Abstract

본 문서는 xx 기능(그룹) 구현을 위한 관련 모듈 들 간의 인터페이스를 정의한 상위 수준의 설계서 이다.

<본 문서는 각 기능 그룹 별로 한 건 씩 작성한다. 본문 중 파란색으로 표시된 부분은 예제이므로 적절히 수정하고, “< >”로 둘러싸인 부분은 guide 이므로 최종 작성 후 모두 삭제한다.

\* doxygen을 사용하여 code로부터 함수헤더를 추출하여 본 문서 작성에 활용 가능함.

1) code에 함수 헤더 작성

2) doxygen으로 함수 헤더를 추축하여 본 문서에 삽입

3) 본 문서의 기타 내용을 수작업 입력 및 Peer Reivew (v1.0 완료)

4) 코딩 및 (필요에 따라 함수 헤더 변경)

5) Doxygen으로 다시 함수 헤더를 추출하여 본 문서 내용 update (v1.1) >

Change History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ver. | Date | Change Description | Author | CR |
| v1.0 | 2004.1.1 | 초안 작성 | 홍길동 | 관련 없음 |
|  |  |  |  |  |
|  |  |  |  |  |

Contents

1. Overview 1

1.1 Purpose 1

1.2 Scope 1

1.3 Terminology & Abbreviation 1

1.4 Related & Reference Documents 1

2. Functional Design 2

2.1 기능명1 [예제] DMB Service일 경우의 채널 변경 2

2.2 기능명n 3

3. Inter Module Interfaces 4

3.1 Inter Module Messge 4

3.1.1 Message\_Name 1 4

3.1.2 Message\_Name n 4

3.2 Inter Module API 4

3.2.1 Inter Module API name 1 4

3.2.2 Inter Module API name n 4

4. Global Data 5

4.1 Define 5

4.2 Type definition 5

4.3 Data definition 5

# Overview

<본 기능의 개요를 간단히 설명한다.>

## Purpose

본 문서는 xx 기능(그룹) 구현을 위한 관련 모듈 들 간의 인터페이스를 정의한 상위 수준의 설계서 이다

## Scope

본 문서는 <어떤 부분> 의 소프트웨어 상위 설계를 정의한 문서이며 각 서브시스템, 블록들에 대하여 아래의 내용들을 포함한다.

* 공통적으로 사용되는 데이터 타입 정의
* 인터페이스 정의 및 function 프로토타입
* 식별된 외부 변수
* 데이터 및 컨트롤 플레인 동작

## Terminology & Abbreviation

Terminology

용어 : 설명

모듈 : DTV에서는 시스템 수준에서 다음과 같은 구성요소를 모듈로 정의한다. 예) UI, CM, PSIP, DASY, CAS, …

Abbreviation

HLD High Level Design

## Related & Reference Documents

Related Documents

[1] 작성자, XX 요구명세서, DMB-SW-SRS-01, v1.0, 200x.0x.0x.

[2] 작성자, XX 구조설계서, Pjt-SW-SAD-0x, v1.0, 200x.0x.0x.

Reference Documents

[3] SW기능설계서템플릿, v1.2++, 2011.7.15

# Functional Design

<기능설계는 하나의 기능이 각 모듈들 간에 어떤 관계(interface)를 가지고 있는지를 정의하는 상위수준의 설계이다. 기능설계에서는 모듈간 interface 위주로 설명하고, 모듈 내부의 상세 설명은 LLD에서 한다.

SRS에서 식별된 기능(Use Case)별로 하위 절로 구분하여 Sequence Diagram위주로 모듈간의 관계를 나타낸다.>

## 기능명1 [예제] DMB Service일 경우의 채널 변경

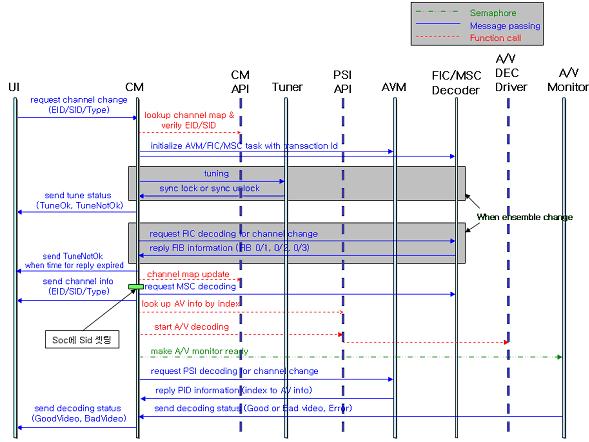


그림2.1 채널 변경시의 Message Sequence Diagram

Figure 2.1 은 채널 변경 시 변경하려는 Service의 Type이 DMB Service일 때의 MSC를 나타낸 것이다. 우선 UI로부터 채널변경 요청 메시지를 수신한 CM은 CM API를 이용하여 메시지에 실려오는 변경하려는 서비스 정보가 유효한 것인지 확인한다. 확인 결과, [유효함]일 경우에는 FIC, MSC Decoder와 AVM을 초기화 하고, 변경하려는 Service가 현재 방송되는 Ensemble과 다를 때에만 Tuner 모듈에 Tuning 명령을 전달한다. Tuning에 대한 결과는 Tuner Task로부터 수신 후에 UI에 전달한다. 확인 결과가, [유효하지 않음]일 경우에는 채널 변경 Rejection에서 자세히 기술한다.

FIC Decoder에게 FIC Decoding을 요청과 응답을 수신하는 것은 변경하려는 Service가 현재 방송되는 Ensemble과 다를 때에만 수행한다. 응답시간이 초과할 경우에는 UI에게 Tune Not Ok 메시지를 전달한다. 응답시간 내에 응답을 수신한 후에는, Channel Map을 갱신하고 SOC에 변경할 Service ID를 설정한다. MSC decoder에는 MSC Decoding을 요청한 다음 UI에게 실제 변경할 Service에 대한 정보를 전달한다. 그 후에, PSI API를 이용하여 AV 정보를 찾고 A/V Decode을 Start시킨다. AV Monitor Task는 CM이 AV Monitor Task에게 Semaphore를 Give함으로써 구동되며 구동시키는 시점은 AV Decoder가 디코딩을 시작했을 때이다. 이 후에 PSI Decoder에 Decoding을 요청한 후 응답을 기다린다. 응답시간이 초과될 경우에는 UI에게 Tune Not Ok 메시지를 전달한다. A/V Monitor로부터는 Decoding Status를 수신 받아 UI에 Decoding 결과를 전달한다. AV Monitor Task에 대해 좀 더 설명하면, AV Monitor Task은 CM이 구동시키며 중지시킨다. CM이 AV Monitor Task에게 Semaphore를 Give함으로써 작동되며 이 시점은 전에 언급하였듯이 AV Decoder가 디코딩을 시작했을 때이다. Semaphore를 받은 AV Monitor는 Block 된 상태에서 Ready 상태로 전환하여 다른 Task로부터 메시지를 받을 수 있는 상태가 된다. 중지시킬 때에는 CM이 AV Monitor에게 [stop AV monitor] 메시지를 전달하며 AV Monitor로 Semaphore를 Take함으로써 Ready상태에서 다시 Block상태로 변하게 된다. CM이 AV Monitor를 중지시키는 시점은 우선 Channel Change 메시지를 받은 직후와 AV decoder를 중지시킨 직후이다.

## 기능명n

# Inter Module Interfaces

## Inter Module Messge

<Sequence Diagram에서 식별된 모듈간에 사용되는 메시지들을 정의한다.>

### Message\_Name 1

#define MSG\_UI2CM\_CHANNEL\_CHANGE MSG\_BASE\_CM(0x1005) /\* Type B, CM\_ChannelChange\_T\* \*/

### Message\_Name n

## Inter Module API

<Sequence Diagram에서 식별된 모듈간에 사용되는 API들을 정의한다.>

### Inter Module API name 1

<모듈간 API는 이곳에서 파라메터만 정의하고 내부 알고리즘에 대한 정의는 해당 모듈의 LLD에서 한다.>

/\*---------------------------------------------------------------------------------------------------------------

\* FUNCTION: API\_STATUS API\_CM\_ClearMap(API\_CHANNEL\_MAP\_T mapType)

\* DESCRIPTION: Clear the specified channel map.

\* INPUT: mapType ; Specify the map to clear

\* OUTPUT: OK ; if succeeded

\* NOT\_OK ; else

-----------------------------------------------------------------------------------------------------------------\*/

### Inter Module API name n

# Global Data

<시스템 전반적으로 공유되어야 할 Global Data들을 정의한다. >

## Define

#define CH\_PACKET\_NUM 10

#define CH\_BANNER\_NUM 10

## Type definition

typedef struct

{

UINT08 numOfVCh; /\* 0x00: Number of Virtual Channels \*/

UINT08 PSIP\_mode; /\* 0x01: PSIP mode for this p-Channel \*/

UINT16 TS\_ID; /\* 0x02: Transport Stream ID \*/

CM\_VCH\_INFO\_T \*pVchInfo; /\* 0x04: Pointer to virtual channels \*/

} CH\_PAC\_T;

## Data definition

CH\_PAC\_ARRAY\_T gChPacket[CH\_PACKET\_NUM];

CH\_BAN\_ARRAY\_T gChBanner[CH\_BANNER\_NUM];