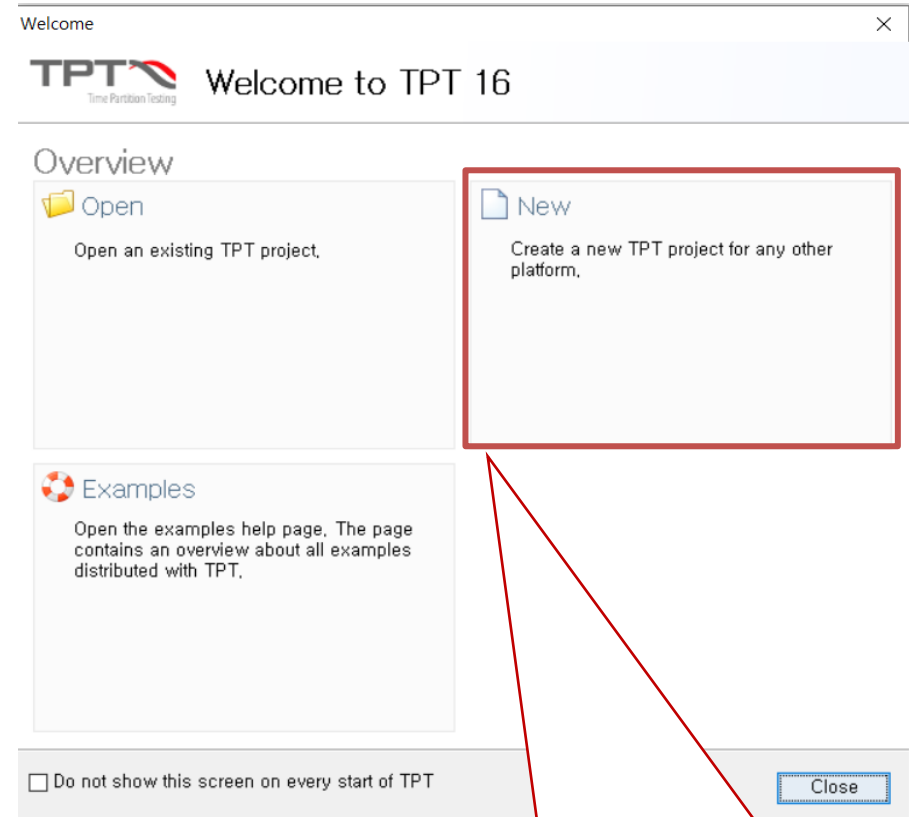
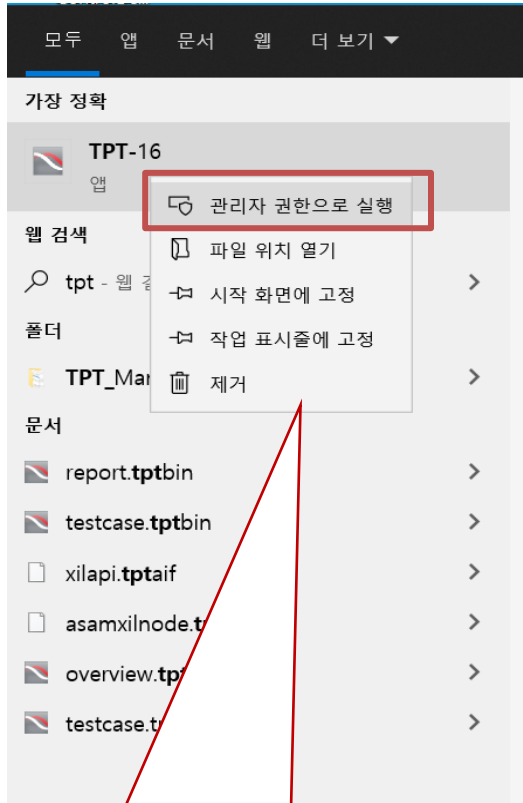
새로 만들기(N)
편집(E)
찾아보기(B)...
삭제(D)
위로 이동(U)
아래로 이동(O)
텍스트 편집(T)...

확인 취소

시스템 속성에서 환경 변수 버튼을 클릭하여 나온 환경 변수 창에서 "Path"를 클릭 한 후 새로 만들기 버튼을 눌러 설치한 MinGW Compiler의 실행 파일 경로를 추가 한다.

TPT Platform 구성(C Platform)

◆ TPT Platform Configuration 실행

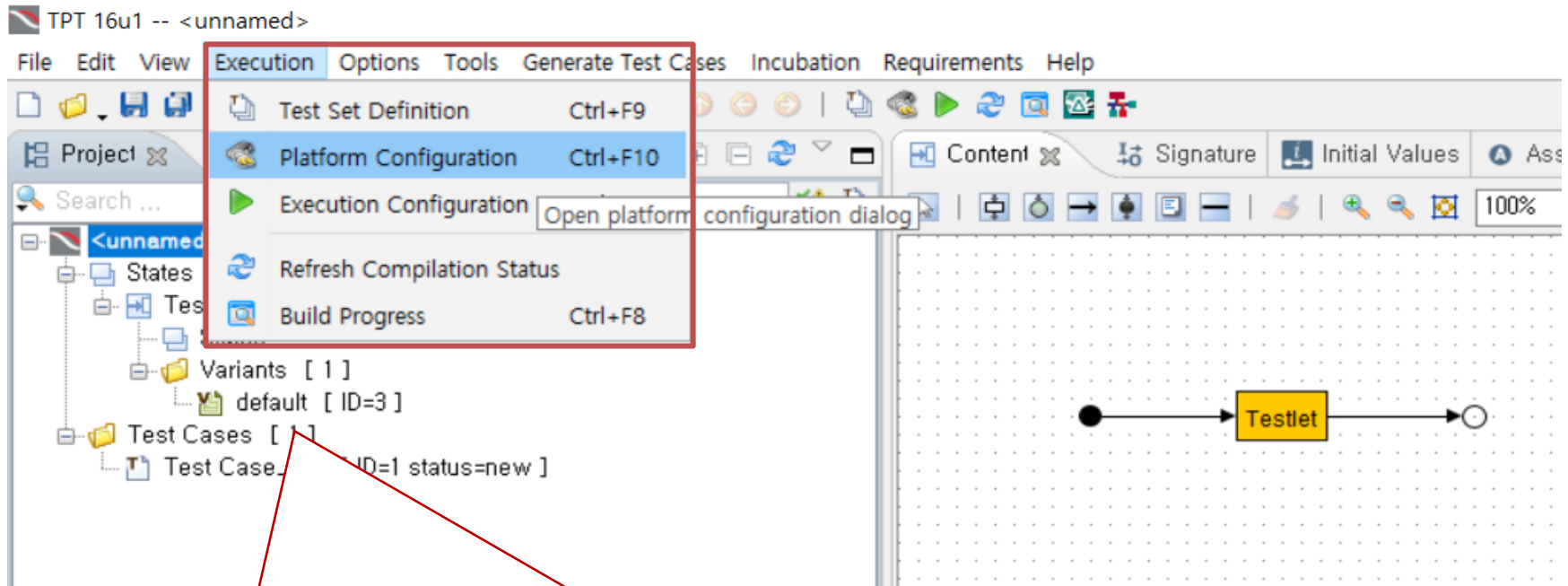


① Window 시작 -> TPT를 검색 한 후 마우스 우 클릭을 통해 나온 팝업 창에서 관리자 권한으로 TPT를 실행 시킨다.

② TPT 실행 후 나온 Welcome 창에서 New버튼을 클릭 한다.

TPT Platform 구성(C Platform)

◆ TPT Platform Configuration 실행



③ New 버튼 클릭 시 unnamed 상태로 TPT 프로젝트가 생성 된다. 메뉴바에서 “Execution -> Platform Configuration” 을 클릭 한다.

※ Windows 10일 경우, TPT에서 제공 하는 C Code Demo 파일을 C:\Program Files\TPT-16u1\examples\SUT-C-Code Simple에서 다른 경로로(Ex, 바탕화면 또는 D:\W) 복사한다.

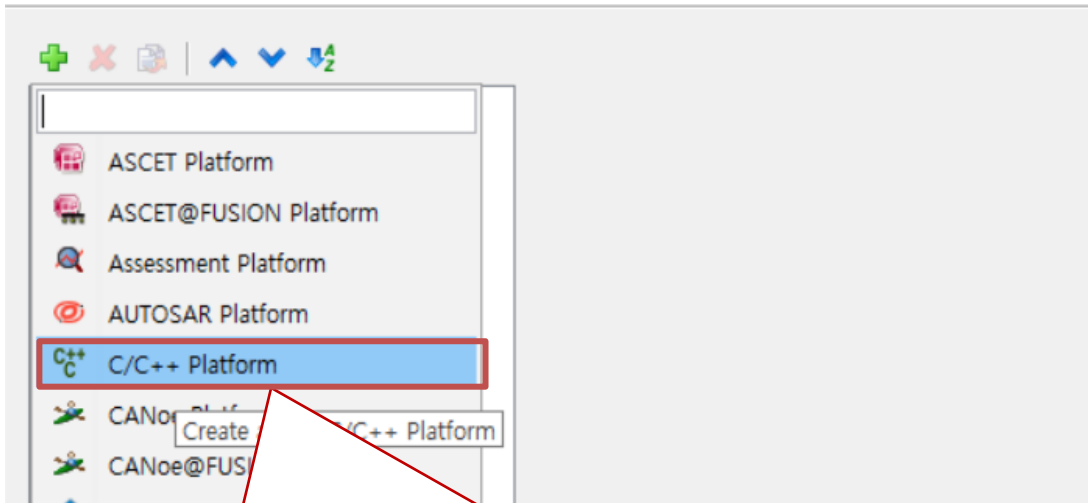
TPT Platform 구성(C Platform)


◆ TPT Platform Configuration 실행

Platform Configuration

Platform Configuration

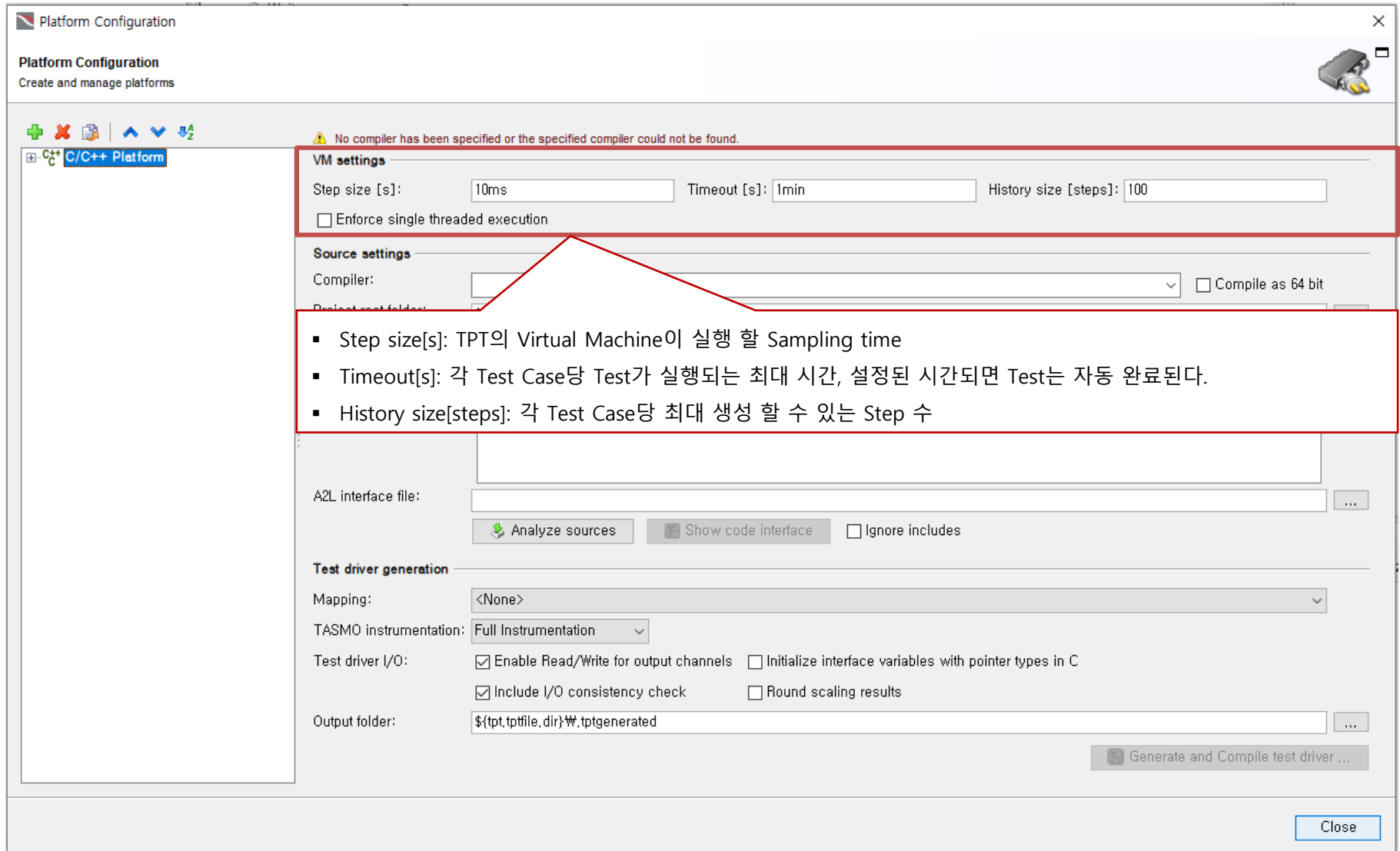
Create and manage platforms



- ④ 메뉴바에서 "Execution -> Platform Configuration" 을 클릭 후 나온 Platform Configuration 창에서  아이콘을 클릭하여 나온 Platform Type 들 중 C/C++ Platform을 클릭한다.

TPT Platform 구성(C Platform)

◆ TPT Platform Configuration 설정 – VM settings



Platform Configuration

Platform Configuration
Create and manage platforms

No compiler has been specified or the specified compiler could not be found.

VM settings

Step size [s]: 10ms Timeout [s]: 1min History size [steps]: 100

Enforce single threaded execution

Source settings

Compiler: Compile as 64 bit

- Step size[s]: TPT의 Virtual Machine이 실행 할 Sampling time
- Timeout[s]: 각 Test Case당 Test가 실행되는 최대 시간, 설정된 시간되면 Test는 자동 완료된다.
- History size[steps]: 각 Test Case당 최대 생성 할 수 있는 Step 수

A2L interface file: ...

Analyze sources Show code interface Ignore includes

Test driver generation

Mapping: <None>

TASMO instrumentation: Full Instrumentation

Test driver I/O: Enable Read/Write for output channels Initialize interface variables with pointer types in C
 Include I/O consistency check Round scaling results

Output folder: ...

Generate and Compile test driver ...

Close

TPT Platform 구성(C Platform)

◆ TPT Platform Configuration 설정 – Source settings

Platform Configuration

Platform Configuration
Create and manage platforms

⚠ No compiler has been specified or the specified compiler could not be found.

VM settings

Step size [s]: 10ms Timeout [s]: 1min History size [steps]: 100

Enforce single threaded execution

Source settings

Compiler: [Dropdown menu open]

- Default MinGW Installation
- MinGW 4.8.1
- Visual Studio 2010
- Visual Studio 2012
- Visual Studio 2013
- Visual Studio 2015

Compile as 64 bit

Project root folder: [Text field]

Sources: [List box]

A2L interface file: [Text field]

Ignore includes

Output folder: \${tpt,tptfile.dir}*.tptgenerated

Generate and Compile test driver ...

Close

Compiler Dropdown 버튼을 클릭하여 C Platform에서 사용할 Compiler를 선택 한다.

※ 설치된 MinGW Compiler 버전이 하나 일 경우, Default MinGW Installation을 선택 하면 된다.

TPT Platform 구성(C Platform)

◆ TPT Platform Configuration 설정 – Source settings

The screenshot shows the 'Platform Configuration' dialog box for a C/C++ Platform. The 'Source settings' section is highlighted, showing the 'Project root folder' field with the value `${tpt,tpfile,dir}*` and a browse button (...). A 'Select' dialog box is open over this field, showing the file explorer view of the 'SUT-C-Code Simple' folder, with the 'Folder name' field containing `C:\Users\JungSub, Kuk\Desktop\SUT-C-Code Simple`.

Project root folder에서 ... 버튼 클릭하여 나온 Project root folder 선택 창에서 Project folder를 선택 한다

Windows 10일 경우 Source Code 파일을 C:\Programfiles\가 아닌 다른 경로로 복사해 두어야 한다.(보안 문제)

TPT Platform 구성(C Platform)

◆ TPT Platform Configuration 설정 – Source settings

The screenshot shows the 'Platform Configuration' dialog for a C/C++ Platform. The 'Source settings' section is active, displaying the following configuration:

- Compiler: MinGW 4,8,1
- Project root folder: \${tpt,tptfile,dir}\#SUT-C-Code Simple
- Sources: A list with a 'Select C Source Files' button highlighted by a red box.

A 'Select C Source Files' dialog is open, showing a file tree with the following items:

- SUT-C-Code Simple [Active: 1 / 4]
- .tptgenerated [Active: 0 / 3]
- lights_control_no_scaling.c (checked)

Red lines connect the 'Select C Source Files' button in the main dialog to the file selection dialog. A text box at the bottom left provides instructions in Korean.

Sources 항목의 Select C Source Files 아이콘()을 클릭하여 나온 창에서 TPT에서 실행 할 모든 C 파일들을 체크 한 후 OK 버튼을 클릭 한다.

TPT Platform 구성(C Platform)

◆ TPT Platform Configuration 설정 – Source settings

The screenshot displays the 'Platform Configuration' dialog box with the 'Source settings' tab selected. The 'Compiler' is set to 'MinGW 4,8,1' and the 'Project root folder' is '\$\{tpt,tpfile,dir\}\SUT-C-Code Simple'. The 'Sources' list contains 'lights_control_no_scaling.c'. The 'Output folder' is '\$\{tpt,tpfile,dir\}\.tptgenerated'. A red box highlights the 'Analyze sources' button. A red callout box with text explains that clicking 'Analyze source' triggers a warning if the output folder is invalid. A 'Select' dialog box is open, showing the 'output' folder selected in the 'SUT-C-Code Simple' directory. A red callout box with text instructs to click the 'Select' button after choosing the output folder. A 'Warning' dialog box is also visible, stating 'The selected output directory is invalid, Log file won't be written.' with 'OK' and 'Details' buttons.

① Analyze source 버튼을 클릭 시 다음과 같이 메시지 창이 팝업 되면 하위 항목 중 Output folder에서 ... 버튼을 클릭하여 Output folder를 지정한다.

② Output folder 지정 후 Select 버튼을 클릭 한다

TPT Platform 구성(C Platform)

◆ TPT Platform Configuration 설정 – Source settings

The image shows two windows from the TPT software. The left window is 'Platform Configuration' and the right is 'Code interface'. Both windows have a warning message: 'The C code configuration was not imported, please extract the interface.' In the 'Platform Configuration' window, the 'Analyze sources' button is highlighted with a red box. In the 'Code interface' window, the 'Import interface' button is highlighted with a red box. A table in the 'Code interface' window lists various functions and variables with their details and connection types.

① Analyze source 버튼을 클릭하여 나온 Code interface 창에서 TPT에서 실행 할 함수 및 변수를 확인 한다.

※ C 파일안의 전역 변수 및 함수의 Details 부분에 connect 부분을 ignore로 변경 시 TPT는 ignore로 설정된 전역 변수 및 함수는 TPT에서 사용되어 지지 않는다.

Name	Details
(v) lights_control_no_scaling.c	
(v) AUTO_MODE: uint8	connect
(v) HYSTERESE_TIME_OFF: double	connect
(v) HYSTERESE_TIME_ON: double	connect
(v) MIN_LIGHT_OFF: double	connect
(v) MIN_LIGHT_ON: double	connect
(v) ON_MODE: uint8	connect
(v) ^S STEPSIZE: double	
(v) headlight: boolean	connect
(v) light_intensity: double	connect
(v) light_switch: uint8	connect
(v) ^S light_switch_old: uint8	
(v) ^S off_counter: uint16	
(v) ^S on_counter: uint16	
(f) ^u lights_control_cycle(): void	schedule
Custom Wrapper Code	
unresolved references	

② Import interface 버튼을 눌러 connect, schedule된 변수 및 함수들을 import 시킨다.

TPT Platform 구성(C Platform)

◆ TPT Platform Configuration 설정 – Source settings

① Import interface 버튼을 클릭하여 나온 import interface 창의 Default All을 클릭하여 TPT에서 사용할 interface로 import 시킨다.

② headlight 시그널의 Mode를 OUT → IN으로 변경 한다.

Status	Action	Rename	Name	Data type	Value	Mode
New	Ignore		o AUTO_MODE	uint8		2,OUT
New	Ignore		o headlight	uint8		0,OUT
New	Ignore		o HYSTERESE_TIME_OFF	double		3,0,OUT
New	Ignore		o HYSTERESE_TIME_ON	double		2,0,OUT
New	Ignore		o light_intensity	double		0,0,OUT
New	Ignore		o light_switch	uint8		0,OUT
New	Ignore		o MIN_LIGHT_OFF	double		70,0,OUT
New	Ignore		o MIN_LIGHT_ON	double		60,0,OUT
New	Ignore		o ON_MODE	uint8		1,OUT

TPT Platform 구성(C Platform)

◆ TPT Platform Configuration 설정 – Source settings

The screenshot displays the 'Code interface' window with the following elements:

- Title Bar:** Code interface
- Header:** Select functions and variables to connect to TPT. Functions can be connected with TPT in different ways. You can either schedule them which means that they will be periodically called within the generated test frame or you can import them and call them within your steplists.
- Warning:** The C code configuration was not imported, please extract the interface.
- Search:** Search ...
- Buttons:** S (v), (f), u
- Table:**

Name	Details
lights_control_no_scaling.c	
(v) AUTO_MODE: uint8	connect
(v) HYSTERESE_TIME_OFF: double	connect
(v) HYSTERESE_TIME_ON: double	connect
(v) MIN_LIGHT_OFF: double	connect
(v) MIN_LIGHT_ON: double	connect
(v) ON_MODE: uint8	connect
(v) ^S STEPSIZE: double	connect
(v) headlight: boolean	connect
(v) light_intensity: double	connect
(v) light_switch: uint8	connect
(v) ^S light_switch_old: uint8	connect
(v) ^S off_counter: uint16	connect
(v) ^S on_counter: uint16	connect
(f) ^u lights_control_cycle(): void	schedule
Custom Wrapper Code	
unresolved references	
- Notification Dialog:** Scheduling functions updated. The functions list for scheduling was updated: Added 1 function. Includes OK, Details, and Don't show this message again options.
- Annotations:** A red box highlights the 'lights_control_cycle(): void' function in the table. A red arrow points from this box to the 'OK' button in the notification dialog. Another red arrow points from the 'OK' button to a text box.

import interface창에서 OK버튼 클릭 시 Source 분석 시 schedule로 분석된 함수는 TPT VM의 Scheduler에 자동 update된다.

TPT Platform 구성(C Platform)

◆ TPT Platform Configuration 설정 – Test driver generation

The screenshot displays the TPT Platform Configuration window for a C/C++ platform. The 'Test driver generation' section is active, showing the following settings:

- Step size [s]: 10ms
- Compiler: MinGW 4.8.1
- Project root folder: `${tpt,tpfile,dir}\SUT-C-Code Simple`
- Output folder: `${tpt,tpfile,dir}\tptgenerated`
- Options: Include I/O consistency check, Round scaling results

A 'Generate and Compile test driver ...' button is highlighted in the bottom right. A red dashed arrow points from this button to a file explorer window showing the contents of the 'output' folder:

이름	수정한 날짜	유형
bin	2021-03-11 오전 9:16	파일 폴더
build_output.log	2021-03-11 오전 9:16	텍스트 문서
lights_control_no_scaling.c	2021-03-11 오전 9:16	C 파일
linkerResponse.rsp	2021-03-11 오전 9:16	RSP 파일
make.bat	2021-03-11 오전 9:16	Windows 배치 파일
TPT_lights_control_no_scaling.c	2021-03-11 오전 9:16	C 파일
TPT_TestDriver.c	2021-03-11 오전 9:16	C 파일
TPT_TestDriver.h	2021-03-11 오전 9:16	C/C++ Header
TPT_TestDriver.tptcovmeta	2021-03-11 오전 9:16	TPTCOVMETA 파일

A dialog box titled 'Test driver generation completed' is shown in the center, with the message 'Test driver generation completed.' and an 'OK' button. A red dashed arrow points from the dialog box back to the configuration window.

Two callout boxes provide instructions:

- ① Generate and Compile test driver 버튼 클릭 시 Compiler 설정이 맞게 동작된다면, Test Driver 생성 완료 팝업창이 발생된다.
- ② Output folder로 지정한 폴더안에 TPT TestDriver 관련 파일들이 자동 생성된 것을 확인 할 수 있다.

TPT Platform 구성(C Platform)

◆ TPT Platform Configuration 설정 – Test driver generation

The screenshot shows the 'Platform Configuration' dialog box for a C/C++ Platform. The 'Source settings' section is active, showing 'Compiler: MinGW 4.8.1' and a checked 'Compile as 64 bit' option. An 'Error' dialog box is overlaid, stating: 'Compiling Test driver failed. Please review the make-script located at "C:\Users\JungSub, Kuk\Desktop\SUT-C-Code Simple\Output\make.bat" and execute it manually.' The 'OK' button in the error dialog is highlighted with a red dashed box. A red dashed arrow points from the 'OK' button to the 'Compile as 64 bit' checkbox. Another red dashed arrow points from the 'Generate and Compile test driver ...' button at the bottom right of the main dialog to the error dialog. A text box at the bottom of the main dialog contains the following text: 'Test Driver 생성 중 위와 같은 오류 발생 시, "Compile as 64 bit" 체크 박스를 체크 한 후 다시 "Generate and Compile test driver..." 버튼을 클릭한다.'

Platform Configuration
Create and manage platforms

Some settings have been changed. The test frame needs to be recompiled.

VM settings
Step size [s]: 10ms Timeout [s]: 1min History size [steps]: 100
 Enforce single threaded execution

Source settings
Compiler: MinGW 4.8.1 Compile as 64 bit

Extra Options Analy... Head...
 ...

Output folder: \${tpt,file,dir}\tptgenerated

Generate and Compile test driver ...

Close

Error
Compiling Test driver failed.
Please review the make-script located at
"C:\Users\JungSub, Kuk\Desktop\SUT-C-Code Simple\Output\make.bat"
and execute it manually.

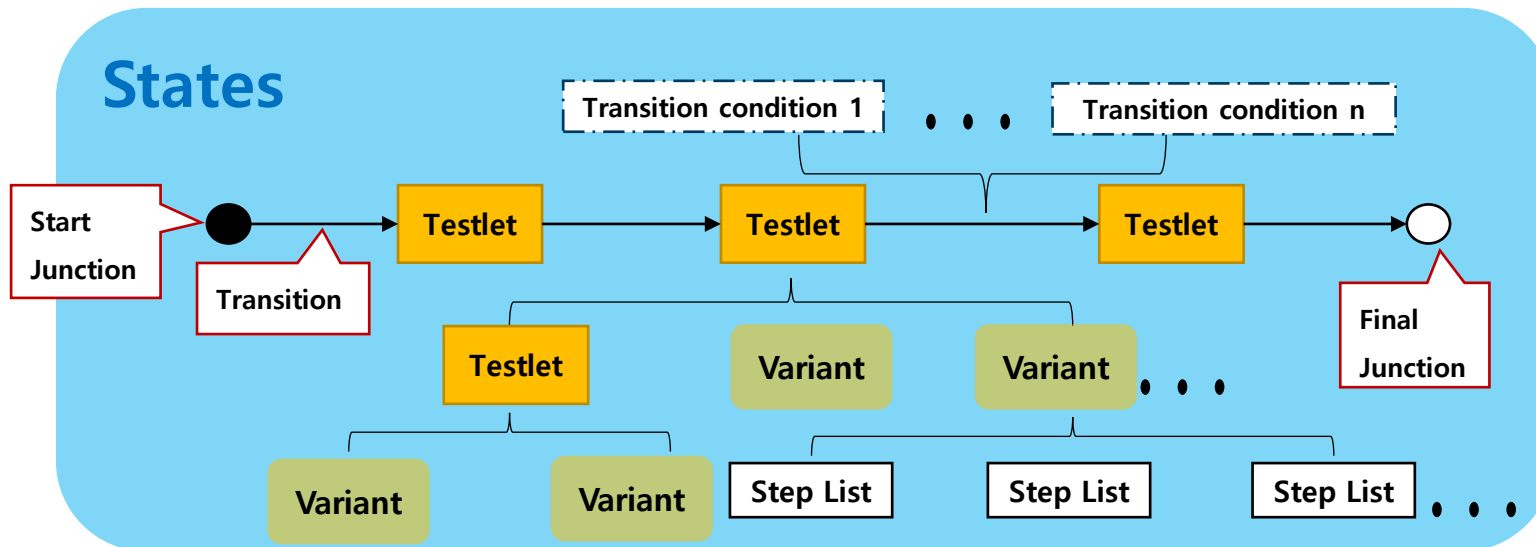
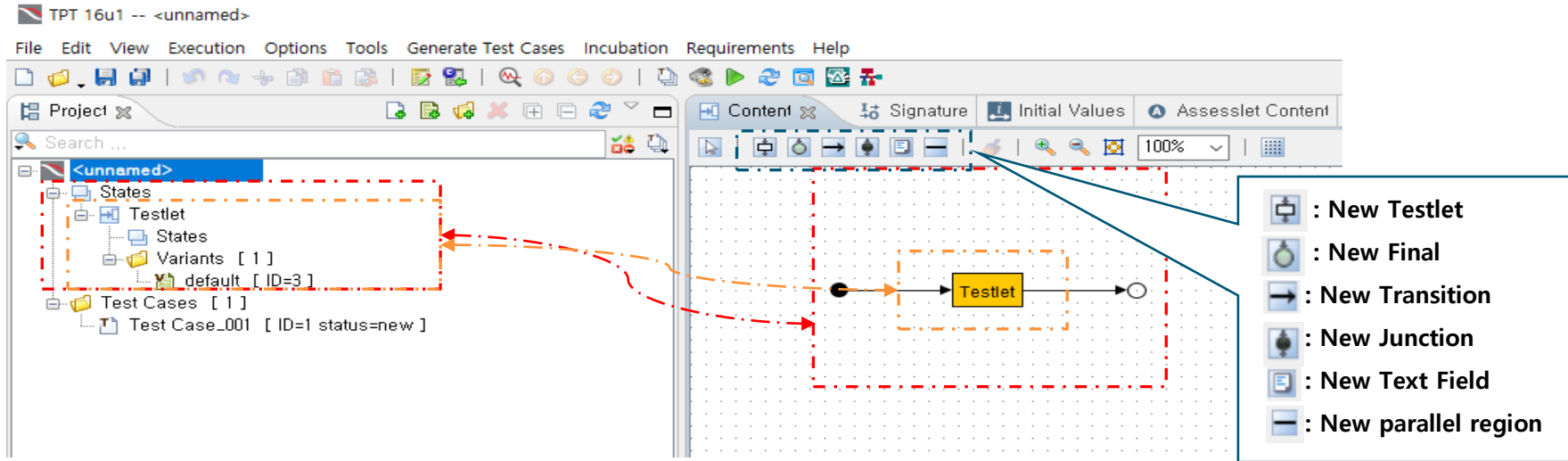
OK Details

Test Driver 생성 중 위와 같은 오류 발생 시, "Compile as 64 bit" 체크 박스를 체크 한 후 다시 "Generate and Compile test driver..." 버튼을 클릭한다.

TPT Test Modeling

TPT Test Modeling

◆ 시나리오 기반 모델링



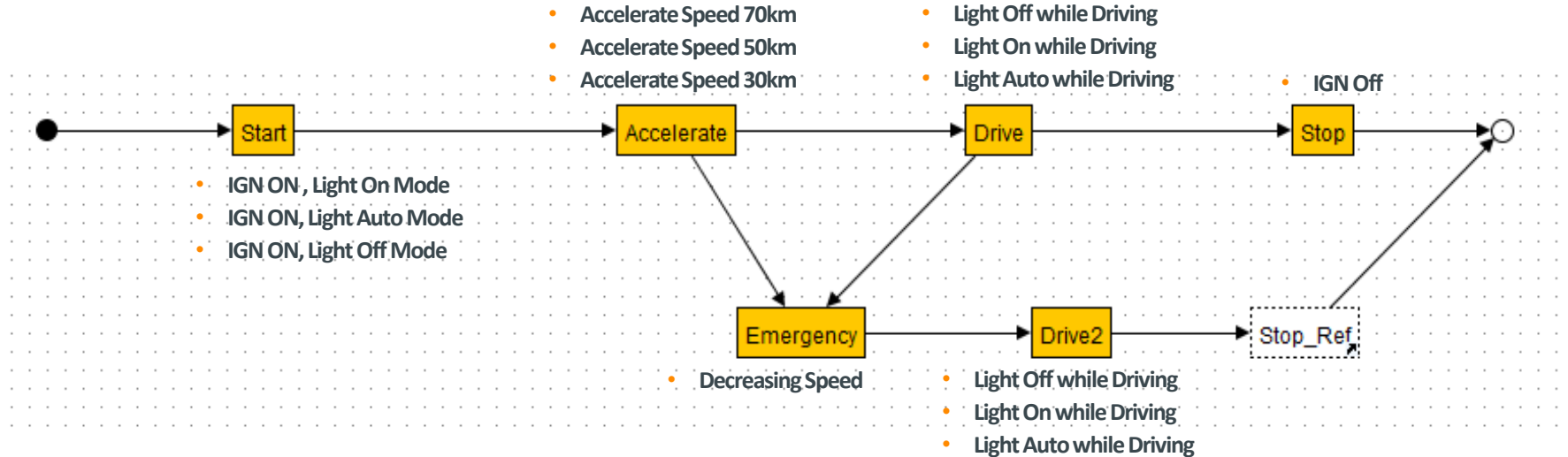
TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

- 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인

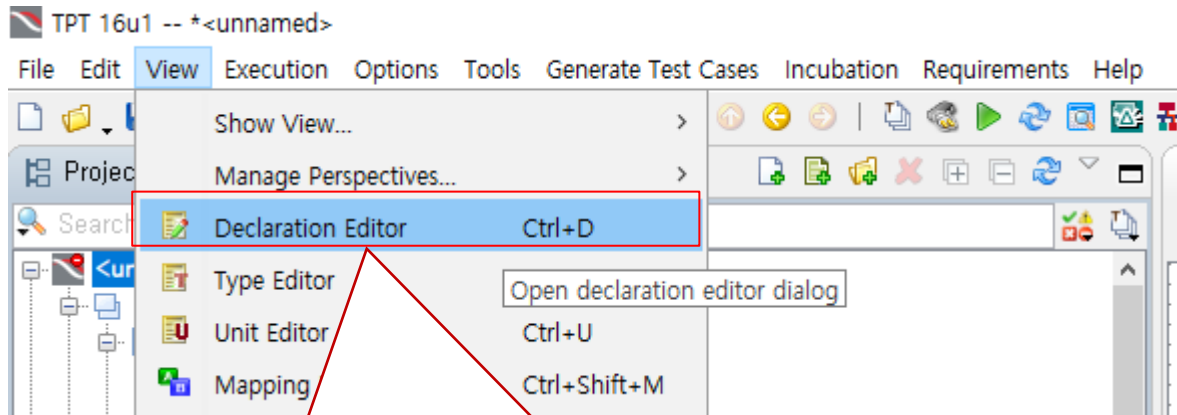


Input		Output
Light switch	Light Intensity	headlight
0(Off)	Don't care	0(Off)
1(On)	Don't care	1(On)
2(Auto)	70 >	0(Off)
2(Auto)	60 <	1(On)



TPT Test Modeling

◆ TPT Declaration Editor - 실행



View → Declaration Editor를 클릭 또는 Ctrl + D 키를 눌러 Declaration Editor를 실행 한다.

TPT Test Modeling

◆ TPT Declaration Editor - Parameter 설정

① Parameter로 변경 할 시그널을 Ctrl을 누른 상태에서 아래 그림과 같이 선택 한다.

Name	Data type	Unit	Value	Equivalence class set	Record	Mode	group	Description
o AUTO_MODE	uint8		2		<input checked="" type="checkbox"/>	OUT		
o headlight	uint8		0		<input checked="" type="checkbox"/>	IN		
o HYSTERESE_TIME_OFF	double		3.0		<input checked="" type="checkbox"/>	OUT		
o HYSTERESE_TIME_ON	double		2.0		<input checked="" type="checkbox"/>	OUT		
o light_intensity	double		0.0		<input checked="" type="checkbox"/>	OUT		
o light_switch	uint8		0		<input checked="" type="checkbox"/>	OUT		
o MIN_LIGHT_OFF	double		70.0		<input checked="" type="checkbox"/>	OUT		
o MIN_LIGHT_ON	double		60.0		<input checked="" type="checkbox"/>	OUT		
o ON_MODE	uint8		1		<input checked="" type="checkbox"/>	OUT		

② Details 탭에서 Type 변경을 위해 Channel에서 Parameter를 선택 한다.

Channel
 Channel
 Measurement
 Constant
 Assessment Variable
Parameter
 Value: <various>

③ Type 변경에 대한 진행여부를 묻는 창에서 "Yes"버튼을 클릭하여 진행 한다.

Change variables type?

Changing the type of a variable removes it from the model and creates a new variable with the same name and properties (value, description and so on). Therefore, occurrences of this variable specific to its original type may be deleted!

Do you really want to change the type of the selected variables?

Don't show this message again

Yes No

④ Type 변경이 완료 되었음을 확인 할 수 있다.

Name	Data type	Unit	Value	Equivalence class set
o headlight	uint8		0	
o light_intensity	double		0.0	
o light_switch	uint8		0	
P o AUTO_MODE	uint8		2	EXCHANGE
P o HYSTERESE_TIME_OFF	double		3.0	EXCHANGE
P o HYSTERESE_TIME_ON	double		2.0	EXCHANGE
P o MIN_LIGHT_OFF	double		70.0	EXCH
P o MIN_LIGHT_ON	double		60.0	EXCH
P o ON_MODE	uint8		1	EXCH

TPT Test Modeling

◆ TPT Declaration Editor – Local 변수 생성

Declaration Editor -- <unnamed>

File View Tools

Mapping: no mapping selected

Name	Data type	Unit	Value	Equivalence class set	Record	Mode	Group	Description
○ AUTO_MODE	uint8		2			EXCHANGE		
○ headlight	uint8		0		<input checked="" type="checkbox"/>	IN		
○ HYSTERESE_TIME_OFF	double		3.0			EXCHANGE		
○ HYSTERESE_TIME_ON	double		2.0			EXCHANGE		
○ light_intensity	double		0.0		<input checked="" type="checkbox"/>	OUT		
○ light_switch	uint8		0		<input checked="" type="checkbox"/>	OUT		
○ MIN_LIGHT_OFF	double		70.0			EXCHANGE		
○ MIN_LIGHT_ON	double		60.0			EXCHANGE		
○ ON_MODE	uint8		1			EXCHANGE		
○ Accelerator_Broken	boolean		false		<input checked="" type="checkbox"/>	LOCAL		

Declaration Editor -- <unnamed>

File View Tools

Mapping: no mapping selected

Hide details...

General Equivalence Class Set

Channel Data type: boolean Description:

Name: Accelerator_Broken

Group: Mode: LOCAL

Unit: Record

Value: false

Type 지정 버튼

Create new channel GUI 버튼 클릭 후 하단의 details 탭에 아래 표의 내용 대로 Name, Data Type, Mode를 지정한다.

Name	Data type	Value	Mode
Accelerator_Broken	Boolean	FALSE	Local
IGN_State	Boolean	FALSE	Local
Speed	double	0	Local

TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

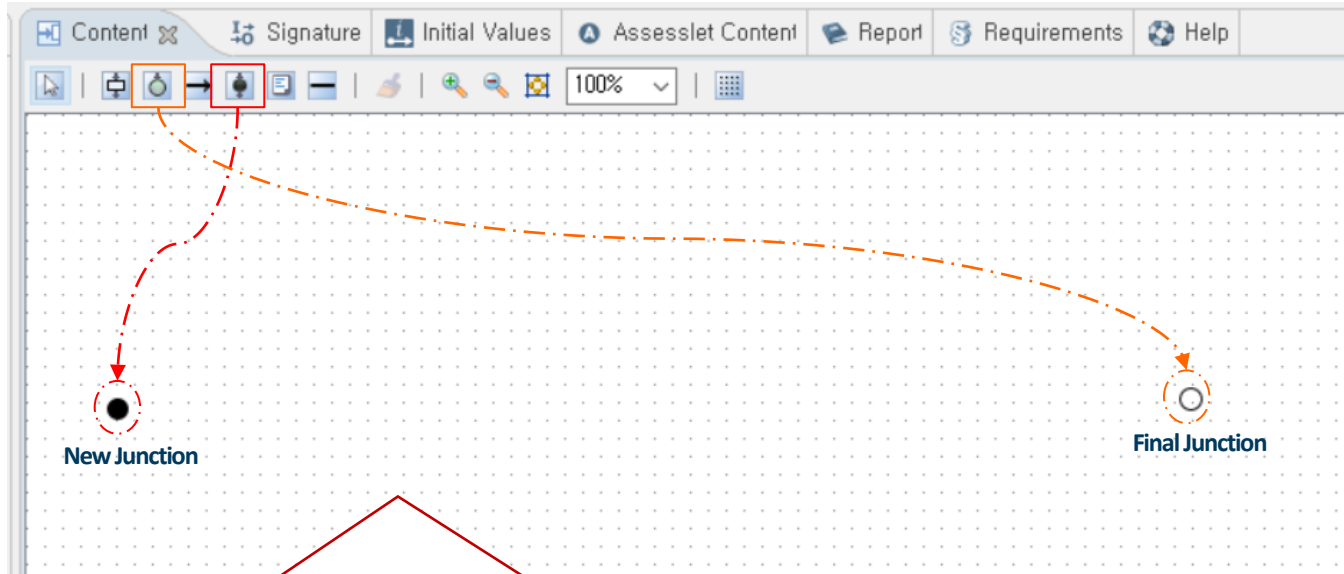
■ 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인

States의 Content 창에서 Ctrl + A 또는 마우스 드래그를 통해 Junction과 Transition, Testlet 모든 영역을 선택 한 후 Testlet 부분에 마우스 오른 클릭 하여 Delete 선택 또는 키보드의 Delete키를 눌러 선택한 Testlet 영역을 전부 삭제 한다.

TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

- 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인 - Testlet 구성

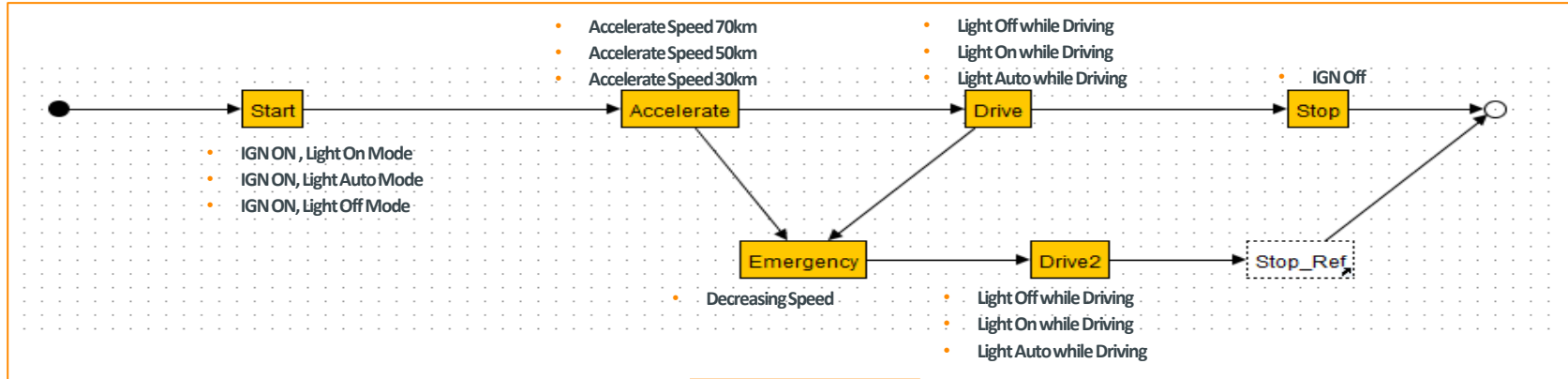


Content 창의 GUI Tool Bar에서 New Junction 아이콘을 클릭 후 White Board의 특정 위치를 클릭하여 New Junction을 생성 한다. 이와 같은 방법으로 Final Junction 아이콘 클릭 후 White Board의 특정 위치를 클릭하여 시나리오 구성의 처음 지점과 끝 지점을 생성 한다.

TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

■ 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인 - Testlet 구성



Content x Signature Initial Values Assesslet Content Report Requirements Help

Content 창의 GUI Tool Bar에서 New Testlet 아이콘을 클릭 후 White Board의 특정 위치를 클릭하여 Testlet을 생성 한다.

Testlet4 Testlet5 Testlet6 Testlet7 Testlet8 Testlet9 Testlet10

TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

■ 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인 - Testlet 구성

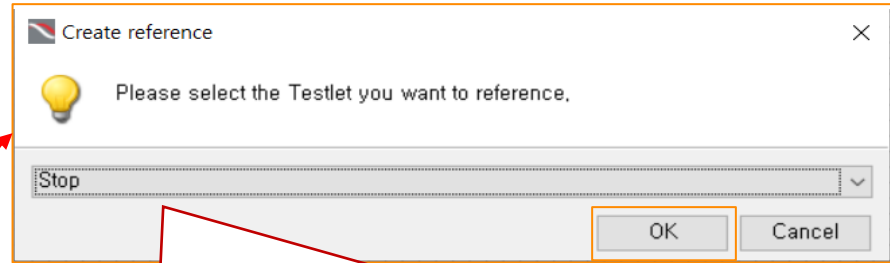
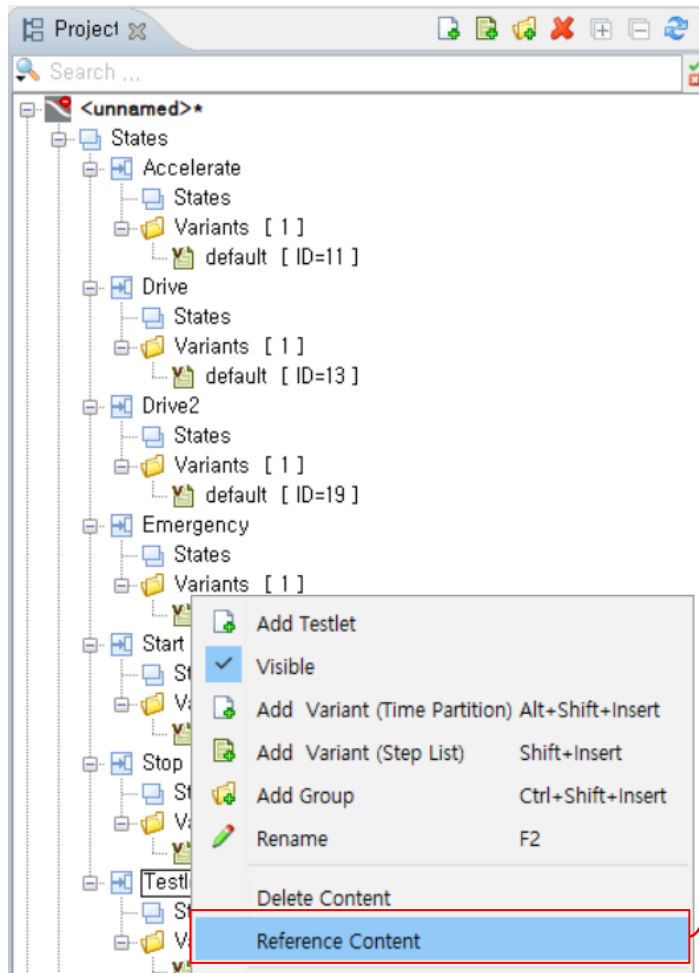
TPT의 탐색창에서 바꾸고자 하는 Testlet을 마우스 오른 클릭, Rename을 선택하여 SW 구동 시나리오에 해당되는 각 상태명을 지정한다.

※ Content White Board에 직접 각 Testlet을 마우스 두 번 클릭하여 변경 할 수 있다.

TPT Test Modeling

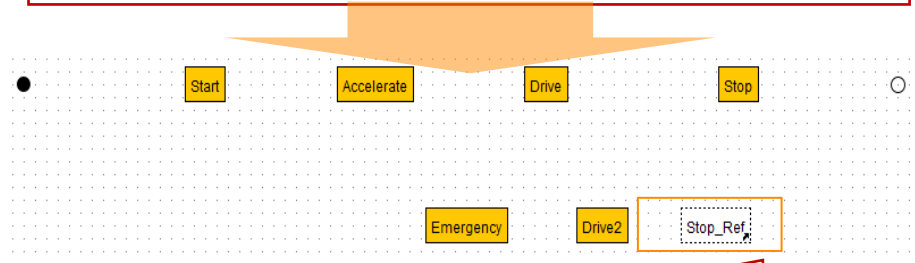
◆ 시나리오 기반 모델링 - 실습

- 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인 – Reference Testlet 구성



프로젝트 탐색 창에서 Testlet10에 마우스 오른 클릭하여 나온 킷 메뉴 중 Reference Content 클릭하여 Create reference 창의 Dropdown 버튼을 클릭하여 동일 속성을 가진 "Stop" Testlet을 선택 한 후 OK 버튼을 클릭 한다.

※ Testlet10일 경우는 Stop Testlet과 동일한 Testlet 속성을 가지고 있으므로, 새로운 Testlet을 생성하지 않고, Reference Testlet으로 구성 할 수 있다.

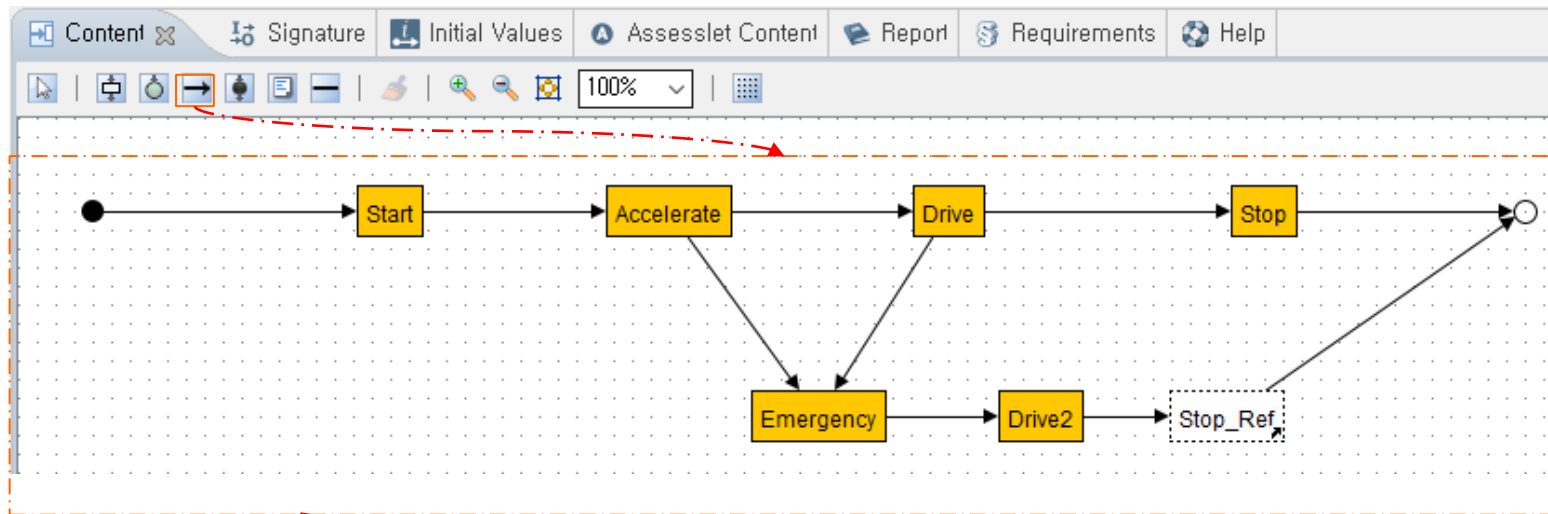



Reference Testlet 생성 후 "Stop_Ref"로 이름을 변경한다.

TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

- 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인 - Transition 연결



Start Junction부터 Final Junction 까지 각 Transition에 맞게  GUI를 클릭 한 후 각 Testlet별 위 그림처럼 연결한다.

TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

- 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인 - Variant 생성

프로젝트 탐색 창에서 각 Testlet별로 Variant 생성을 위해 Testlet안의 Variants 마우스 오른쪽 클릭하여 나온 킷 메뉴의 "Add Variant" 항목을 클릭하여 Testlet 별 Variant를 생성 한다. 생성된 Variant 및 기존의 Default Variant의 킷 메뉴에서 Rename을 클릭, F2키를 눌러 각 해당하는 Variant명을 오른쪽 이미지와 같이 변경 한다.

Reference Testlet은 Variant가 존재하지 않는다.

TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

■ 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인 - Variant별 Step List 생성 방법

The screenshot shows the TPT software interface with a project tree on the left and a content editor on the right. A red dashed line highlights the path from the 'Ignition_On_Mode' testlet in the project tree to the 'Channel' configuration window. The 'Channel' window shows a list of channels including 'Accelerator_Broken', 'IGN_State', 'light_intensity', and 'light_switch'. The 'IGN_State' channel is selected, and its value is set to 'true'.

① Project 탐색 창의 특정 Testlet의 특정 Variant를 클릭 한다.

② GUI 클릭 후 나온 Step List 중 사용하고자 하는 Step List를 클릭 한다.
일반적으로는 특정 입력 시그널에 값을 할당 하기 위해서는 Channel Step List를 사용 한다.

③ 생성된 Channel의 이름을 지정하기 위해 namespace에 마우스 커서 활성화 시킨 후 CTRL + Space 버튼 키를 누르면 Channel 명으로 입력 가능한 시그널 리스트들이 팝업 된다. 팝업 된 리스트 중 사용 할 Channel을 선택 한다.

④ 생성된 Channel의 값을 입력 한다.

TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

- 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인 – Start Testlet의 Step List Variant 생성

Step List Variants

- Ignition_On_On_Mode
- Ignition_On_Auto_Mode
- Ignition_On_Off_Mode

Step	Channel	Parameter	Value	Condition
1	Channel	IGN_State	:= true	Always
2	Channel	Speed	:= 0	Always
3	Wait		5	
4	Channel	light_switch	:= ON_MODE	Always
5	Channel	light_Intensity	:= 0	Always
6	Wait		5	

수정

수정

Start Testlet의 Variant를 위와 같이 생성 한다. 각 Variant의 차이는 light_switch의 값의 차이만 나므로, 하나의 Variant만 생성 한 후 나머지 Variant에 복사 한 후 light_switch의 값만 각 Variant에 맞게 수정 하면 된다.

TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

- 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인 -Accelerator Testlet의 Step List Variant 생성

Step List Variants

Accelerate_
Speed=70km

Accelerate_
Speed=50km

Accelerate_
Speed=30km

수정

수정

추가

Compare Step List는 기입된 시그널의 판정을 한다. 기입된 내용처럼 headlight 시그널이 1과 동일 할 경우 테스트 결과는 "PASS" 그렇지 아닐 경우 "FAIL"이 판정되어 테스트 리포트로 출력된다.

Step	Channel	Value	Operator	Unit	Options	From Scaling	Always
1	Channel	Speed	:=	70			Always
2	Channel	Accelerator_Broken	:=	false			Always
3	Wait	@					
4	Compare	headlight	==	1	with absolute tolerance +/- 0	<input type="checkbox"/>	Always
5	Wait	@					

Step	Channel	Value	Operator	Unit	Options	From Scaling	Always
1	Channel	Speed	:=	50			Always
2	Channel	Accelerator_Broken	:=	false			Always
3	Wait	@					
4	Compare	headlight	==	1	with absolute tolerance +/- 0	<input type="checkbox"/>	Always
5	Channel	Accelerator_Broken	:=	true			Always
6	Wait	@					

Step	Channel	Value	Operator	Unit	Options	From Scaling	Always
1	Channel	Speed	:=	30			Always
2	Channel	Accelerator_Broken	:=	false			Always
3	Wait	@					
4	Compare	headlight	==	1	with absolute tolerance +/- 0	<input type="checkbox"/>	Always
5	Channel	Accelerator_Broken	:=	true			Always
6	Wait	@					

TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

- 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인 – Drive Testlet의 Step List Variant 생성

Step List Variants



Driving_light_off



Driving_light_on



Driving_light_auto

Using namespaces:

Step	Type	Parameter	Value	Condition	Frequency
1	Channel	Speed	:= 100		Always
2	Wait	@			
3	Channel	light_Intensity	:= 90		Always
4	Wait	@			
5	Channel	light_switch	:= 0		Always
6	Wait	1			
7	Compare	headlight	== 0	with absolute tolerance +/- 0	Always
8	Wait	@			

수정

5	Channel	light_switch	:= ON_MODE		Always
---	---------	--------------	------------	--	--------

수정

5	Channel	light_switch	:= AUTO_MODE		Always
7	Compare	headlight	== 0	with absolute tolerance +/- 0	Always
8	Channel	Accelerator_Broken	:= true		Always

추가

TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

- 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인 – Emergency Testlet의 Step List Variant 생성

Step List
Variants



Decreasing
_speed

Step	Icon	Variable	Value	Operator	Options
1	If value	Accelerator_Broken	==	true	with tolerance +/- 0, Once
2	Wait	@			
3	While ...	Speed != 0			
4	Channel	Speed	:=	Speed - 1	Always
5	Wait	@			
6	End (While)				
7	Wait	@			
8	Channel	light_switch	:=	0	Always
9	Wait	@			
10	Compare	headlight	==	0	with absolute tolerance +/- 0, Always
11	Channel	Accelerator_Broken	:=	false	Always
12	Wait	@			
13	Else				
14	Wait	@			
15	End (If)				

Accelerator가 고장이 났을 경우에만 Speed가 10단위로 감소되고, headlight가 off되는 구문을 IF문과 WHILE문을 통하여 만들 수 있다.

※ 주의할 점은 IF문과 While 문 안에는 Wait를 두어 각 Step List가 처리되는 시간을 두어야 하며 End를 IF와 While에 맞게 생성해 주어야 한다.

TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

- 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인 – Drive2 Testlet의 Step List Variant 생성

Step List Variants

- Driving_light_off
- Driving_light_on
- Driving_light_auto

The screenshot shows the Drive2 Testlet Step List Variants editor. The top variant is selected, and the others are shown below with red boxes and arrows indicating modifications.

Step	Type	Channel	Value	Condition
1	Channel	Speed	:= 100	Always
2	Wait	@		
3	Channel	light_Intensity	:= 90	Always
4	Wait	@		
5	Channel	light_switch	:= 0	Always
6	Wait	1		

수정

5	Channel	light_switch	:= ON_MODE	Always
---	---------	--------------	------------	--------

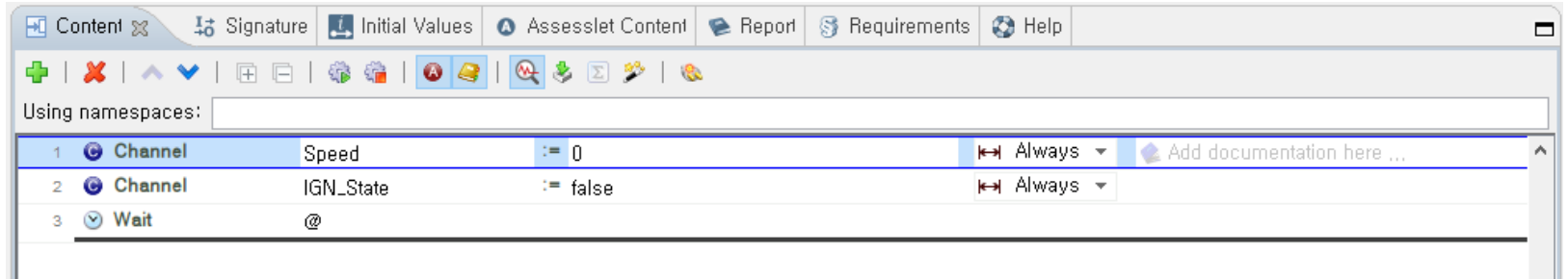
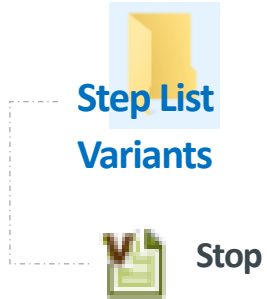
수정

5	Channel	light_switch	:= AUTO_MODE	Always
---	---------	--------------	--------------	--------

TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

- 차량 주행 중 Light Switch, Light Intensity에 따른 Headlight 동작 확인 – Stop Testlet의 Step List Variant 생성



TPT Test Modeling

◆ 시나리오 기반 모델링 - 실습

■ Variant의 기반의 Test Case 자동 생성

① Project 탐색 창에서 Test Cases 폴더를 선택 한다.

② Generate Test Cases → “... from Variants”를 클릭 한다.

③ **Generate** 체크 박스 선택, Generation Mode는 Combination으로 설정 한 후 OK 버튼을 클릭한다.

TPT Test Modeling

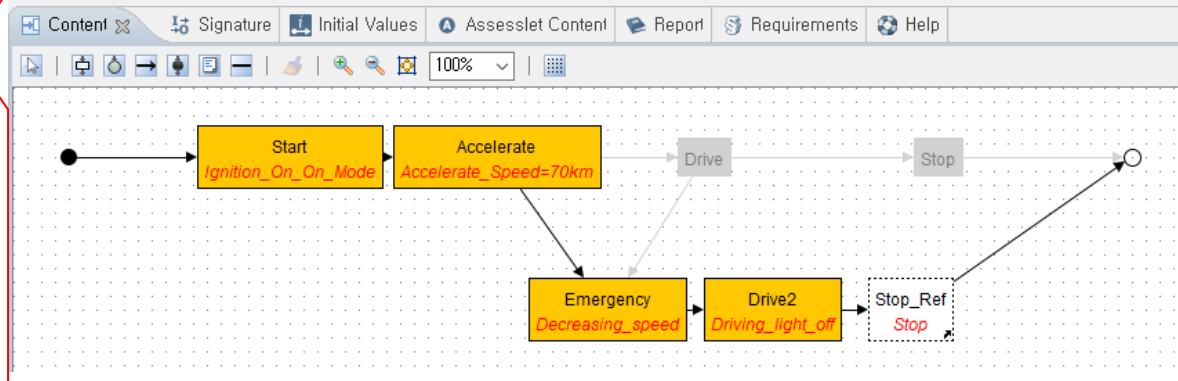
◆ 시나리오 기반 모델링 - 실습

■ Variant의 기반의 Test Case 자동 생성

Test Cases [135]

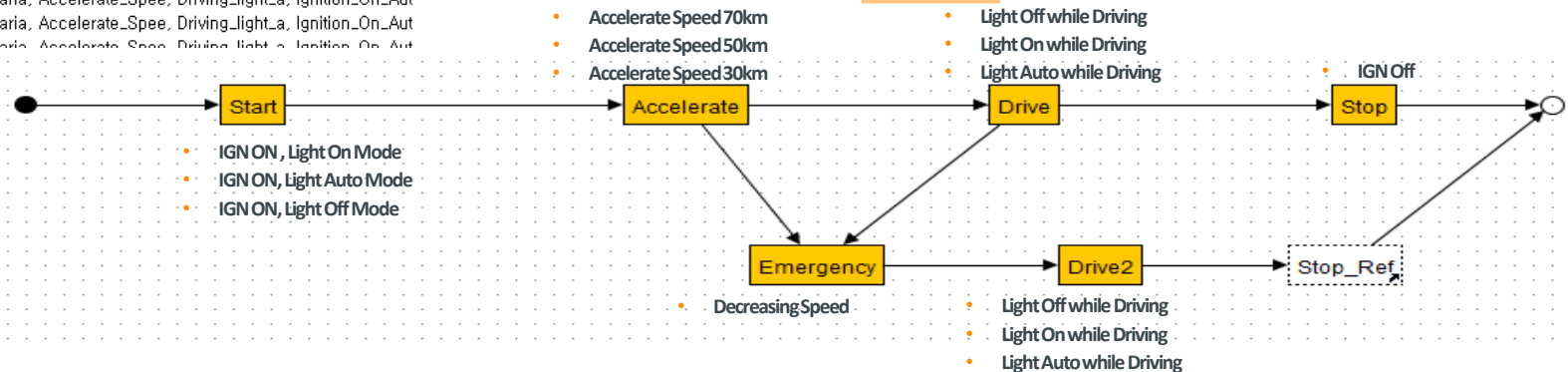
- TC1: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_
- TC2: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_
- TC3: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_
- TC4: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_
- TC5: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_
- TC6: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_
- TC7: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_On_
- TC8: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_On_
- TC9: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_On_
- TC10: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Off
- TC11: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Off
- TC12: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Off
- TC13: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Off
- TC14: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Off
- TC15: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Off
- TC16: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_Off
- TC17: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_Off
- TC18: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_Off
- TC19: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Aut
- TC20: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Aut
- TC21: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Aut
- TC22: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Aut
- TC23: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Aut
- TC24: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Aut
- TC25: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_Aut
- TC26: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_Aut
- TC27: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Aut
- TC28: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Aut

생성된 Test Case중 임의로 하나 선택 시 해당되는 시나리오를 Content로 확인해 볼 수 있다.
- 각 Testlet별 Variant 정보 및 Transition 정보



각 상태 천이 Path 별 모든 경우의 Test Case 자동 생성

- Start -> Accelerate -> Drive -> Stop(27)
- Start -> Accelerate -> Emergency -> Drive2 -> Stop_Ref(27)
- Start -> Accelerate -> Drive -> Emergency -> Drive2 -> Stop_Ref(81) = 135 Test Case 생성




TPT Test Execution

TPT Test Execution

◆ Test Execution

The screenshot displays the TPT software interface. The main window shows a project tree on the left with 'Test Cases [135]' highlighted. The 'Execution Configuration' dialog is open, showing settings for 'Data storage', 'Report', and 'Execution'. The 'Test set' dropdown is set to '-- Selected test cases -- (135)'. The 'Run' button is highlighted in the bottom right corner of the dialog.

① Execution → Execution Configuration 또는  GUI 버튼을 클릭하여 Execution Configuration을 실행 시킨다.

※ Test 실행 할 테스트 케이스 선택은 Project 탐색 창의 Test Cases 항목에서 선택 가능하다. 모든 테스트 케이스를 선택 할 경우 Test Cases Group을 선택하면 되고 특정 테스트 케이스만 실행 할 경우 하위 특정 테스트 케이스를 선택한 후 Execution Configuration을 실행 하면 된다.

② Run 버튼을 클릭하여 Test를 실행 한다.

TPT Test Execution

◆ Test Execution

The screenshot displays the TPT Build Progress window with the following components:

- Test Case Table:** A table listing test cases from TC1 to TC15. TC12 is currently selected and highlighted in blue. The 'Messages' column shows 'Execution finished.' for TC12, while others show 'Done.' with green checkmarks or red X marks.
- Details Panel:** Shows the execution steps for the selected test case: Initialization, Execution, Assessment, Report generation, and Summary.
- Log Output:** A text area showing the execution process, including parameter binding, compilation, and the execution of the test executable.
- Progress Bar:** Located at the bottom, it indicates 'Runs: 11 / 135 (02:14 remaining)'.

Test 실행 시 TPT Build Progress 창에서 선택한 테스트 케이스가 순차적으로 실행되는 것을 확인 할 수 있다.

※ Timeout 시간 이내에 각 테스트 케이스의 Step List 실행이 완료 되었을 경우 바로 다음 테스트 케이스가 실행 된다.

TPT Test Execution

◆ Test Report

The screenshot shows the TPT Build Progress window with a list of test cases and an Execution Configuration dialog. The test cases list includes TC118 through TC135, with TC135 highlighted. The Execution Configuration dialog shows the Report Format set to HTML (embedded resources).

모든 테스트 케이스 실행이 완료되면, "Overview Report" GUI 버튼이 활성화된다. 해당 버튼을 클릭하여 되면 Test Execution 창에서 설정한 Report 양식대로 Report 화면이 팝업 된다.

Report 양식은 HTML, PDF 가 지원된다.

Test Case	Messages	From	To	Result
TC118: Region :0 varia, Accelerate...	Done.			✖
TC119: Region :0 varia, Accelerate...	Done.			✖
TC120: Region :0 varia, Accelerate...	Done.			✖
TC121: Region :0 varia, Accelerate...	Done.			✖
TC122: Region :0 varia, Accelerate...	Done.			✖
TC123: Region :0 varia, Accelerate...	Done.			✖
TC124: Region :0 varia, Accelerate...	Done.			✖
TC125: Region :0 varia, Accelerate...	Done.			✖
TC126: Region :0 varia, Accelerate...	Done.			✖
TC127: Region :0 varia, Accelerate...	Done.			✔
TC128: Region :0 varia, Accelerate...	Done.			✔
TC129: Region :0 varia, Accelerate...	Done.			✔
TC130: Region :0 varia, Accelerate...	Done.			✖
TC131: Region :0 varia, Accelerate...	Done.			✖
TC132: Region :0 varia, Accelerate...	Done.			✖
TC133: Region :0 varia, Accelerate...	Done.			✖
TC134: Region :0 varia, Accelerate...	Done.			✖
TC135: Region :0 varia, Accelerate...	Done.			✖

Starting test executable "C:\Users\JungSub.Kuk\Desktop\SUT-C-Code Simple\output\bin\TPT_TestDriver.exe" C:\Users\JungSub.Kuk\...
saving tracks done
Done.
Execution finished.

[Compare No. 7: headlight == 0]
expected headlight == 0 always in interval {11.04s <= t < 11.05s}
failure at t=11.04s: expected 0, got 1

[Summary]
Done.

Runs: 135 / 135 (total t...

TPT Test Execution

Test Report

TPT Report: Overview

Test Summary

- 54 Passed (40%)
- 81 Failed (60%)
- 0 Inconclusive
- 0 Execution Errors
- 135 Total

Platform	Passed	Failed	Inconclusive	Execution Error	Total
C/C++ Platform	54	81	0	0	135

1 C/C++ Platform

Test Case Name
TC1: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_
TC2: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_
TC3: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_
TC4: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_
TC5: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_
TC6: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_
TC7: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_On_
TC8: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_On_
TC9: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_On_
TC10: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Off
TC11: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Off
TC12: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Off
TC13: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Off
TC14: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Off
TC15: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_Off
TC16: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_Off
TC17: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_Off
TC18: Region :0 varia, Accelerate_Spee, Driving_light_a, Ignition_On_Off

TPT Test Case Report: TC1: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_

1 Test Information

1.1 Meta Information

Test Case Name	TC1: Region :0 varia, Accelerate_Spee, Driving_light_o, Ignition_On_On_
Test Case ID	1004
Test Result	Passed
Execution started	08:35:29 16.03.2021
Assessment started	08:35:30 16.03.2021
Duration	293ms (execution: 104ms, assessment: 189ms)
Test Case Group	
TPT File	C_Platform_for_Mando.tptz
Directory	!C_C_Platform000_TC1_Region_0_varia_Accela
Test Data	Open Test Data
Execution Config	Default Configuration
Platform Config	C/C++ Platform
Platform Mapping	<none>

1.2 Asseslet Summary

Asseslet Name	!Accelerate_Speed=70km (in state 'Accelerate')
---------------	--

2 Signals

2.1 System Constants

2.2 Assessments

2.3 Inputs

Name	Initial Value	Value	From	To
headlight	0		0s	11.06s

2.4 Outputs

Name	Initial Value	Value	From	To
light_intensity	0.0		0s	11.06s
light_switch	0		0s	11.06s

TPT Report Overview 창에서 Test Case 리스트 중에 하나를 선택하여 선택한 테스트 케이스의 자세한 내용을 확인해 볼 수 있다.

- Input, Output, Assessment 등.

Thank you

인텔리전트 융합 솔루션 전문기업
(주)한컴MDS

