**RTC\_DUMMY Porting**

Step 1: Coping source code from old BSP to new BSP

**From:** azalea\_r1760/bsp/kernel/drivers/rtc/ *rtc-dummy.c*

**To:** dolphin\_plus\_bsp/bsp/kernel\_4.4/drivers/rtc

*cp azalea\_r1760/bsp/kernel/drivers/rtc/rtc-dummy.c dolphin\_plus\_bsp/bsp/kernel\_4.4/drivers/rtc*

**Step 2:** Edit Makefile to build *rtc-dummy*:

Opening the Makefile *azalea\_r1760/bsp/kernel/drivers/rtc*

Coping *obj-$(CONFIG\_RTC\_DRV\_DUMMY) += rtc-dummy.o*

* Pasting into the Makefile:

*dolphin\_plus\_bsp/bsp/kernel\_4.4/drivers/rtc /Makefile*

**Step 3:** Update Kconfig file for creating a menu configuration:

Opening Kconfig in old source location and copy contents of “**RTC\_DRV\_DUMMY**” into the Kconfig file of new BSP as following samle.

New menuconfig on Kernel 4.4.120:

*config RTC\_DRV\_DUMMY*

*tristate "Dummy RTC driver"*

*help*

*Say Y to enable Dummy RTC support on Lily systems.*



**Step 4: Check menuconfig on new BSP:**

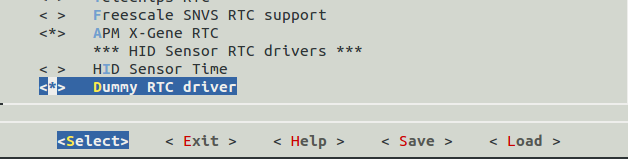
$ make ARCH=arm64 menuconfig

Selecting as path:

**Device Drivers --->**

[\*] Real Time Clock --->

And slide down cursor to **select** Dummy RTC driver and save this configuration



**Step 5:**Save the defconfig to use in the next time.

$ make ARCH=arm64 savedefconfig

Copy to /arch/arm\*/xxx\_defconfig

$ cp defconfig arch/arm64/configs/my\_defconfig

**Step 6**: Build driver module

* Navigate to source code kernel:

*$ cd dolphin\_plus\_bsp/bsp/kernel\_4.4/*

* Export arm toolchain

*$ export PATH=/opt/gcc-linaro-7.3.1-2018.05-i686\_aarch64-linux-gnu/bin:$PATH*

* Make config

*$ sudo make ARCH=*arm64 my\_defconfig

* Export Environment

$ export *ARCH=arm64 CROSS\_COMPILE=/opt/gcc-linaro-7.3.1-2018.05-i686\_aarch64-linux-gnu/bin/aarch64-linux-gnu-*

* Build*:*

*$ make M=drivers/rtc*

* **© Porting is successful.**