

TCC GMAC 모듈 구조 설계서

김동민

1 모듈 개요.....	2
1.1 모듈 목적.....	2
1.2 모듈의 역할	2
1.3 모듈의 범위	3
2 모듈 구조.....	4
2.1 구현 뷰	5
2.2 모듈의 전체 흐름.....	8
3 주요 데이터 구조.....	10
3.1 tcc_dwmac 구조체	10
4 모듈 인터렉션.....	14
4.1 dwmac-tcc-v2.c.....	14
4.2 dwmac-tcc-ecid.c.....	26
4.3 Dwmac-tcc-debugfs.c	29

1 모듈 개요

1.1 모듈 목적

TCC GMAC 모듈은 다음과 같은 목적으로 설계되었다.

- GMAC 디바이스를 사용하기 위한 초기화 작업

1.2 모듈의 역할

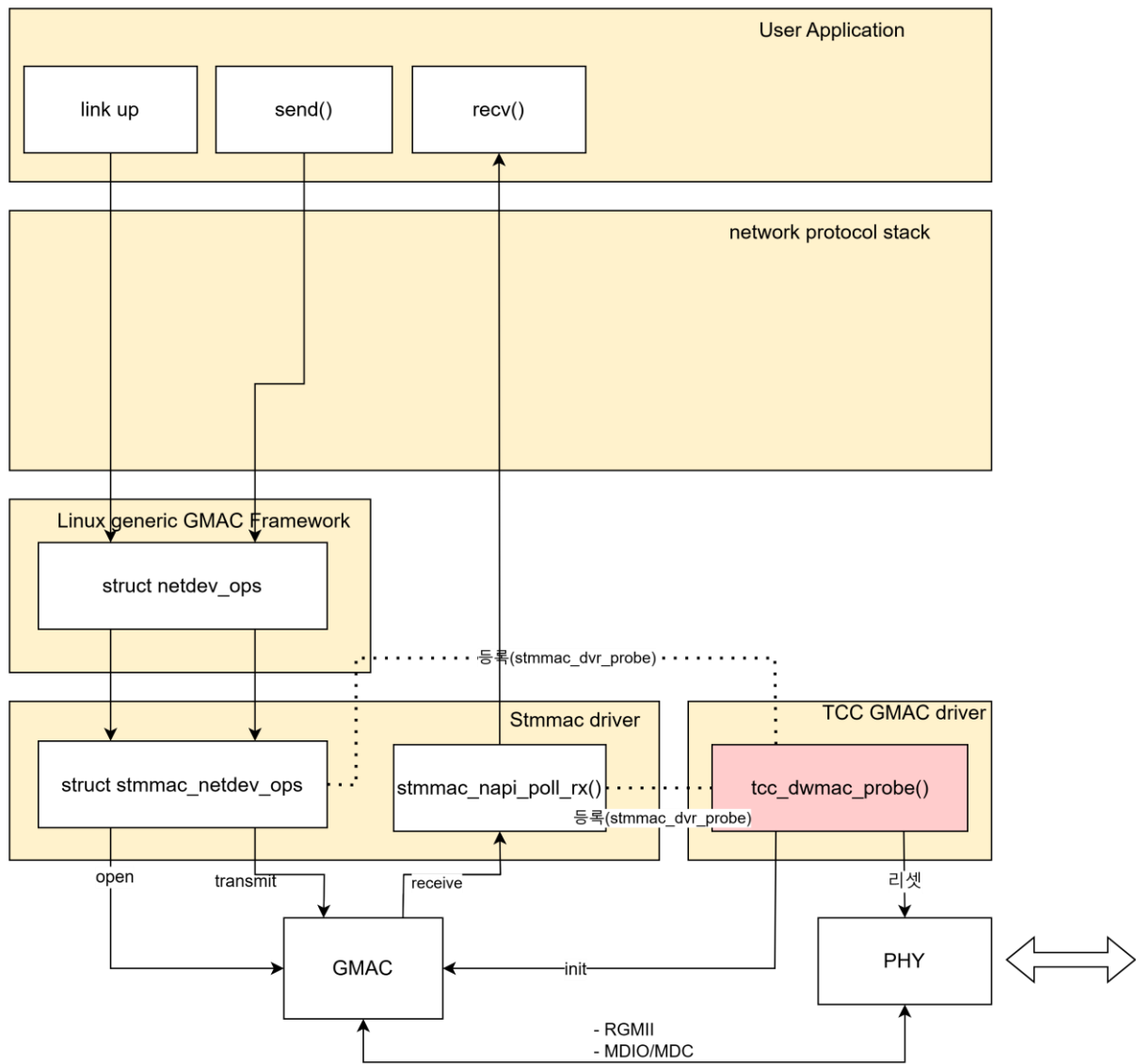
TCC GMAC 모듈의 목적은 다음과 같다.

- 디바이스 트리로부터 디바이스 초기화에 필요한 속성값을 얻음
- MAC 디바이스에 사용할 PHY 간 인터페이스 설정
- PHY 리셋 관련 Gpio 핀 설정
- PHY 리셋
- Clock 설정
- Stmmac driver 에서 사용할 데이터구조체 생성
- 디버깅 파일 생성
- MAC 주소를 얻음

1.3 모듈의 범위

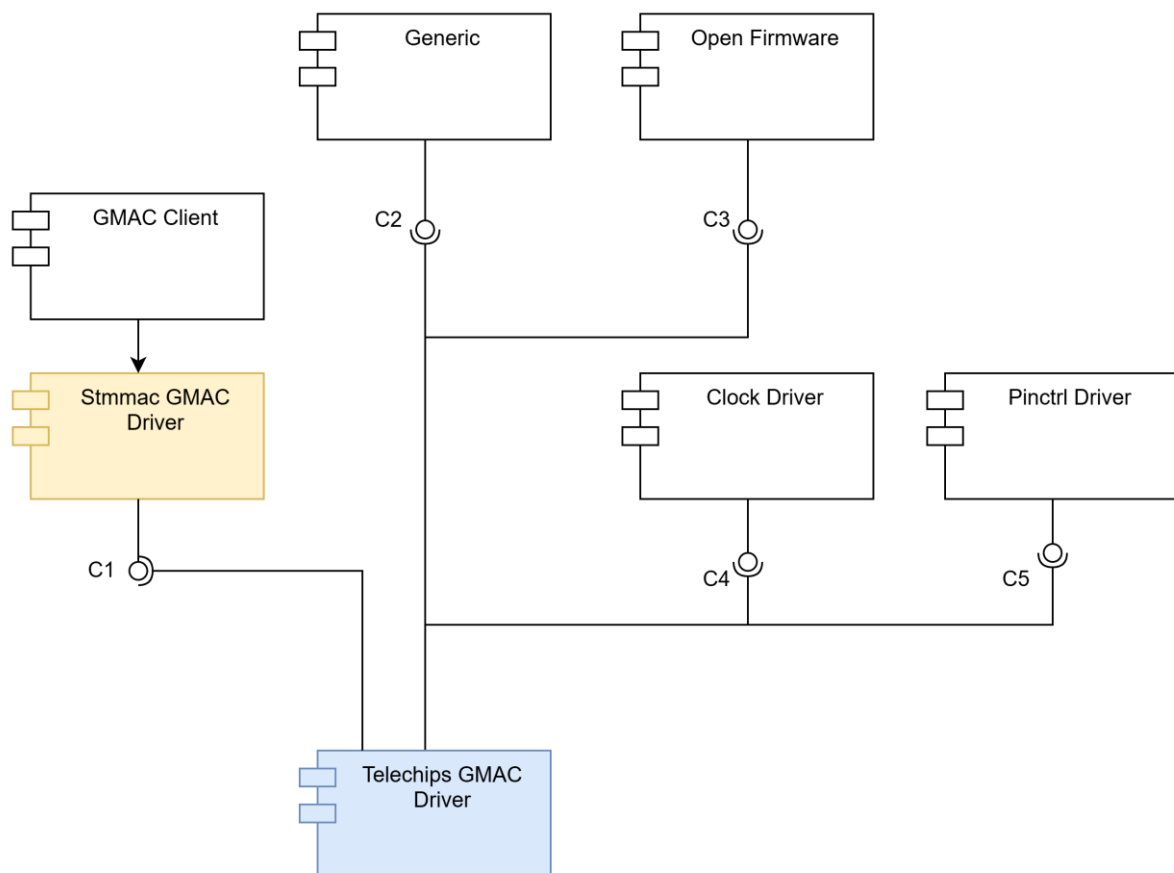
MAC 과 관련된 아키텍처는 아래 그림으로 요약할 수 있다.

- TCC GMAC Driver
 - 디바이스 트리에서 속성 값을 읽어 GMAC 디바이스를 설정한다.
 - stmmac_dvr_probe 를 호출하여 나머지 초기화를 진행한다.(ex. 송수신 함수 등록)
- Stmmac Driver
 - 실제로 송수신을 수행하는 것은 stmmac 의 함수들이다.



2 모듈 구조

2.1 구현 뷰

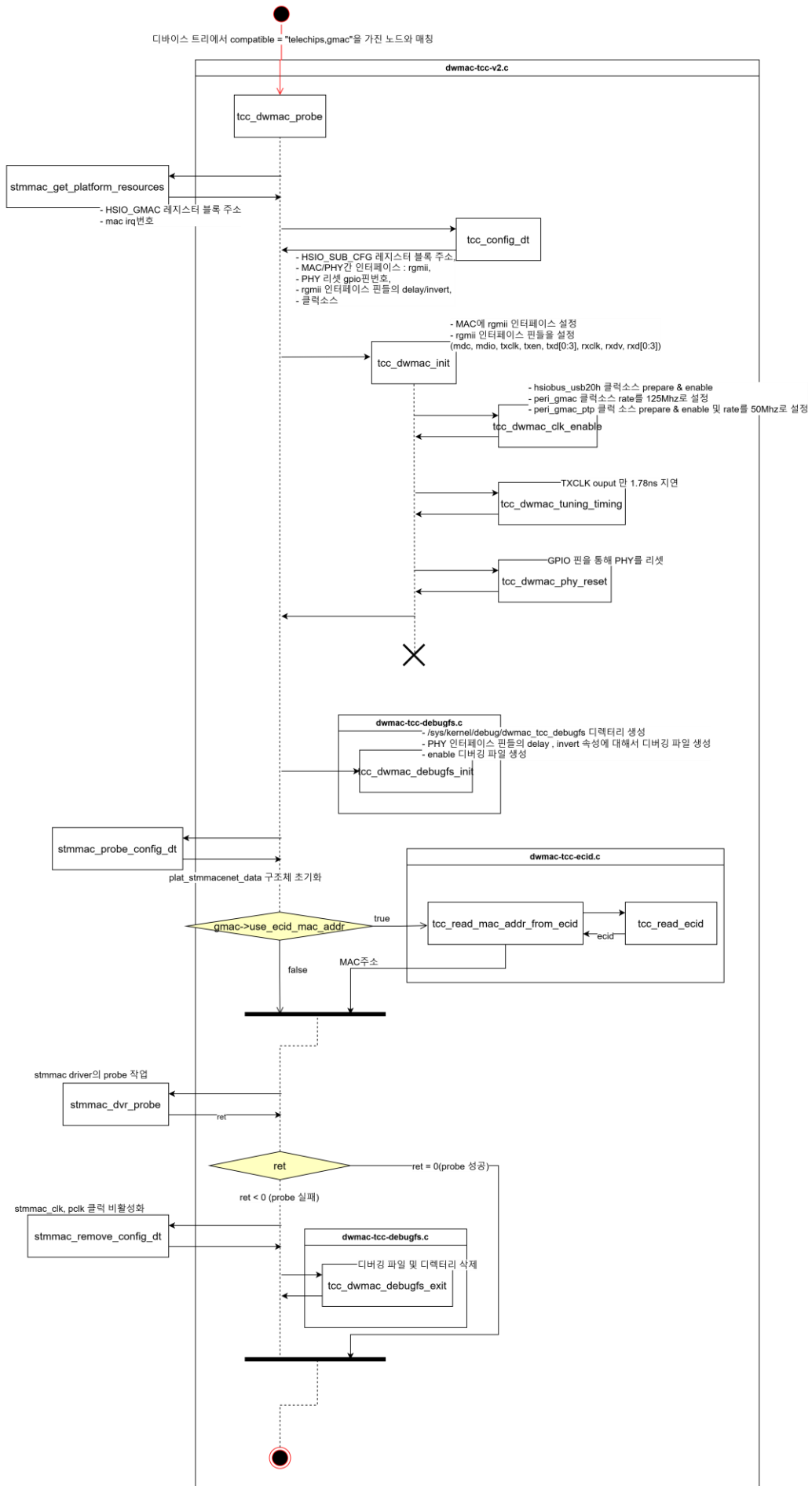


구현요소	역할
Telechips GMAC Driver	Probe 함수에서 GMAC 디바이스와 관련된 초기화 작업을 수행
Stmmac GMAC Driver	실제 GMAC 디바이스를 제어하는 여러 함수를 제공
Generic	커널 메모리 할당 요청을 처리
Open Firmware	디바이스 트리 노드로부터 속성값을 찾아 반환

Clock Driver	Clock 을 설정 및 제어
Pinctrl Driver	핀을 설정 및 제어

커넥터	역할
c1	Stmmac GMAC driver 에서 제공하는 stmmac_dvr_probe 로 송수신 함수 및 net device 를 등록
c2	커널 메모리 할당 요청
c3	Open firmware 에서 제공하는 함수로 디바이스트리에서 하드웨어 속성값을 읽음
c4	GMAC 관련 Clock 을 설정 및 활성화
c5	Phy 리셋 및 Rgmii 인터페이스에 필요한 핀들을 설정

2.2 모듈의 전체 흐름



3 주요 데이터 구조

3.1 tcc_dwmac 구조체

- GMAC 디바이스 설정에 필요한 속성값을 저장하는 구조체

```
struct tcc_dwmac {  
    int phy_mode;  
    uintptr_t hsio_block;  
    uintptr_t gmac_block;  
    struct clk *hsio_clk;  
    struct clk *gmac_clk;  
    struct clk *ptp_clk;  
    struct clk *gmac_hclk;
```

u32 phy_on;
u32 phy_rst;
u32 txclk_i_dly;
u32 txclk_i_inv;
u32 txclk_o_dly;
u32 txclk_o_inv;
u32 txen_dly;
u32 txer_dly;
u32 txd0_dly;
u32 txd1_dly;
u32 txd2_dly;
u32 txd3_dly;
u32 txd4_dly;
u32 txd5_dly;
u32 txd6_dly;
u32 txd7_dly;
u32 rxclk_i_dly;
u32 rxclk_i_inv;
u32 rxdv_dly;
u32 rxer_dly;
u32 rxd0_dly;
u32 rxd1_dly;
u32 rxd2_dly;
u32 rxd3_dly;
u32 rxd4_dly;
u32 rxd5_dly;

```

u32 rxd6_dly;

u32 rxd7_dly;

u32 crs_dly;

u32 col_dly;


// Clause 45 register

unsigned int c45_dev_addr_shift;

unsigned int c45_dev_addr_mask;

unsigned int c45_reg_addr_shift;

unsigned int c45_reg_addr_mask;


bool use_ecid_mac_addr;

bool rmii_tx_clk_off;

#ifdef CONFIG_DEBUG_FS

struct dentry *dbgfs_dir;

#endif

};

```

멤버이름	설명
Phy_mode	MAC 과 PHY 간 인터페이스
Hsio_block	HSIO_SUB_CFG 레지스터 블록 시작 주소(가상주소)
Gmac_block	HSIO_GMAC 레지스터 블록의 시작 주소(가상주소)
Hsio_clk	HSIO 에 사용되는 클럭
Gmac_clk	GMAC 에 사용되는 클럭
Ptp_clk	PTP(Precision Time Protocol)을 지원하는 클럭
Gmac_hclk	GMAC 용 고속 클럭
Phy_on	PHY 를 켜는 데 사용하는 GPIO 핀

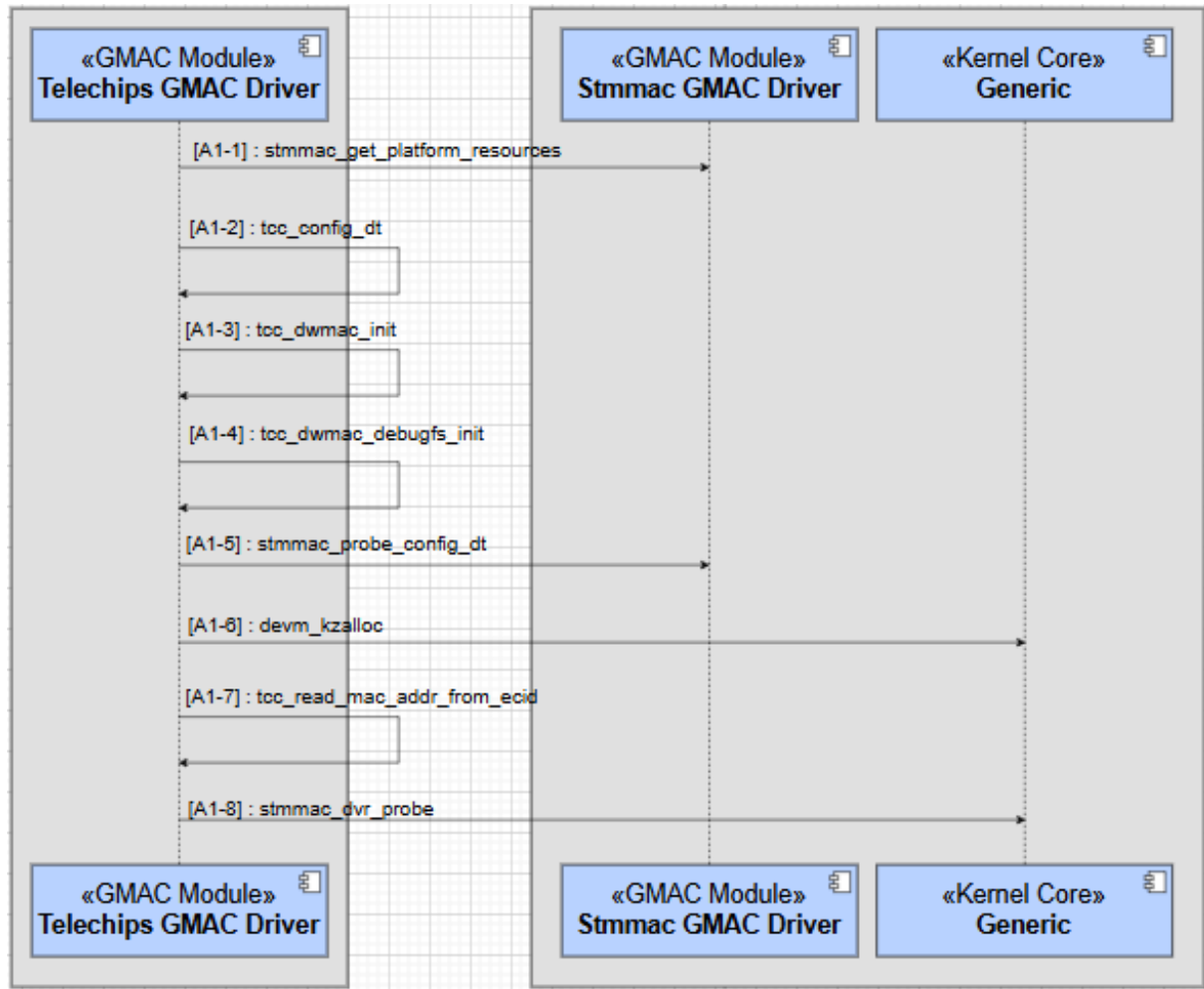
Phy_rst	PHY 를 리셋하는 데 사용하는 GPIO 핀
Txclk_i_dly	송신 클럭 input delay
Txclk_i_inv	송신 클럭 input 반전
Txclk_o_dly	송신 클럭 output delay
Txclk_o_inv	송신 클럭 output 반전
Txen_dly	txen 신호 delay
Txer_dly	Txer 신호 delay
Txd(0~7)_dly	데이터 송신 신호 delay
Rxclk_i_dly	수신 클럭 input delay
Rxclk_i_inv	수신 클럭 input 반전
Rxdv_dly	Rxdv 신호 delay
Rxer_dly	Rxer 신호 delay
Rxd(0~7)_dly	데이터 수신 신호 delay
Crs_dly	반송파 감지(carrier sense) 신호 delay
Col_dly	충돌 신호 delay
Use_ecid_mac_addr	Ecid 로 부터 mac 주소를 가져올지 여부
Rmii_tx_clk_off	Rmii 인터페이스를 사용할 때, 송신 클럭 off 여부
Dbgfs_dir	Debugfs 파일시스템 디렉터리의 dentry 구조체

4 모듈 인터렉션

4.1 dwmac-tcc-v2.c

- 해당 파일의 목적 : 부팅시 디바이스 트리에서 GMAC 관련 속성값을 읽어와 GMAC 디바이스를 초기화한다.

MAD-IV-ID	인터랙션 뷰 명
GMAC-001	Tcc_dwmac_probe

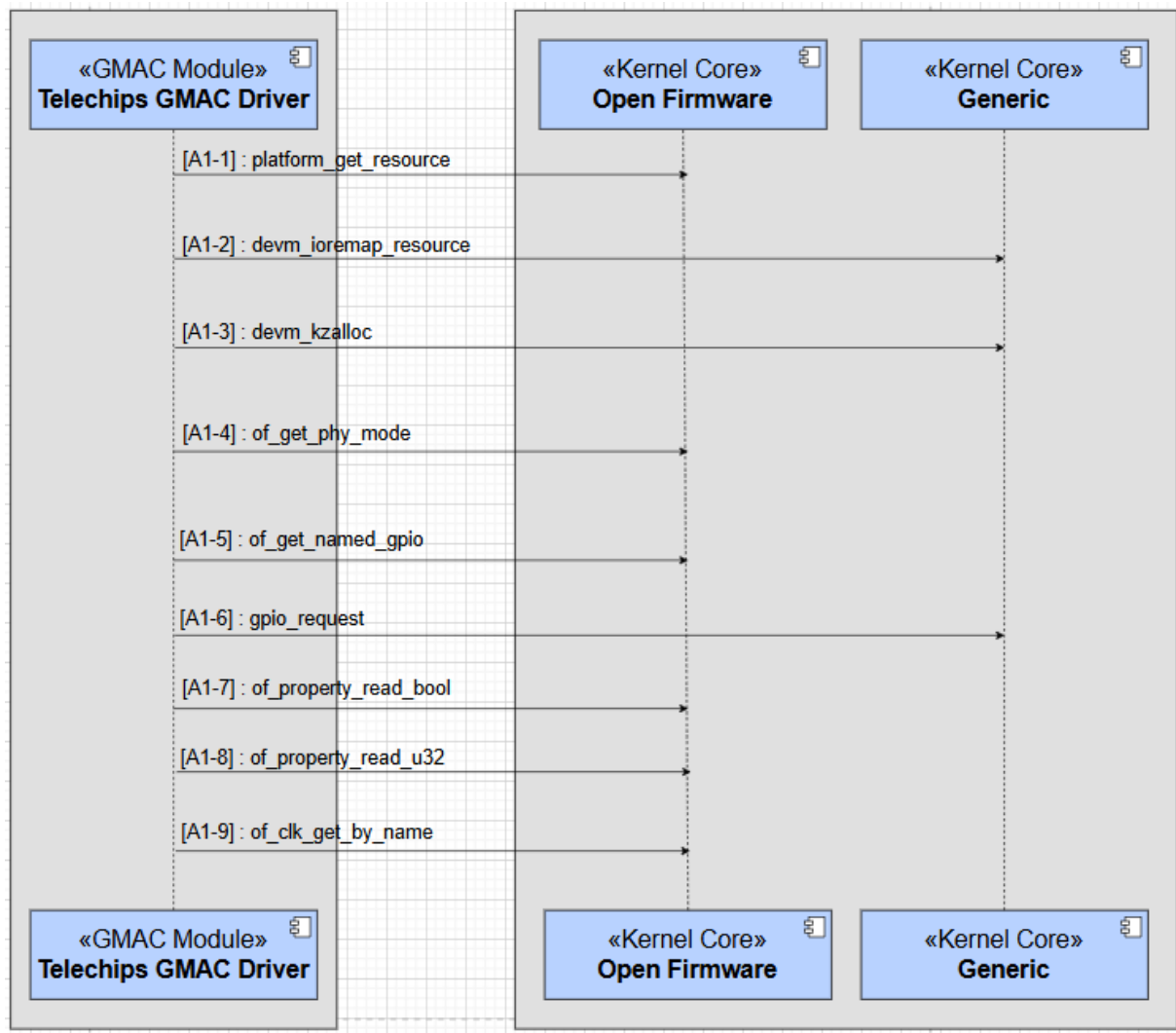


No	Message	Description
1	Stmmac_get_platform_resource	Device tree 로부터 속성을 가져옴. - HSIO_GMAC 레지스터 블록 시작주소 - mac irq 번호
2	Tcc_config_dt	Device tree 로부터 속성을 가져옴. - HSIO_SUB_CFG 레지스터 블록 시작주소 - Phy 간 인터페이스(rgmii) - Phy 리셋용 gpio 핀 번호 - Rgmii 인터페이스 핀들의 delay, invert - GMAC 에 필요한 클럭 소스

3	Tcc_dwmac_init	<p>1. GMAC 디바이스에 사용할 인터페이스(rgmii) 설정.</p> <p>2. Rgmii 인터페이스로 사용할 gpio 핀들을 설정. (mdc, mdio, txclk, txen, txd[0:3], rxclk, rxdv, rxd[0:3])</p> <p>3. GMAC 관련 클럭 활성화</p> <p>4. Rgmii 인터페이스 핀들의 delay, invert 설정</p> <p>5. Phy(realtek)을 리셋</p>
4	Tcc_dwmac_debugfs_init	<p>Debugfs 파일시스템에 rgmii 인터페이스 핀들의 delay, invert 를 설정할 수 있는 디버깅파일 생성</p>
5	Stmmac_probe_config_dt	<p>디바이스 트리에서 GMAC 관련 속성값을 읽어옴</p>
6	Devm_kzalloc	<p>MAC 주소를 저장하기 위한 커널메모리를 할당</p>
7	Tcc_read_mac_addr_from_ecid	<p>ECID로부터 MAC 주소를 읽어옴.</p>
8	Stmmac_dvr_probe	<p>Stmmac 에서 제공하는 probe 기능 수행</p> <ul style="list-style-type: none"> • 송신함수가 포함된 stmmac_netdev_ops 등록 • 수신을 수행하는 stmmac_napi_poll_rx 등록

MAD-IV-ID	인터랙션 뷰 명
GMAC-002	Tcc_config_dt

디바이스 트리로부터 GMAC 디바이스 초기화에 필요한 속성정보를 읽어온다.



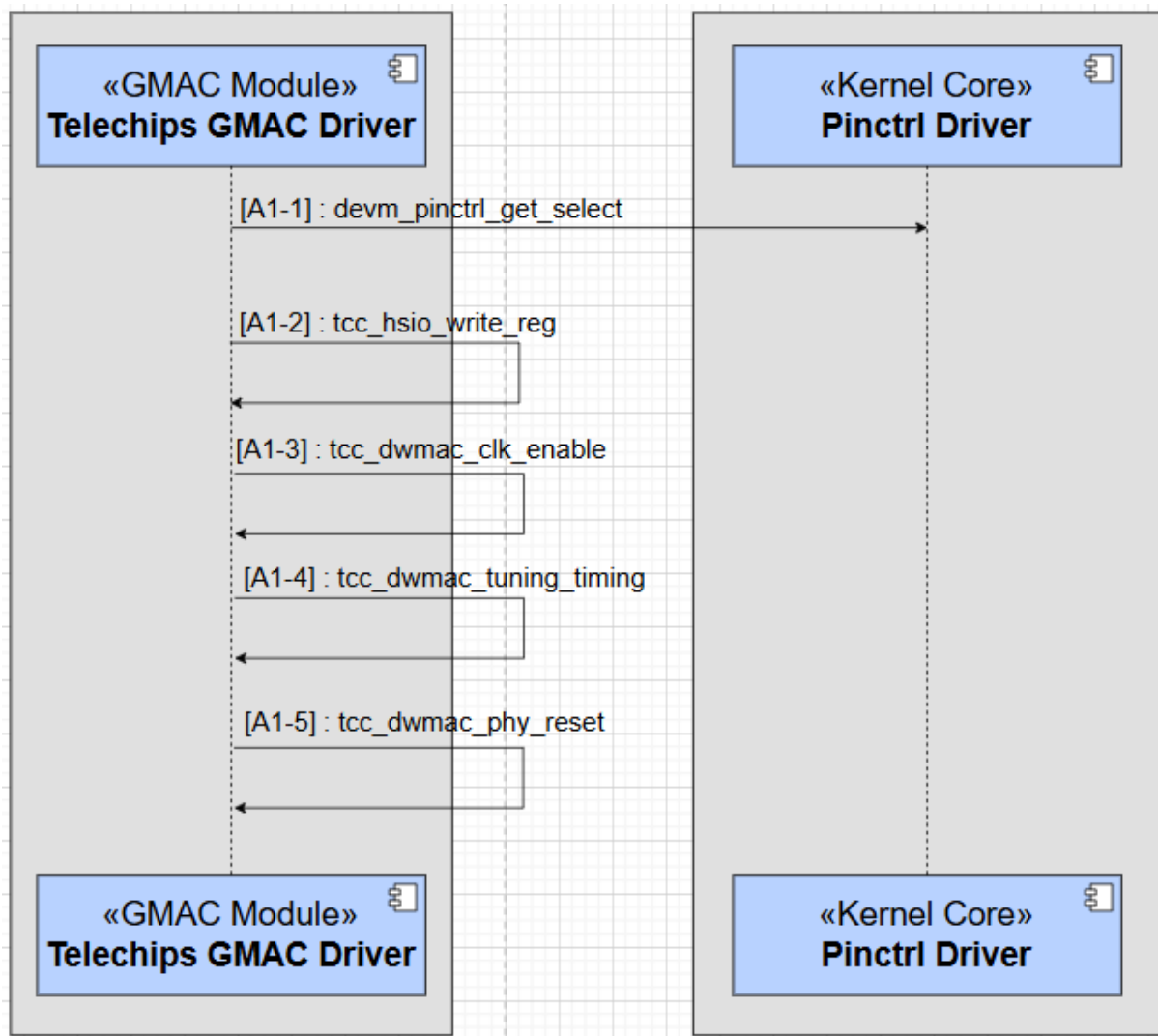
No	Message	Description
1	Platform_get_resource	Device tree로부터 속성값을 읽음 - HSIO_SUB_CFG 레지스터 블록의 시작주소(물리주소) + 크기
2	Devm_ioremap_resource	물리주소를 가상주소로 매핑
3	Devm_kzalloc	Tcc_dwmac 구조체를 위한 커널메모리를 할당
4	Of_get_phy_mode	Device tree로부터 속성값을 읽음 - Mac,phy 간 인터페이스 -> rgmii
5	Of_get_named_gpio	Device tree로부터 속성값을 읽음 - Phy 리셋용 gpio 핀번호
6	Gpio_request	Phy 리셋용 Gpio 핀 사용 요청
7	Of_property_read_bool	Device tree에 해당 속성 존재 여부 반환

		<ul style="list-style-type: none"> - ecid-mac-addr -> false - telechips,rmii_tx_clk_off -> false
8	Of_property_read_u32	<p>Device tree로부터 32bit 정수값을 읽음</p> <ul style="list-style-type: none"> - phy interface 핀들에 대한 delay 값, invert 여부
9	Of_clk_get_by_name	<p>Device tree로부터 어떤 clk source 를 사용할지 알아내고, clock driver로부터 clk source 를 가져옴.</p> <ul style="list-style-type: none"> - "gmac-pclk" -> peri_gmac - "gmac-hckl" -> hsiobus_usb20h - "ptp-pclk" -> peri_gmac_ptp

MAD-IV-ID	인터랙션 뷰 명
GMAC-003	Tcc_dwmac_init

디바이스를 초기화

- GMAC 의 phy interface 설정
- 클럭 활성화
- Delay 설정
- PHY 리셋

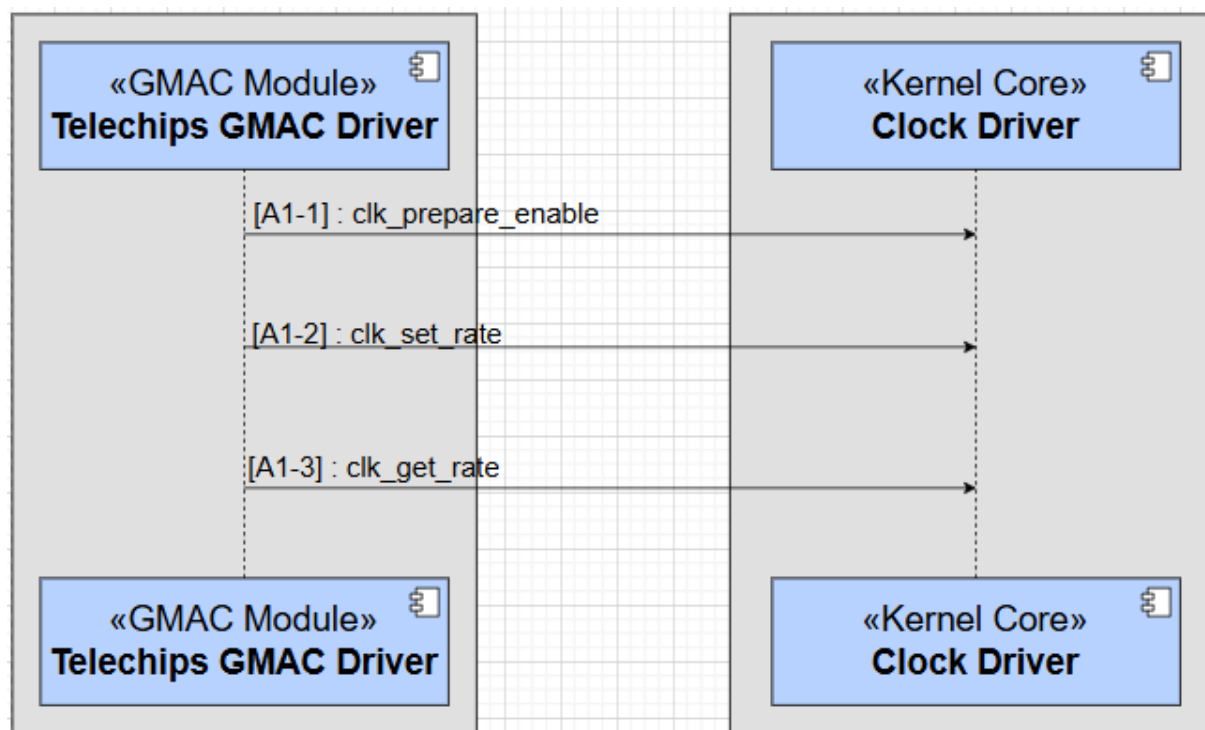


No	Message	Description
1	Devm_pinctl_get_select	<p>Device tree 를 참고하여 phy 인터페이스용 핀들을 설정</p> <ul style="list-style-type: none"> - Rgmii 이므로 mdc, mdio, txclk, txen, txd[0:3], rxclk, rxdv, rxd[0:3] gpio 핀들을 설정
2	Tcc_hsio_write_reg	<p>HSIO_SUB_CFG -> GMAC_CFG0(+0x68)레지스터에 write</p> <ul style="list-style-type: none"> - TX divider = 1 - TX z 클럭을 1/TXDIV 로 divide (rgmii 이기 때문에 건너뛴) <p>HSIO_SUB_CFG -> GMAC_CFG1(+0x6c)레지스터에 write</p>

		<ul style="list-style-type: none"> - GMAC 클럭 활성화 - GMAC 디바이스에 phy interface 설정 -> rgmii
3	Tcc_dwmac_clk_enable	클럭을 활성화
4	Tcc_dwmac_tuning_timing	Phy interface 핀들의 delay, invert 설정
5	Tcc_dwmac_phy_reset	PHY 디바이스를 reset

MAD-IV-ID	인터랙션 뷰 명
GMAC-004	Tcc_dwmac_clk_enable

GMAC 에 필요한 클럭을 활성화

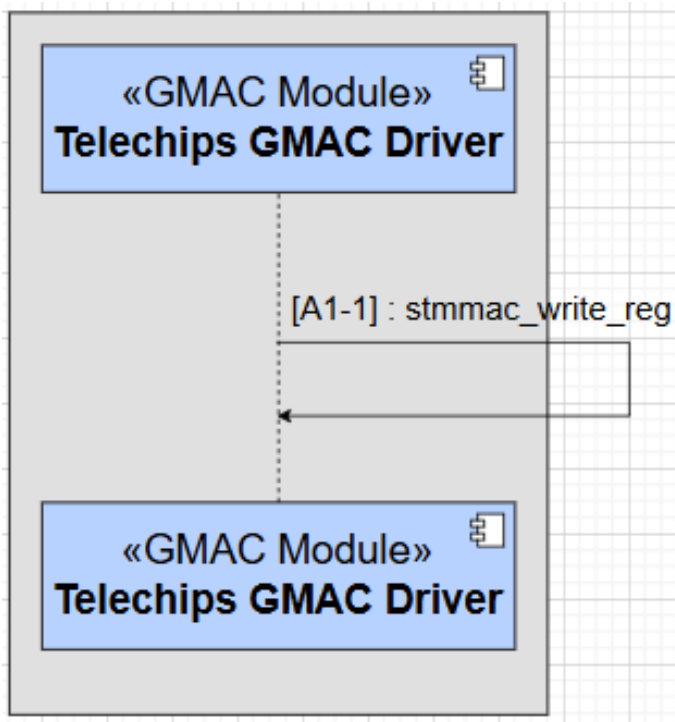


No	Message	Description
----	---------	-------------

1	Clk_prepare_enable	클록을 준비 & 활성화 - Gmac_hclk - Gmac_ptp_clk
2	Clk_set_rate	클록을 특정 rate 로 설정 -> rgmii 이므로 gmac_clk 을 125Mhz 로 설정 -> ptp_clk 을 50Mhz 로 설정
3	Clk_get_rate	클록의 현재 설정된 rate 를 반환 - Gmac_clk 의 rate 를 반환

MAD-IV-ID	인터랙션 뷰 명
GMAC-005	Tcc_dwmac_tuning_timing

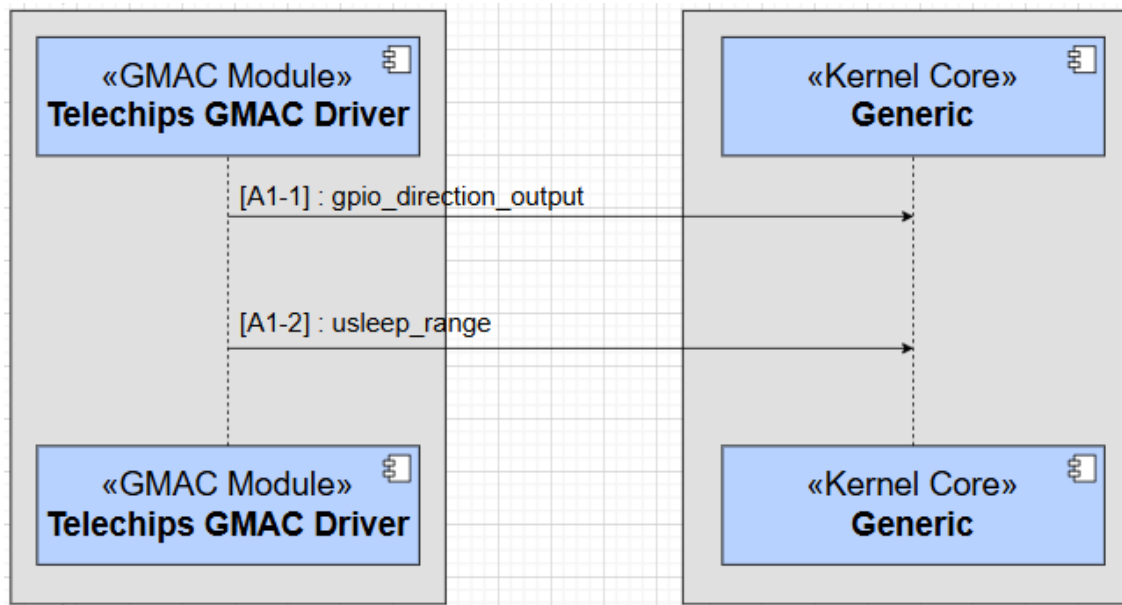
GMAC 에 delay 를 설정



No	Message	Description
1	Stmmac_write_reg	<p>다음 레지스터에 write 하여 phy 인터페이스 핀들의 delay, invert 를 설정</p> <ul style="list-style-type: none"> - HSIO_GMAC->DLYCON0 - HSIO_GMAC->DLYCON1 - HSIO_GMAC->DLYCON2 - HSIO_GMAC->DLYCON3 - HSIO_GMAC->DLYCON4 - HSIO_GMAC->DLYCON5 - HSIO_GMAC->DLYCON0 <p>결과적으로 TXCLK output 에 대해서만 1.78ns 지연.</p> <p>나머지 핀들은 지연 X, invert X</p>

MAD-IV-ID	인터랙션 뷰 명
GMAC-006	Tcc_dwmac_phy_reset

PHY 를 reset 하기위해 PHY reset 용 gpio 핀에 특정패턴으로 출력을 제어한다.



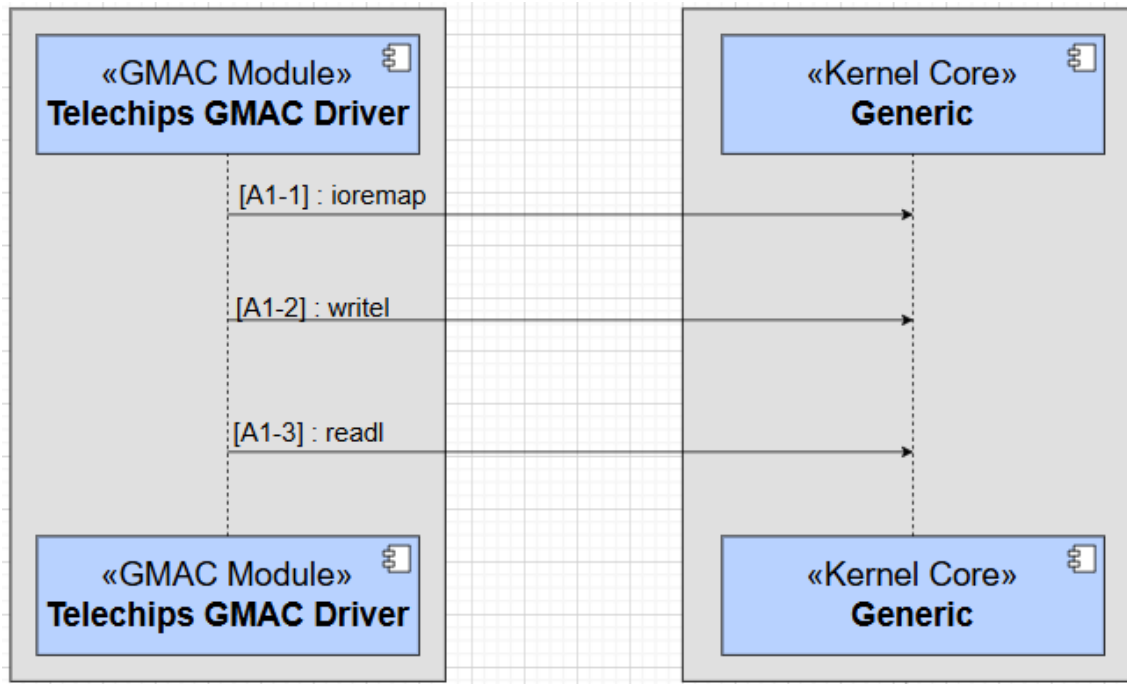
No	Message	Description
1	Gpio_direction_output	특정 gpio 의 출력 설정(PHY reset 용 gpio 핀에 0,1 설정)
2	Usleep_range	Min, max 범위 내에서 sleep

4.2 dwmac-tcc-ecid.c

- 해당 파일의 목적 : ECID 로부터 MAC 주소를 가져온다.

MAD-IV-ID	인터렉션 뷰 명
GMAC-007	Tcc_read_ecid

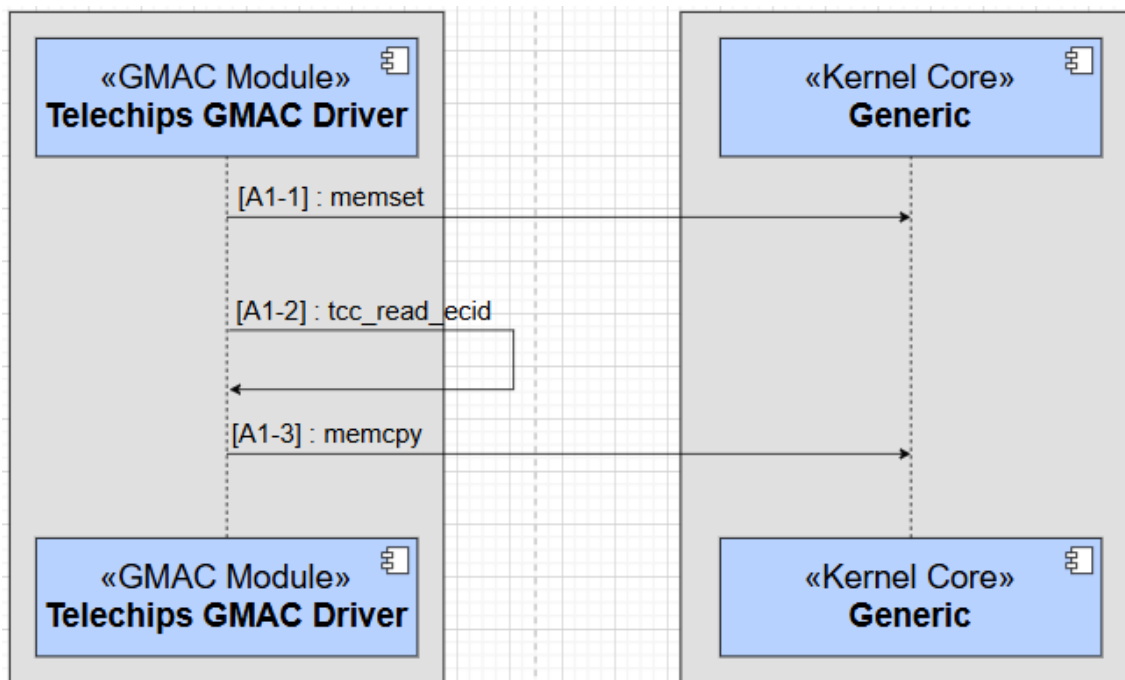
MAC 하드웨어의 레지스터로부터 ecid 를 읽어들인다.



No	Message	Description
1	ioremap	GPIO 물리주소로부터 0x100000 크기를 매핑하고 시작 가상주소를 반환
2	writel	가상주소에 매핑된 물리주소의 메모리에 값을 쓴다.
3	readl	가상주소에 매핑된 물리주소로부터 값을 읽는다. (ecid 를 읽음)

MAD-IV-ID	인터랙션 뷰 명
GMAC-008	Tcc_read_mac_addr_from_ecid

MAC 주소를 완성하여 stmmac_mac_addr 에 저장한다.



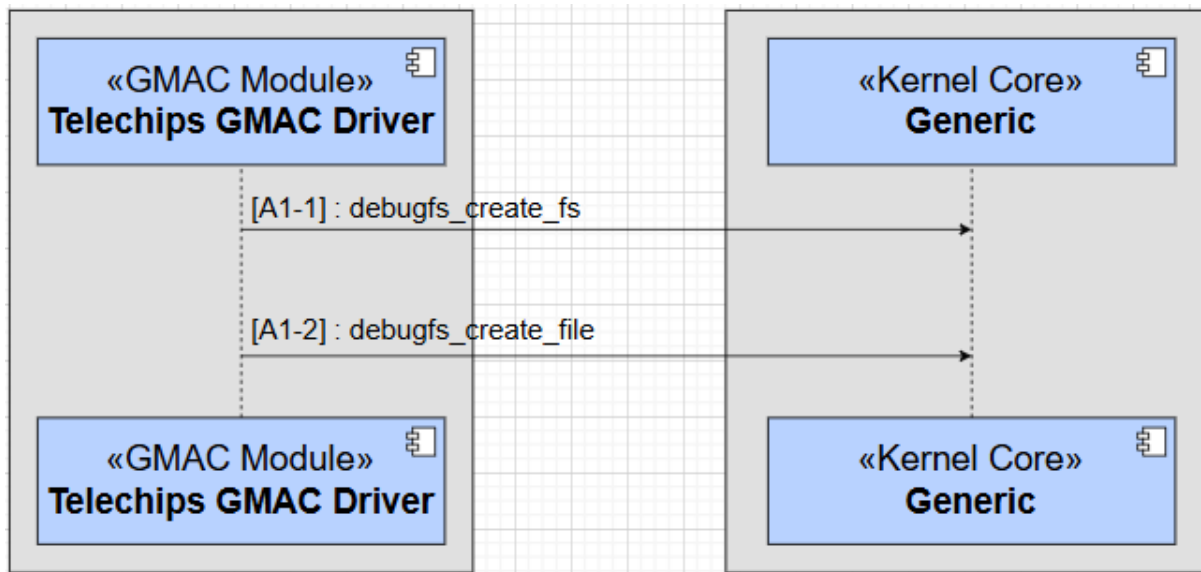
No	Message	Description
1	memset	MAC 주소를 담을 메모리를 0 으로 초기화
2	Tcc_read_ecid	ECID 를 읽어온다.
3	memcpy	ECID 로부터 뽑아낸 MAC 주소를 stmmac_mac_addr 로 복사

4.3 Dwmac-tcc-debugfs.c

- GMAC 디버깅 파일을 생성한다.

MAD-IV-ID	인터렉션 뷰 명
GMAC-012	Tcc_dwmac_debugfs_init

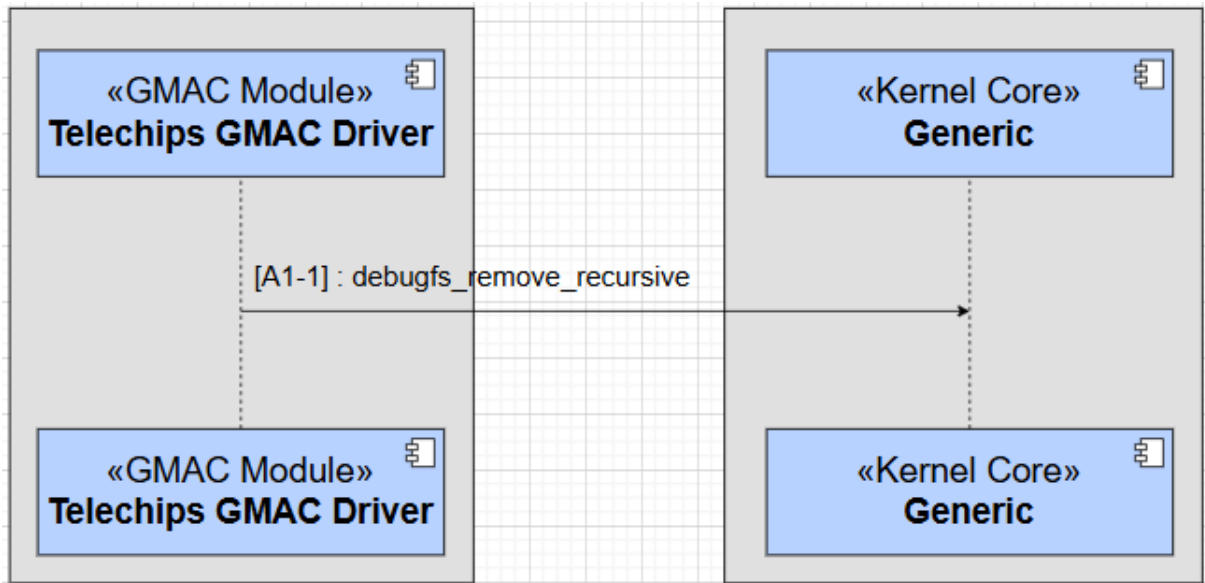
Debugfs 파일시스템에 디버깅 파일들을 생성한다.



No	Message	Description
1	Debugfs_create_dir	Debugfs 파일시스템에 디렉터리 생성 - /sys/kernel/debug/dwmac_tcc_debugfs
2	TCC_DEBUG_FS_CREATE_FILE(=debugfs_create_file)	Phy interface 의 delay, invert 속성을 설정할 수 있는 디버깅 파일을 생성

MAD-IV-ID	인터렉션 뷰 명
GMAC-013	Tcc_dwmac_debugfs_exit

Debugfs 파일시스템에서 디버깅 파일들을 삭제한다.



No	Message	Description
1	Debugfs_remove_recursive	디버깅 파일 삭제(디렉터리 포함)

END