사용자 매뉴얼

1. 디바이스 트리

GMAC 관련 디바이스 트리 속성을 수정하려면 아래의 파일에서 "telechips,gmac" 검색

1-1) <커널 소스 경로>/arch/arm64/boot/dts/tcc/ tcc805x.dtsi

```
gmac: gmac@11C00000 {
    compatible = "telechips,gmac";
    reg = <0x0 0x11C00000 0x0 0x3000>,
    <0x0 0x11DA0000 0x0 0x0008>;
    interrupts = <GIC_SPI 139 IRQ_TYPE_LEVEL_HIGH>;
    status = "disabled";
    index = <1>;
    clocks = <&clk_peri PERI_GMAC &clk_fbus FBUS_HSIO &clk_peri PERI_GMAC_PTP &clk_hsio HSIO
    BUS_GMAC>; //PERI_GMAC0, HSIOBUS, PERI_PTP0, HCLK_GMAC
    clock-names = "gmac-pclk", "hsio-clk", "ptp-pclk", "gmac-hclk";
};
```

속성	값
reg	<"Base address(2 cell)" "Size(2 cell)"> HSIO_GMAC 레지스터 블럭 - 0x0 0x11C00000 : Base address - 0x0 0x3000 : 크기 : Size HSIO_SUB_CFG 레지스터 블럭 - 0x0 0x11DA0000 : Base address - 0x0 0x0008 : Size
interrupts	<"인터럽트 컨트롤러 번호" "인터럽트 번호" "트리거 유형"> 인터럽트 컨트롤러 번호 - GIC_SPI* - GIC_PPI 인터럽트 번호 - 139 트리거 유형 - IRQ_TYPE_NONE - IRQ_TYPE_EDGE_RISING - IRQ_TYPE_EDGE_FALLING - IRQ_TYPE_EDGE_BOTH - IRQ_TYPE_LEVEL_HIGH* - IRQ_TYPE_LEVEL_LOW
Status	장치 활성화/비활성화 - "okay" - "disabled"*

1-2) <커널 소스 경로>/arch/arm64/boot/dts/tcc/tcc8050-linux-ivi.dtsi

```
&gmac {
    status = "okay";
    compatible = "snps,dwmac-4.10a","telechips,gmac";
    phyrst-gpio = <&gpmc 11 0>;
    phy-interface = "rgmii";
    interrupt-names = "macirq";
    phy-mode = "rgmii";
    snps,txpbl = <8>;
    snps,rxpbl = <2>;
    //ecid-mac-addr;
    txclk-o-dly = <31>;
    txclk-o-inv = <0>;
    txclk-i-dly = <0>;
    txclk-i-inv = <0>;
    txen-dly = <0>;
    txer-dly = <0>;
    txd0-dly = <0>;
    txd1-dly = <0>;
    txd2-dly = <0>;
    txd3-dly = <0>;
    txd4-dly = <0>;
    txd5-dly = <0>;
    txd6-dly = <0>;
    txd7-dly = <0>;
    rxclk-i-dly = <0>;
    rxclk-i-inv = <0>;
    rxdv-dly = <0>;
    rxer-dly = <0>;
    rxd0-dly = <0>;
    rxd1-dly = <0>;
    rxd2-dly = <0>;
    rxd3-dly = <0>;
    rxd4-dly = <0>;
    rxd5-dly = <0>;
    rxd6-dly = <0>;
    rxd7-dly = <0>;
    crs-dly = <0>;
    col-dly = <0>;
    pinctrl-names = "default", "mii", "rmii", "gmii", "rgmii";
    pinctrl-0 = <>;
    pinctrl-1 = <&gmac1_mdc &gmac1_mdio &gmac1_col &gmac1_crs &gmac1_txer &gmac1_rxer
        &gmac1_txclk &gmac1_txen &gmac1_txd0 &gmac1_txd1 &gmac1_txd2 &gmac1_txd3
        &gmac1_rxclk &gmac1_rxdv &gmac1_rxd0 &gmac1_rxd1 &gmac1_rxd2 &gmac1_rxd3>;
    pinctrl-2 = <&gmac1_mdc &gmac1_mdio
        &gmac1_txclk &gmac1_txen &gmac1_txd0 &gmac1_txd1
        &gmac1_rxer &gmac1_rxdv &gmac1_rxd0 &gmac1_rxd1>;
    pinctrl-3 = <&gmac1_mdc &gmac1_mdio &gmac1_col &gmac1_crs &gmac1_txer &gmac1_rxer
        &gmac1_txclk &gmac1_txen &gmac1_txd0 &gmac1_txd1 &gmac1_txd2 &gmac1_txd3
        &gmac1_txd4 &gmac1_txd5 &gmac1_txd6 &gmac1_txd7
        &gmac1_rxclk &gmac1_rxdv &gmac1_rxd0 &gmac1_rxd1 &gmac1_rxd2 &gmac1_rxd3
        &gmac1_rxd4 &gmac1_rxd5 &gmac1_rxd6 &gmac1_rxd7>;
    pinctrl-4 = <&gmac1_mdc &gmac1_mdio
        &gmac1_txclk &gmac1_txen &gmac1_txd0 &gmac1_txd1 &gmac1_txd2 &gmac1_txd3
        &gmac1_rxclk &gmac1_rxdv &gmac1_rxd0 &gmac1_rxd1 &gmac1_rxd2 &gmac1_rxd3>;
#if 0 /* If using fixed link, enable this */
    fixed-link {
    speed = <1000>;
    full-duplex;
};
#endif
```

속성	값
status	장치 활성화/비활성화 - "okay" - "disabled"(*)
Phyrst-gpio	Phy 리셋용 gpio 핀 정보 <"&gpio그룹" "핀 번호" "플래그"> gpio그룹 : gpmc(gpiomc) 핀번호 : 11 플래그: - GPIO_ACTIVE_HIGH(0)* - GPIO_ACTIVE_LOW(1)
Phy-mode	Mac과 phy간 사용할 인터페이스 - Mii - Rmii - Gmii - Rgmii*
*-dly	신호 지연(step)
*-inv	신호 반전 여부 - 0x00(반전 X)* - 0x01(반전 O)
Pinctrl-n	Mii/rmii/gmii/rgmii 인터페이스를 구성하는 gpio핀 노드 phandle *현재 rgmii 인터페이스이므로 아래의 gpio핀 노드를 참고하여 핀을 설정한다. - gmac1_mdc - gmac1_mdio - gmac1_txclk - gmac1_txen - gmac1_txd0 - gmac1_txd1 - gmac1_txd2

	- gmac1_txd3 - gmac1_rxclk - gmac1_rxdv - gmac1_rxd0 - gmac1_rxd1 - gmac1_rxd2 - gmac1_rxd3
Ecid-mac-addr	ecid에서 mac주소 얻음 여부 (주석처리하면 비활성화)

2. 빌드 및 포팅

1-1) ./autolinux -c configure

```
vboxuser@topst:~/autolinux/topst$ ./autolinux -c configure
The command is configure or Add configuration options(sdk,core,manifest)
Configure Start

Choose a core to build
  1.tcc8050-main
  2.tcc8050-sub
Choose core(1-2): 1
Choose the Features at tcc8050-main
* 1.  network             Install network packages
* 2.  ssh-server-openssh  Install openssh with network packages
* 3.  meta-micom          Enable Micom
* 4.  meta-update         Enable Update
* 5.  with-subcore        Booting with sub-core
* 6.  with-cr5core        Booting with cr5-core
* 7.  support-4k-video    Support 4k Video Contents
  0.Exit
Choose Features enable/disable (1-7): 0
Choose the Features at tcc8050-sub
* 1.  meta-micom          Enable Micom
* 2.  meta-update         Enable Update
  0.Exit
Choose Features enable/disable (1-2): 0
Read configuration from autolinux.config

Do you want to use sstate? (y/n) : y
/home/vboxuser/autolinux/topst/poky/download_web.sh ubuntu
Enabled with-subcore, The autolinux script make new configuration files of Sub-core automaticall
y
================================================================================
SDK=tcc805x_linux_ivi
MANIFEST=tcc8050_linux_ivi_tost_0.1.0.xml
DATE=2022/04/08
MACHINE=tcc8050-main
VERSION=release
FEATURES=network,ssh-server-openssh,meta-micom,meta-update,with-subcore,with-cr5core,support-4k-
video
SUBFEATURES=meta-micom,meta-update
================================================================================
Time taken in 0:00:04.661540
vboxuser@topst:~/autolinux/topst$
```

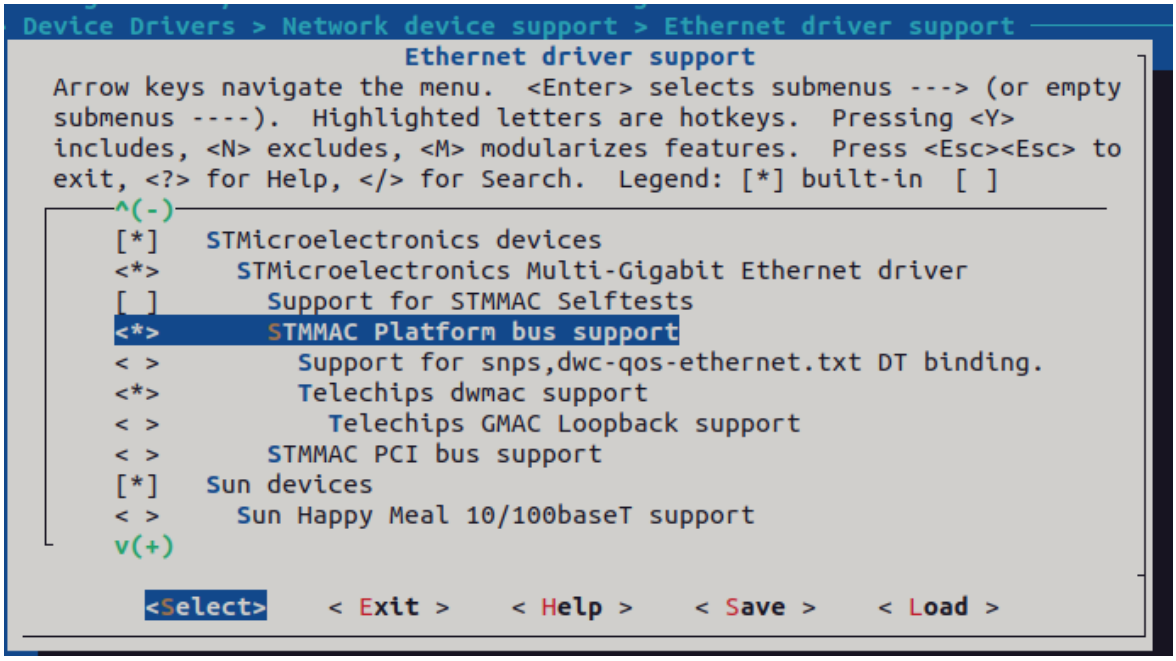
1-2) ./autolinux -c build "linux-telechips -c menuconfig"

--> Device Drivers

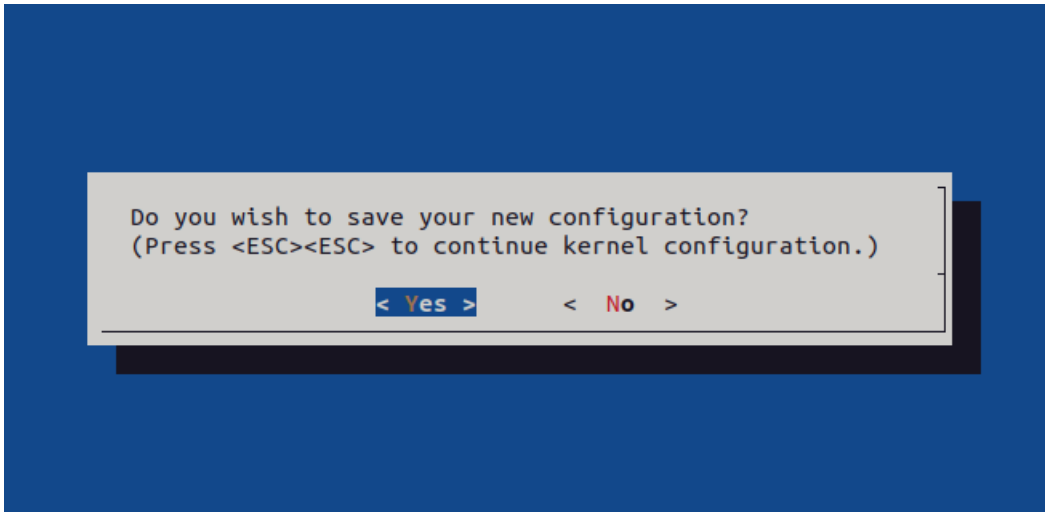
--> Network device support(*)

--> Ethernet driver support(*)

"STMMAC_Platform bus support"선택("Telechips dwmac support"도 자동으로 선택됨.)



save하면 .config 파일 생성



1-3) 리눅스 커널 빌드하기

autolinux –c build

```
vboxuser@topst:~/autolinux/topst$ ./autolinux -c build
Read configuration from autolinux.config

/home/vboxuser/autolinux/topst/poky/download_web.sh ubuntu
Enabled with-subcore, The autolinux script build Sub-core Image automatically in background
Enabled with-cr5, The autolinux script build cr5-core Image automatically in background
Loading cache: 100% |##################################################################################################################################
Loaded 1785 entries from dependency cache.
Parsing recipes: 100% |################################################################################################################################
Parsing of 1125 .bb files complete (1123 cached, 2 parsed). 1787 targets, 268 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.46.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal"
TARGET_SYS           = "aarch64-telechips-linux"
MACHINE              = "tcc8050-main"
DISTRO               = "poky-telechips-systemd"
DISTRO_VERSION       = "3.1.15"
TUNE_FEATURES        = "aarch64 cortexa72 crc crypto"
TARGET_FPU           = ""
LINUX_VERSION        = "5.4.159"
KBUILD_DEFCONFIG     = "tcc805x_linux_ivi_defconfig"
TCMODE               = "default"
INVITE_PLATFORM      = " with-subcore gpu-vz support-4k-video network v4l-utils dp2hdmi wifi-pci-ax200 USB-WEBCAM USB-WiFi-MT7601U
IMAGE_FEATURES       = " debug-tweaks ssh-server-openssh tools-debug eclipse-debug"
SDKIMAGE_FEATURES    = "dev-pkgs"
GCCVERSION           = "arm-9.2"
GLIBCVERSION         = "2.31%"
meta
meta-poky
meta-gplv2           = "HEAD:16634c7cfeea783d70108d31c57fb1e4250b4100"
meta-arm-toolchain   = "HEAD:1cfb0a1d773dd6eac3f387fd68ded7c907402d08"
meta-telechips-bsp   = "HEAD:3b93f65c0817ba3f16754d6c4de84792127f3b0b"
meta-qt5             = "HEAD:e3c49254dfa17fe2ebd3367a2adc7bf6016d718c"
meta-core
```

1-4) fai파일 만들기

autolinux –c make_fai

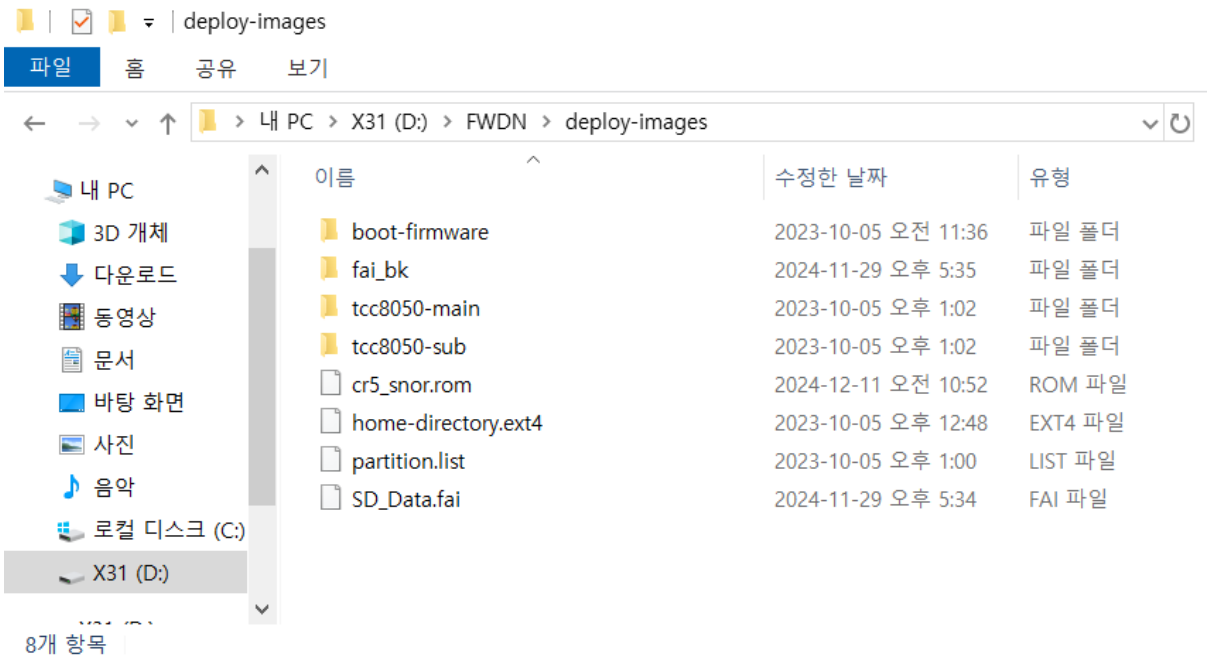
```
vboxuser@topst:~/autolinux/topst$ ./autolinux -c make_fai
Read configuration from autolinux.config

/home/vboxuser/autolinux/topst/poky/download_web.sh ubuntu
Loading cache: 100% |#########################################################################################################################
Loaded 1785 entries from dependency cache.
Parsing recipes: 100% |#######################################################################################################################
Parsing of 1125 .bb files complete (1123 cached, 2 parsed). 1787 targets, 268 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.46.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal"
TARGET_SYS           = "aarch64-telechips-linux"
MACHINE              = "tcc8050-main"
DISTRO               = "poky-telechips-systemd"
DISTRO_VERSION       = "3.1.15"
TUNE_FEATURES        = "aarch64 cortexa72 crc crypto"
TARGET_FPU           = ""
LINUX_VERSION        = "5.4.159"
KBUILD_DEFCONFIG     = "tcc805x_linux_ivi_defconfig"
TCMODE               = "default"
INVITE_PLATFORM      = " with-subcore gpu-vz support-4k-video network v4l-utils dp2hdmi wifi-pci-ax200 USB-WEBCAM USB
IMAGE_FEATURES       = " debug-tweaks ssh-server-openssh tools-debug eclipse-debug"
SDKIMAGE_FEATURES    = "dev-pkgs"
GCCVERSION           = "arm-9.2"
GLIBCVERSION         = "2.31%"
meta
meta-poky
meta-gplv2           = "HEAD:16634c7cfeea783d70108d31c57fb1e4250b4100"
meta-arm-toolchain   = "HEAD:1cfb0a1d773dd6eac3f387fd68ded7c907402d08"
meta-telechips-bsp   = "HEAD:3b93f65c0817ba3f16754d6c4de84792127f3b0b"
meta-qt5             = "HEAD:e3c49254dfa17fe2ebd3367a2adc7bf6016d718c"
meta-core
meta-extra
meta-qt5
```

1-5) fai파일을 fwdn/deploy-images에 이동

```
vboxuser@topst:~/autolinux/topst/build/tcc8050-main/tmp/deploy/fwdn$ ll SD_Data.fai
-rw-rw-r-- 1 vboxuser vboxuser 7607622528 12월 10 11:45 SD_Data.fai
vboxuser@topst:~/autolinux/topst/build/tcc8050-main/tmp/deploy/fwdn$
```

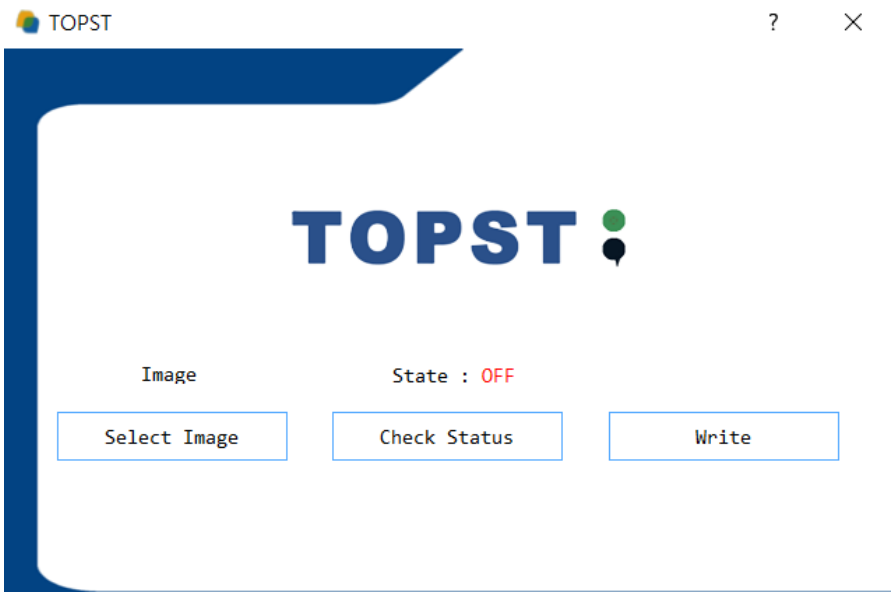


1-6) fwdn/TOPST.exe로 fai파일을 TOPST에 굽기

Select image : 아까 가져온 fai파일 선택

Check Status : fwdn 케이블 연결 후 클릭하여 연결 체크

Write : fai파일 굽기 시작



1-7) 부팅 후 dmesg | grep dwmac 명령어로 다음 메시지가 출력되면 성공

```
root@jammy:~# dmesg | grep dwmac
[    1.699853] tcc_dwmac_probe
[    1.699868] tcc-dwmac 11c00000.gmac: IRQ eth_wake_irq not found
[    1.699872] tcc-dwmac 11c00000.gmac: IRQ eth_lpi not found
[    1.699979] tcc_dwmac_init.
[    1.781829] tcc-dwmac 11c00000.gmac: Cannot get CSR clock
[    1.781846] tcc-dwmac 11c00000.gmac: PTP uses main clock
[    1.781851] tcc-dwmac 11c00000.gmac: no reset control found
[    1.781923] tcc-dwmac 11c00000.gmac: User ID: 0x85, Synopsys ID: 0x51
[    1.781927] tcc-dwmac 11c00000.gmac:         DWMAC4/5
[    1.781932] tcc-dwmac 11c00000.gmac: DMA HW capability register supported
[    1.781935] tcc-dwmac 11c00000.gmac: RX Checksum Offload Engine supported
[    1.781937] tcc-dwmac 11c00000.gmac: TX Checksum insertion supported
[    1.781940] tcc-dwmac 11c00000.gmac: Enable RX Mitigation via HW Watchdog Timer
[    1.781946] tcc-dwmac 11c00000.gmac: device MAC address 9e:69:f6:ab:6b:ac
[    3.547123] tcc-dwmac 11c00000.gmac eth0: PHY [stmmac-0:00] driver [RTL8211E Gigabit Ethernet]
[    3.569824] tcc-dwmac 11c00000.gmac eth0: No Safety Features support found
[    3.569835] tcc-dwmac 11c00000.gmac eth0: IEEE 1588-2008 Advanced Timestamp supported
[    3.569958] tcc-dwmac 11c00000.gmac eth0: registered PTP clock
[    3.569967] tcc-dwmac 11c00000.gmac eth0: configuring for phy/rgmii link mode
[    3.993859] tcc-dwmac 11c00000.gmac eth0: PHY [stmmac-0:00] driver [RTL8211E Gigabit Ethernet]
[    4.005108] tcc-dwmac 11c00000.gmac eth0: No Safety Features support found
[    4.005117] tcc-dwmac 11c00000.gmac eth0: IEEE 1588-2008 Advanced Timestamp supported
[    4.005911] tcc-dwmac 11c00000.gmac eth0: registered PTP clock
[    4.005920] tcc-dwmac 11c00000.gmac eth0: configuring for phy/rgmii link mode
[    7.074600] tcc-dwmac 11c00000.gmac eth0: Link is Up - 1Gbps/Full - flow control rx/tx
```