|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document Change Histroy | | | | |
| Date  (YYYY-MM-DD) | Ver. | Editor | Chap | 내용(개정 전 -> 개정 후) |
| 2016-07-29 | 1.0 | J Jung |  | * Initial Creation |
| 2016-08-30 | 1.1 | J Jung |  | * 오타 수정 |
| 2017-02-15 | 1.2 | J Jung |  | * SDAT integration code화 업데이트 |

**Purpose (목적):**

This document provides the method of using the SWC Error Management.

**Scope (범위):**

The definitions and documented decisions in this document are relevant for the application development through SWP.

**Remark (비고):**

본 문서는 AUTOSAR 플랫폼 기반으로 AUTOSAR에서 제공하는 각각의 기능에 대해 사용자가 쉽게 사용하도록 하기 위한 문서이다.

본 문서에는 해당 모듈의 ASW 관점의 설정 및 생성 API 사용, 그리고 쉽게 범하기 쉬운 실수에 대한 Guide 내용이 포함되어 있다.

**Table of Contents**

[1 Overview 3](#_Toc474939405)

[1.1 Reference 3](#_Toc474939406)

[1.2 AUTOSAR System 3](#_Toc474939407)

[**1.2.1** Overview of Software Layers 3](#_Toc474939408)

[**1.2.2** Sample SDAT 4](#_Toc474939409)

[**1.2.3** Limitations 4](#_Toc474939410)

[**1.2.4** Deviation 4](#_Toc474939411)

[**1.2.5** Abbreviations 4](#_Toc474939412)

[2 SDAT 설정 6](#_Toc474939413)

[2.1 integration\_ErrM 추가 및 삭제 6](#_Toc474939414)

[**2.1.1** integration\_ErrM 삭제 6](#_Toc474939415)

[**2.1.2** integration\_ErrM 추가 6](#_Toc474939416)

[2.2 Dcm DidRange 설정 7](#_Toc474939417)

[**2.2.1** Did Info설정 7](#_Toc474939418)

[**2.2.2** Did Range 설정 7](#_Toc474939419)

[2.3 SWComponent 설정 8](#_Toc474939420)

[**2.3.1** SWComponent 추가 8](#_Toc474939421)

[**2.3.2** Port Interface 설정 9](#_Toc474939422)

[2.4 Dem Event 설정 주의사항 11](#_Toc474939423)

[3 SDAT 적용 12](#_Toc474939424)

[3.1 ErrM\_Cbk\_User.c 수정 방법 12](#_Toc474939425)

[3.2 SDAT Protocol 12](#_Toc474939426)

[**3.2.1** Sample SDAT Protocol 12](#_Toc474939427)

[**3.2.2** Sample SDAT Protocol CANoe 검증 13](#_Toc474939428)

[3.3 Error Code 15](#_Toc474939429)

[4 FAQ 16](#_Toc474939430)

# Overview

본 문서는 샘플로 제공되는 Error Management 사용을 위해 AUTOSAR 플랫폼 사용할 때, 사용자가 파라미터 설정 또는 시스템 설계를 할 때 주의하거나 참고할 사항을 제공한다. Autosar 표준 SRS/SWS를 기반으로 작성 되었으며, 모듈 사용시 보다 자세한 기능적인 설명이 필요한 경우, 아래 Reference 문서를 참고한다.

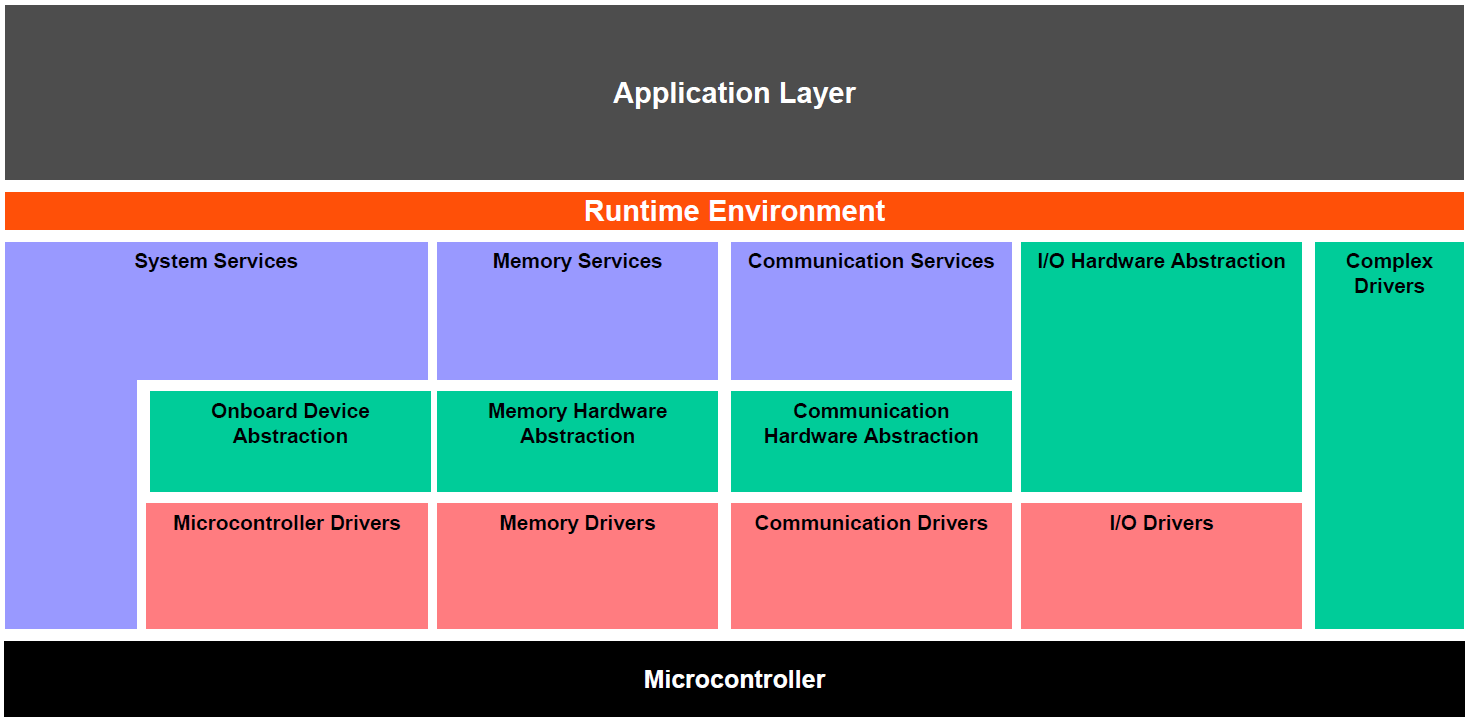
## Reference

| Sl. No. | Title | Version |
| --- | --- | --- |
| 1. | AUTOSAR\_SWS\_DiagnosticCommunicationManager.pdf | 4.0.3 |
| 2. | AUTOSAR\_SWS\_DiagnosticEventManager.pdf | 4.0.3 |
| 3. | AUTRON\_AUTOSAR\_Os\_UM.doc | 1.6 |
| 4. | ES95486-00 | 1.9.0 |

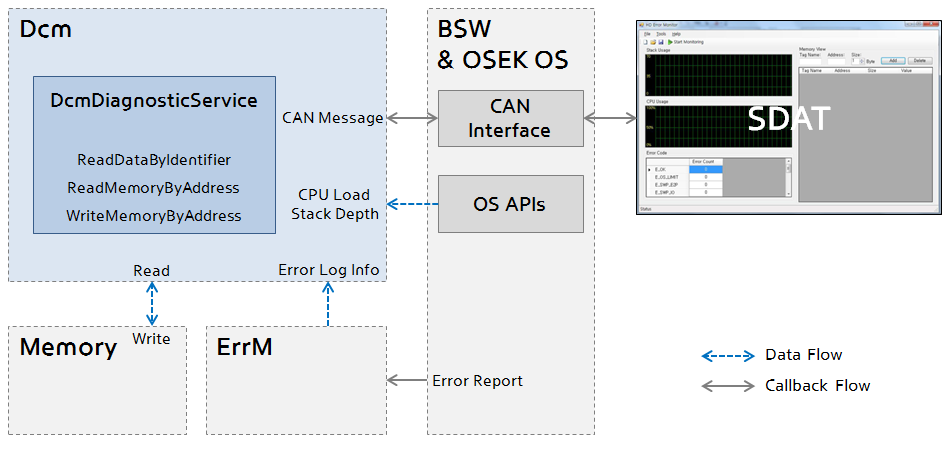
## AUTOSAR System

### Overview of Software Layers

AUTOSAR 플랫폼의 Layered Architecture 는 아래와 같다. AUTOSAR 플랫폼은, Service Layer, ECU Abstraction Layer, Complex Device Drivers및 Microcontroller Abstraction Layer 로 구분될 수 있다.



### Sample SDAT



- SDAT Tool에서 필요한 정보는 Stack Usage, CPU Load, Error Code 등이 있다.

- SDAT Tool에서 Read/Write Memory 기능은 지원하지 않는다.

- Dcm은 UDS Service를 통해 위 정보들을 제공한다.

- ReadDataByIdentifier (DidRange) : Stack Usage, CPU Load, Error Code

- ReadMemoryByAddress : Read Memory

- WrtieMemoryByAddress : Write Memory

### Limitations

- SDAT Tool 에서 Read/Write Memory 기능은 지원하지 않는다.

### Deviation

- 없음

### Abbreviations

|  |  |
| --- | --- |
| Abbreviations | Definition |
| **AR** | acronym for AUTOSAR |
| **ARXML** | File extension for AUTOSAR XML-formatted file(s) |
| **ASW** | Application Software, comprising any number of SWCs and/or Compositions |
| **AUTOSAR** | AUTomotive Open System ARchitecture - a standardized software architecture targeted at automotive applications aimed at fostering the reuse of application software over multiple vehicle platforms. |
| **BSW** | Basic Software |
| **BswM** | Basic Software Mode Manager |
| **BSWMD** | Basic Software Module Description |
| **CDD** | Complex Driver |
| **COM** | Communication |
| **ComM** | Communication Manager |
| **ECU** | Electronic Control Unit |
| **EcuM** | ECU State Manager module |
| **HW** | Hardware |
| **IOC** | Inter OS-Application Communicator |
| **ISR** | Interrupt Service Routine |
| **MCAL** | Micro-Controller Abstraction Layer |
| **MCU** | Micro Controller Unit |
| **OS** | Operating System |
| **OSEK/VDX** | Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug |
| **RTE** | Run-Time Environment |
| **RTOS** | Real Time Operating System |
| **SAG** | Software platform Application Guide |
| **SDAT** | Software Dynamic Analysis Tool |
| **SRS** | Software platform Requirement Specification |
| **SW** | Software |
| **SWC** | an atomic Software Component (or component, for short), the ‘smallest’ application software entity in AUTOSAR |

# SDAT 설정

## integration\_ErrM 추가 및 삭제

### integration\_ErrM 삭제

integration\_ErrM 의 삭제는 다음에 소개되는 추가 가이드를 참고하여 진행한다. 소스코드와 설정파일을 삭제하고, generate.py 를 삭제한다. SDAT 기능 지원을 위하여 Dcm과 Port Interface 설정해준 부분을 삭제한다.

### integration\_ErrM 추가

① integration\_ErrM 파일 추가

SDAT 기능을 적용하고자 하는 프로젝트에 integration\_ErrM을 추가한다. 이때 기존에 ErrM 관련 코드가 남아있다면 삭제한다.

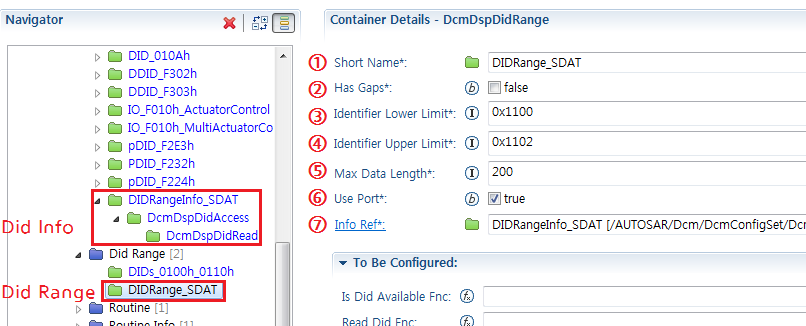
② generate.py 수정

GenerateRte 코드의 source에 App\_ErrM을 추가한다 (아래 그림 참고). App\_ErrM에는 SDAT를 위한 Callback의 P-Port가 기설정되어있다.



## Dcm DidRange 설정

Stack Usage, CPU Load, Error Code Read를 위해 진단서비스 ReadDataByIdentifier (SID22) 를 활용한다.



### Did Info설정

Stack Usage, CPU Load, Error Code 등에 대한 정보를 Read만 하기 때문에 DcmDspDidInfo 컨테이너 하위에 위 그림과 같이 DcmDspDidRead 만 생성한다. (DcmDspDidInfo의 DcmDspDidDynamicallyDefined는 false로 설정)

### Did Range 설정

DcmDspDidRange 컨테이너를 생성하고 위 그림을 참고하여 아래와 같이 설정한다.

① Short Name : "DIDRange\_SDAT" 라고 설정 (변경 불가)

② Has Gaps : false 로 설정 (변경 불가)

③ Identifier Lower Limit : SDAT 용 Did Range Callback을 호출하기 위한 Identifier의 하한. SDAT 툴과 설정 값이 동일해야하며, 유일한 Did 값이어야 한다.

④ Identifier Upper Limit : SDAT 용 Did Range Callback을 호출하기 위한 Identifier의 상한. SDAT 툴과 설정 값이 동일해야하며, 유일한 Did 값이어야 한다.

⑤ Max Data Length : Did Range Callback을 통해서 보낼 Data Length (Byte). Timestamp, Stack Usage, CPU Load의 유무와 Error Code의 개수에 따라 Response Length가 달라지는 것을 고려한다. Dcm Buffer 사이즈 역시 고려한다.

**Note** Timestamp (4-Byte) | Stack Usage (4-Byte) | CPU Load (2-Byte) | Error 개수 \* 3-Byte <= Max Data Length

**Note** Max Data Length + 3 <= Dcm Tx Buffer Size

⑥ Use Port : true 로 설정 (변경 불가)

⑦ Info Ref : 2.2.1 에서 생성한 Did Info 를 레퍼런스

Did Info, Did Range 설정을 완료한 후 Generate Swc 한다 (R-Port 생성).

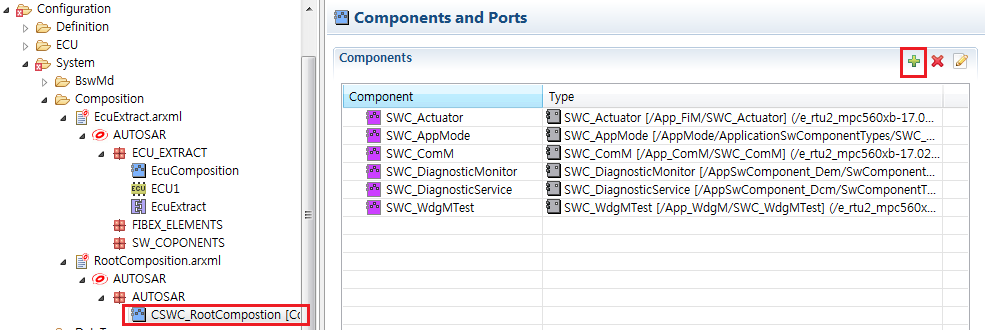
## SWComponent 설정

**Note** SWC 설정에 앞서 Dcm, Dem Swcd Generator가 수행되어야 한다.

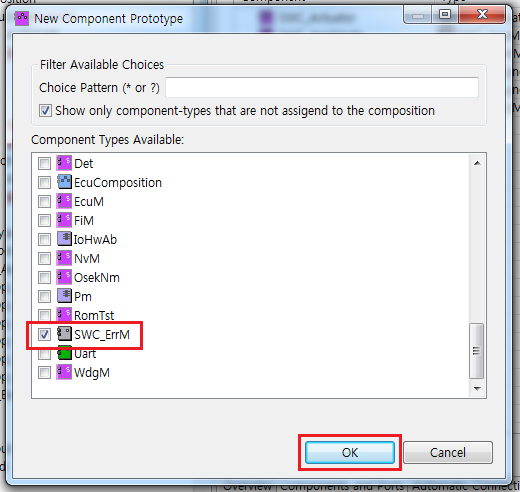
### SWComponent 추가

ErrM 의 SWComponent를 플랫폼에 추가하기 위하여 다음 순서로 작업한다.

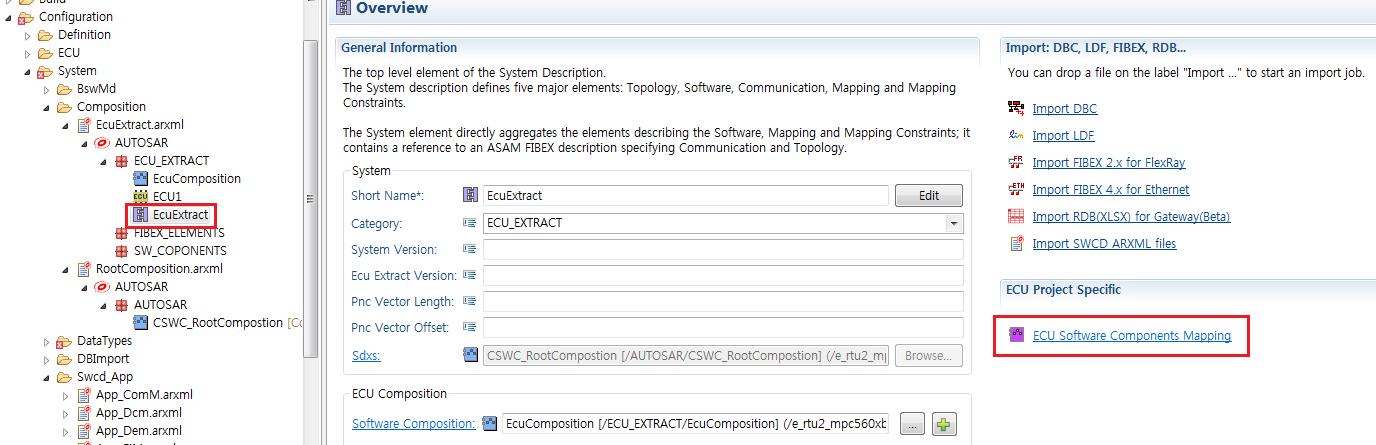
① RootComposition.arxml/CSWC\_RootComposition의 Components and Ports 탭을 연다. New (초록색 더하기) 를 클릭한다 (아래 그림 참고).



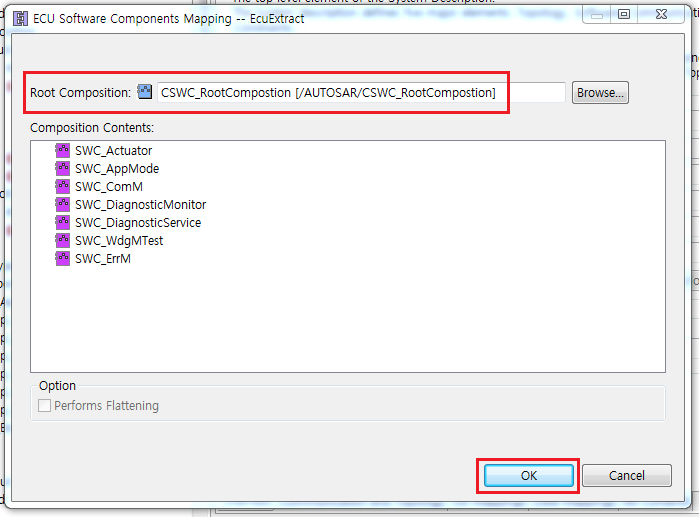
② SWC\_ErrM을 추가한 후 OK를 클릭한다 (아래 그림 참고).



③ EcuExtract.arxml/EcuExtract 의 ECU Software Components Mapping 을 클릭한다 (아래 그림 참고).

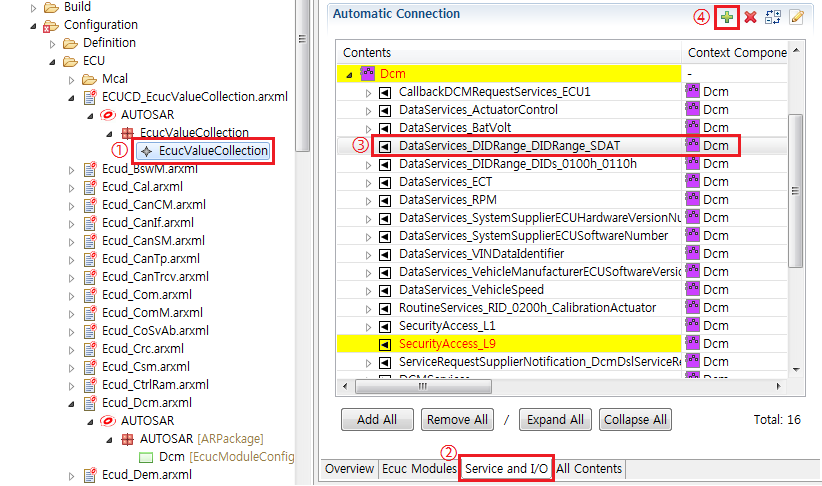


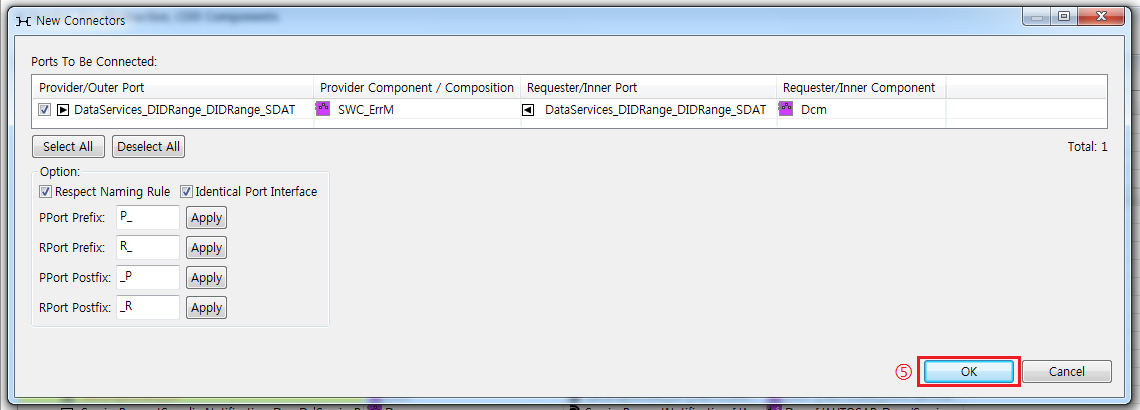
④ OK를 클릭하여 RootComposition 의 변경사항을 적용한다 (아래 그림 참고).



### Port Interface 설정

App\_ErrM.arxml 에 기설정되어 제공한 P-Port (Chapter 2.1)와 Dcm 설정을 통하여 생성한 R-Port (Chapter 2.2) 를 연결하는 작업이다.





위 그림을 참고하여 아래와 같이 작업한다.

① EcucValueCollection 클릭

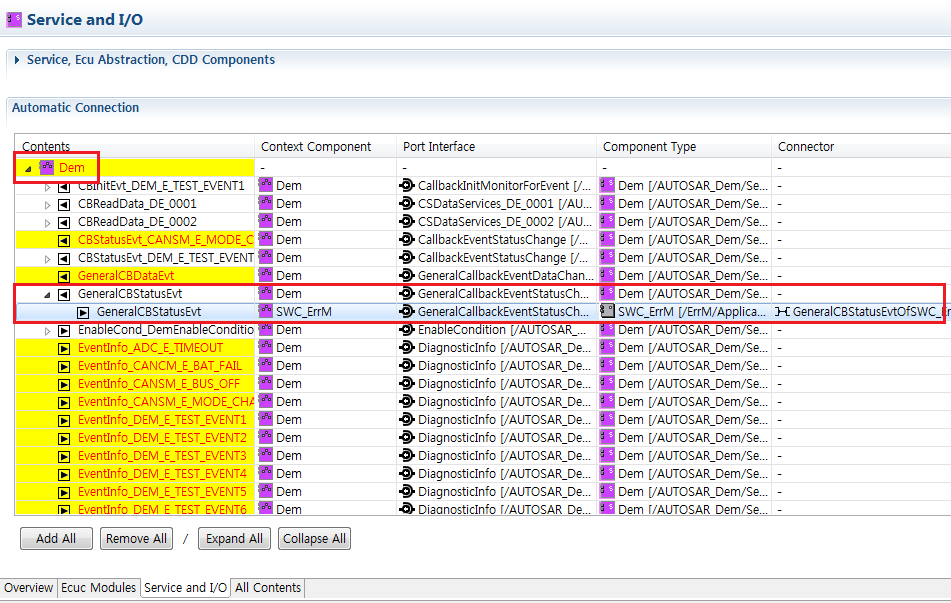
② Service and I/O Tab 클릭

③ DataServices\_DIDRange\_DIDRange\_SDAT 클릭

④ New Connectors (초록색 더하기 아이콘) 클릭

⑤ 이미 설정되어 있으므로 OK 클릭

마찬가지로 Dem에서도 GeneralCBStatusEvt를 SWC\_ErrM의 포트와 연결한다 (아래 그림 참고).



## Dem Event 설정 주의사항

SDAT를 지원하기 위해 Dem 설정은 아래 주의사항을 만족하여야 한다.

**Note** SDAT에 사용할 모든 Dem Bsw Event Id는 1번부터 순차적으로 설정되어야 한다.

**Note** 각 Bsw Event는 각 모듈에 참조되어야 한다. 각 모듈에 Event 설정 시 플랫폼 배포 이후 사용자 설정이 불가능한 항목이 존재하므로 [현대차표준플랫폼\_시스템에러코드.xlsx]에 있는 각 이벤트에 대해 모듈 담당자에게 문의 후 설정할 필요가 있다.

**Note** MCAL 용 Dem Event를 위한 dummy arxml (Ecud\_Dem\_Mcal.arxml) 가 있을 경우 MCAL 용 Event와 Dem Event의 EventName, EventId는 각각 동일해야 한다.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ex) Ecud\_Dem\_Mcal.arxml에 아래와 같은 이벤트가 설정되어 있다면,   |  |  |  | | --- | --- | --- | | **Name** | **EventId** | **EventKind** | | A | 1 | BSW | | B | 2 | BSW |   Ecud\_Dem.arxml은 아래와 같이 설정되어야 한다.   |  |  |  | | --- | --- | --- | | **Name** | **EventId** | **EventKind** | | A | 1 | BSW | | B | 2 | BSW | | C | 3 | BSW | | D | 4 | BSW |   \* C, D 는 MCAL이 아닌 Bsw 모듈 이벤트이다. |

# SDAT 적용

## ErrM\_Cbk\_User.c 수정 방법

ErrM\_Cbk\_User.c 파일은 Sample 로 제공되는 코드이므로 단순 적용을 금지한다. 아래 주의 사항을 참고하여 사용자의 프로젝트 환경에 맞도록 변경이 필요하다.

**Note** Stack Usage, CPU Load, Timestamp 와 관련한 API는 OS 유저매뉴얼 Appendix 참고

**Note** User Config Macros 를 다음과 같이 정의

|  |  |
| --- | --- |
| **Macro** | **Description** |
| OS\_MAX\_ERROR\_NUM | Os Error 총 개수를 확인하여 사용자가 수정 |
| DEM\_MAX\_ERROR\_NUM | SDAT에서 사용할 이벤트의 개수를 확인하여 사용자가 수정 |
| DIDRANGE\_MAX\_DATA\_LENGTH | 사용자가 DcmDspDidRange 컨테이너에서 설정한 Max Data Length 값 |

**Note** DcmDspDidRange 컨테이너에서 설정한 Identifier Lower Limit, Identifier Upper Limit 의 DID값을 ErrM\_SDAT\_DidRange() 의 switch문에 적용

## SDAT Protocol

**Note** 다음은 Sample로 제공되는 SDAT의 프로토콜이다. 해당 내용은 사용자의 Dcm 설정, Error 설정, ErrM\_Cbk\_User.c 수정 내용에 따라 변경될 수 있다.

### Sample SDAT Protocol

- Dcm ReadDataByIdentifier(SID22) 를 이용하며 Data를 제외한 Request, Response Protocol은 ES95486 진단 사양을 따른다.

- ReadDataByIdentifier의 Did는 기존에 사용중이지 않은 독립적인 Did를 사용하며, 제공되는 Sample에는 0x1100, 0x1101, 0x1102로 설정되어 있다. 각 Did는 다음과 같은 정보를 전달한다.

|  |  |
| --- | --- |
| **Did** | **Response Datas** |
| 0x1100 | Time Stamp (4byte), MainStack Usage (4byte), CPU Load (4byte) |
| 0x1101 or 0x1102 | Time Stamp (4byte), Os Error Code (3byte) \* 총 개수 + Dem Error Code (3byte) \* 총 개수 (단, 총 DIDRANGE\_MAX\_DATA\_LENGTH개까지만 받고 나머지 버림) |

### Sample SDAT Protocol CANoe 검증

1) Usage Data 요청 (ReadDataByIdentifier with Did 0x1100)



- Request Message : 22 11 00

|  |  |  |
| --- | --- | --- |
| **순서** | **메시지** | **의미** |
| 1 byte | 22 | ReadDataByIdentifier Service ID |
| 2 byte | 11 | Did |
| 3 byte | 00 | Did |

- Response Message : 62 11 00 1A B9 C1 F7 00 00 04 D0 00 9E

|  |  |  |
| --- | --- | --- |
| **순서** | **메시지** | **의미** |
| 1 byte | 62 | ReadDataByIdentifier Service Positive Response ID |
| 2 byte | 11 | Did |
| 3 byte | 00 | Did |
| 4 byte | 1A | Time Stamp |
| 5 byte | B9 | Time Stamp |
| 6 byte | C1 | Time Stamp |
| 7 byte | F7 | Time Stamp |
| 8 byte | 00 | Main Stack Usage |
| 9 byte | 00 | Main Stack Usage |
| 10 byte | 04 | Main Stack Usage |
| 11 byte | D0 | Main Stack Usage |
| 12 byte | 00 | CPU Load |
| 13 byte | 9E | CPU Load |

2) Error Code 요청 (ReadDataByIdentifier with Did 0x1101 or 0x1102)

2-1) Error 가 발생하지 않은 경우



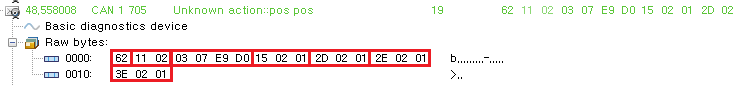
- Request Message : 22 11 02

|  |  |  |
| --- | --- | --- |
| **순서** | **메시지** | **의미** |
| 1 byte | 22 | ReadDataByIdentifier Service ID |
| 2 byte | 11 | Did |
| 3 byte | 02 | Did |

- Response Message : 62 11 02 01 EB A4 37

|  |  |  |
| --- | --- | --- |
| **순서** | **메시지** | **의미** |
| 1 byte | 62 | ReadDataByIdentifier Service Positive Response ID |
| 2 byte | 11 | Did |
| 3 byte | 02 | Did |
| 4 byte | 01 | Time Stamp |
| 5 byte | EB | Time Stamp |
| 6 byte | A4 | Time Stamp |
| 7 byte | 37 | Time Stamp |

2-2) ErrorId 0x15, 0x2D, 0x2E, 0x3E의 4개의 Dem Error 가 각 1회 발생한 경우



- Request Message : 22 11 02

|  |  |  |
| --- | --- | --- |
| **순서** | **메시지** | **의미** |
| 1 byte | 22 | ReadDataByIdentifier Service ID |
| 2 byte | 11 | Did |
| 3 byte | 02 | Did |

- Response Message : 62 11 02 03 07 E9 D0 15 02 01 2D 02 01 2E 02 01 3E 02 01

|  |  |  |
| --- | --- | --- |
| **순서** | **메시지** | **의미** |
| 1 byte | 62 | ReadDataByIdentifier Service Positive Response ID |
| 2 byte | 11 | Did |
| 3 byte | 02 | Did |
| 4 byte | 03 | Time Stamp |
| 5 byte | 07 | Time Stamp |
| 6 byte | E9 | Time Stamp |
| 7 byte | D0 | Time Stamp |
| 8 byte | 15 | Dem Error Code 0x15 : Error Id |
| 9 byte | 02 | Dem Error Code 0x15 : Error Type |
| 10 byte | 01 | Dem Error Code 0x15 : Error Count |
| 11 byte | 2D | Dem Error Code 0x2D : Error Id |
| 12 byte | 02 | Dem Error Code 0x2D : Error Type |
| 13 byte | 01 | Dem Error Code 0x2D : Error Count |
| 14 byte | 2E | Dem Error Code 0x2E : Error Id |
| 15 byte | 02 | Dem Error Code 0x2E : Error Type |
| 16 byte | 01 | Dem Error Code 0x2E : Error Count |
| 17 byte | 3E | Dem Error Code 0x3E : Error Id |
| 18 byte | 02 | Dem Error Code 0x3E : Error Type |
| 19 byte | 01 | Dem Error Code 0x3E : Error Count |

## Error Code

- Os 및 Dem Error Code는 [현대차표준플랫폼\_시스템에러코드.xlsx]을 따른다.

# FAQ

없음