

GSR-006
rev 1.00.00

COP

변경 내역

버전	배포 날짜	변경사항
1.10.00	2005.10.11	초안작성

지적 재산권에 대한 공지

본 문서 및 문서와 함께 제공되는 일체의 정보는 SK 텔레콤의 WIPI 추가 규격을 명시함으로써, SK 텔레콤에 납품하는 단말기 제조사들 및 콘텐츠 개발업체가 WIPI 추가 규격을 구현하도록 하기 위함이다. 본 문서 및 문서와 함께 제공되는 일체의 정보는 SK 텔레콤의 소유물이다. 이의 일부 또는 전부는 SKT WIPI 를 위해서만 SKT 의 승인 아래 복사되거나 배포될 수 있다.

에스케이텔레콤 주식회사

목 차

1 일반사항	5
1.1 개요	5
1.2 목적	5
1.3 규격의 범위.....	5
1.4 용어정의	5
2 의존성	6
2.1 WIPI 의존성.....	6
2.2 GIGA 의존성	6
2.3 GIGA 의존성	6
2.4 OAT 의존성.....	6
3 제조사 포팅 가이드	7
4 개발자 가이드	8
4.1 Type 정의.....	8
4.2 C API	8

세부목차

1 일반사항	5
1.1 개요	5
1.2 목적	5
1.3 규격의 범위.....	5
1.4 용어정의	5
2 의존성	6
2.1 WIPI 의존성.....	6
2.2 GIGA 의존성	6
2.3 GIGA 의존성	6
2.4 OAT 의존성.....	6
3 제조사 포팅 가이드	7
4 개발자 가이드	8
4.1 Type 정의.....	8
4.2 C API	8
4.2.1 OEMC_gCOP_Initialize	10
4.2.2 OEMC_gCOP_Prepare.....	11
4.2.3 OEMC_gCOP_Unprepare	12
4.2.4 OEMC_gCOP_Terminate	13
4.2.5 OEMC_gCOP_PMode.....	14
4.2.6 OEMC_gCOP_Clock	15
4.2.7 OEMC_gCOP_CreateHandle	16
4.2.8 OEMC_gCOP_LockData	17
4.2.9 OEMC_gCOP_UnlockData.....	18
4.2.10 OEMC_gCOP_ReleaseHandle.....	19

1 일반사항

1.1 개요

HW가속기 초기화 및 실행 이미지 로딩, HW 가속기 모드 제어, 내부 메모리 및 레지스터 Access 를 실행하기 위한 API는 다음과 같다. GIGA CLASS 3에서부터 포함되어 졌다.

1.2 목적

본 규격은 SK Telecom(주)(이하 “SK Telecom” 이라 한다.)의 COP를 위한 요구사항 및 규격을 명시하는데 목적이 있다.

1.3 규격의 범위

본 문서는 SK Telecom에서 서비스 예정인 COP를 위한 규격에 대하여 기술한다.

1.4 용어정의

본 문서에서 언급되는 용어들을 정의한다.

2 의존성

본 규격 구현을 위하여, 필요로 하는 WIPI 버전과, 타 WSR 및 본 규격의 검증을 위해 필요한 OAT 버전을 명시한다.

2.1 WIPI 의존성

본 WSR 규격은 아래 WIPI Core 버전 이상이 적용된 단말에 포팅 되어야 한다.

Base WIPI Core 버전
WIPI 1.08.08

2.2 GIGA 의존성

본 GSR 규격은 아래 GIGA Core 버전 이상이 적용된 단말에 포팅 되어야 한다.

Base GIGA Core 버전
GIGA 2.00.00

2.3 GIGA 의존성

본 GSR을 포팅하기 위하여서는 아래 명시된 GSR들을 모두 포팅 하여야 한다.

의존성을 갖는 GSR	설 명

2.4 OAT 의존성

본 규격의 테스트를 위해서는 다음 버전의 OAT가 필요하다.

3 제조사 포팅 가이드

(*) 추후 배포 예정임

4 개발자 가이드

4.1 Type 정의

Type	Define
BOOL	Unsigned char
U32	Unsigned long
U16	Unsigned short
U8	Unsigned char
S32	Signed long
S16	Signed short
S8	Signed char
PVOID	Void *
HCOP_MEM	int
INT32	signed long

4.2 C API

Function	Description
U32 OEMC_gCOP_Initialize()	가속기를 초기화한다 (VOD와 연동되는 경우엔 기가엔진의 Initialize 및 Prepare 처리를 통하여 초기화 기능 구현 가능)
U32 OEMC_gCOP_Prepare(U32 pg_id, U32 *plmgPtr, U32 *plmgCinit)	실행할 프로그램 id (Pg_id)에 따라 실행 코드 이미지 *imgPtr 과 데이터 이미지 *plmgCinit를 필요한 메모리 영역을 확보하여 로딩한 후 각 실행 프로그램 등록을 수행한다
U32 OEMC_gCOP_Unprepare(U32 pg_id)	실행 프로그램 등록을 해제하고 메모리를 반환한다.
U32 OEMC_gCOP_Terminate()	실행 프로그램을 종료한다. (VOD와 연동되는 경우엔 VOD모드로 리턴되도록 초기화 함수(기가엔진의 Initialize 및 Prepare)를 호출해주는 작업을 수행한다.)
U32 OEMC_gCOP_PMode(U32 power_mode)	COP Power Mode Setting
U32 OEMC_gCOP_Clock(U32 clock_speed)	COP Clock Speed 조절

HCOP_MEM OEMC_gCOP_CreateHandle (U32 type , U32 size , const PVOID pData)	가속기의 Heap Memory 영역을 할당하고 주어진 데이터를 복사한다 return 된 Handle을 통하여 가속기 Memory I/F 기능을 제공한다
PVOID OEMC_gCOP_LockData (HCOP_MEM handle, PVOID pBuff)	가속기에 할당된 메모리를 편집하기 위해서 Host 에 메모리를 할당한 후 내용을 복사한 후 어드레스를 알려준다.
void OEMC_gCOP_UnlockData (HCOP_MEM handle)	Host 에서 편집이 끝난 후 COP에 할당되어 있는 메모리에 편집된 내용을 반영하기 위해서 이 함수를 호출한다 호출한 후 Host 에서 편집할 때 사용한 메모리는 사용자가 해제하여 준다.
void OEMC_gCOP_ReleaseHandle (HCOP_MEM handle)	가속기의 Heap Memory 에 할당된 지정된 메모리 영역을 해제해 준다.

4.2.1 OEMC_gCOP_Initialize

설명

기가 가속 모듈을 초기화한다

프로토타입

U32 OEMC_gCOP_Initialize()

매개변수

없음

반환값

0: 정상적으로 수행한 경우

1: 에러가 발견된 경우.

참고 사항

없음

4.2.2 OEMC_gCOP_Prepare

설명

실행할 프로그램 id (Pg_id)에 따라 실행 코드 이미지 *imgPtr 와 데이터 이미지 *pImgCinit를 필요한 메모리 영역을 확보하여 로딩한 후 각 실행 프로그램 등록을 수행한다

프로토타입

U32 OEMC_gCOP_Prepare(U32 pg_id, U32 *imgPtr, U32 *pImgCinit)

매개변수

pg_id

[in]

실행 이미지 인덱스

imgPtr

[out]

실행 이미지의 포인터

pImgCinit

[out]

실행에 필요한 데이터 이미지에 대한 포인터

반환값

0 : 정상적으로 수행한 경우

1 : 에러가 발견된 경우.

참고 사항

없음

4.2.3 OEMC_gCOP_Unprepare

설명

실행 프로그램 등록을 해제하고 메모리를 반환한다

프로토타입

U32 OEMC_gCOP_Unprepare(U32 pg_id)

매개 변수

pg_id

[in]

실행 모드 아이디

반환값

0: 정상적으로 수행한 경우

1: 에러가 발견된 경우.

참고 사항

없음

4.2.4 OEMC_gCOP_Terminate

설명

실행 프로그램을 종료한다.

프로토타입

OEMC_gCOP_Terminate()

매개변수

없음

반환값

0: 정상적으로 수행한 경우

1: 에러가 발견된 경우.

참고 사항

VOD와 연동되는 경우엔 VOD 모드로 Switching 하도록 VOD 초기화 함수를 호출해 주는 작업을 수행하고 종료를 완료한다.

4.2.5 OEMC_gCOP_PMode

설명

Coprocessor 의 Power Mode를 제어한다

프로토타입

U32 OEMC_gCOP_PMode(U32 power_mode)

매개변수

power_mode

Power Mode 는 아래와 같다

PM_FULL

PM_HALF

PM_SAVE

PM_SLEEP

반환값

0 : 정상적으로 수행한 경우

1 : 에러가 발견된 경우.

참고 사항

없음

4.2.6 OEMC_gCOP_Clock

설명

Coprocessor 의 Clock Speed 를 조절한다

프로토타입

U32 OEMC_gCOP_Clock(U32 clock_speed)

매개변수

clock_speed : programmable 한 clock speed

반환값

0 : 정상적으로 수행한 경우

1 : 에러가 발견된 경우.

참고 사항

고정된 Clock 을 사용할 경우 Optional 로 구현한다

4.2.7 OEMC_gCOP_CreateHandle

설명

가속기의 Heap Memory 영역을 할당하고 포인터에 주어진 데이터를 복사한다 return 된 Handle을 통하여 가속기 Memory I/F 기능을 제공한다.

(The OEMC_gCOP_CreateHandle create data handle and copying given data to special memory)

프로토타입

HCOP_MEM OEMC_gCOP_CreateHandle (U32 type, U32 size, const PVOID pData)

매개변수

type

[in]

데이터 타입

TYPE_CHAR =0

TYPE_WORD =1

TYPE_LONG = 2

size

[in]

데이터 사이즈. (byte 단위)

pData

[in]

데이터 이미지 포인터

NULL 인 경우, COP heap Memory를 size 만큼 할당한 후 0으로 초기화한다

반환값

0 : 정상적으로 수행한 경우

1 : 에러가 발견된 경우.

참고 사항

The OEMC_gCOP_CreateHandle create data handle and copying given data to special memory type parameter can be int, short, long It return a data handle.

4.2.8 OEMC_gCOP_LockData

설명

가속기에 할당된 메모리를 편집하기 위해서 Host에 메모리를 할당하고 내용을 복사한 후 어드레스를 알려준다.

프로토타입

PVOID OEMC_gCOP_LockData (HCOP_MEM handle , PVOID pBuff)

매개변수

handle

[in]

데이터 핸들

pBuff

[in]

데이터를 Copy 해올 Pointer

NULL 이 아닌 경우 pBuff 에 데이터를 복사한다

NULL인 경우 메모리를 할당해서 return 해준다

반환값

NULL : 핸들 오류

NULL 이 아닌 경우 데이터를 복사해온 Pointer

pBuff 를 사용하는 경우 NULL 이 아닌 값을 return 하여 오류와 구분한다

참고 사항

The OEMC_gCOP_LockData lock system memory data and return it's local memory address. You can modify the data unsing pointer.

4.2.9 OEMC_gCOP_UnlockData

설명

Host 에서 편집이 끝난 후 COP에 할당되어 있는 메모리에 편집된 내용을 반영하기 위해서 이 함수를 호출한다 호출한 후 Host 에서 편집할 때 사용한 메모리는 사용자가 해제하여 준다

프로토타입

```
void OEMC_gCOP_UnlockData (HCOP_MEM handle)
```

매개변수

Handle

[in]

데이터 핸들

반환값

없음

참고 사항

The OEMC_gCOP_UnlockData unlock system memory data

Implementation Note

DSP only

4.2.10 OEMC_gCOP_ReleaseHandle

설명

가속기의 Heap Memory 에 할당된 지정된 메모리 영역을 해제해 준다

프로토타입

```
void OEMC_gCOP_ReleaseHandle (HCOP_MEM handle)
```

매개변수

handle

[in]

데이터 핸들

반환값

없음

참고 사항

The OEMC_gCOP_ReleaseHandle release the memory of the data buffer and the handle of the data

Implementation Note