**JK\_RTC\_Module Porting**

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

azalea\_r1760/bsp/kernel/drivers/char/jk\_rtc

* dolphin\_plus\_bsp/bsp/kernel\_4.4/drivers/char/jk\_rtc

**Step 2:** Add build condition in Makefile:

Opening the Makefile azalea\_r1760/bsp/kernel/drivers/char/ Makefile

Coping obj-$(CONFIG\_RTC\_DRV\_FOR\_APP) += jk\_rtc/

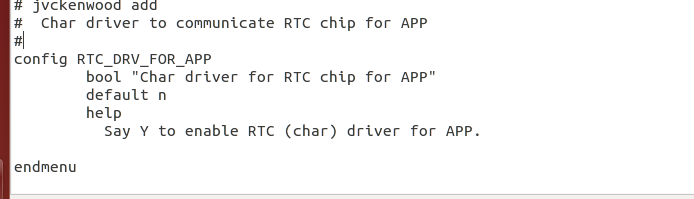
* Pasting into the Makefile:

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

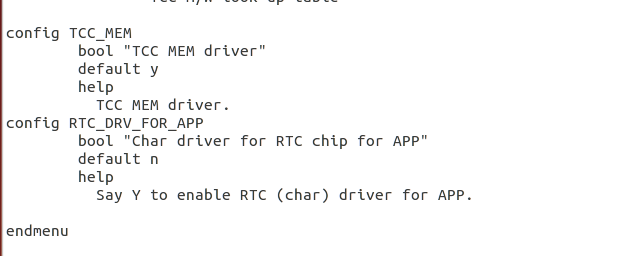
**Step 3:** Update Kconfig file for creating a menu configuration:

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

Old munuconfig on Kernel 3.18.82



New menuconfig on Kernel 4.4.120:



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

$ make ARCH=arm64 menuconfig

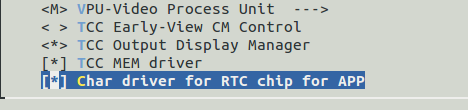
Selecting as path:

**Device Drivers --->**

Input device support --->

[\*] Char driver for RTC chip for APP

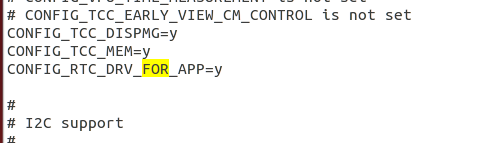
And slide down cursor to **select** Char driver for RTC chip for APP and save this configuration



**Step 5:** Check configuration have selected into .config file

**$ gedit .config**

see CONFIG\_RTC\_DRV\_FOR\_APP=y in .config file



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

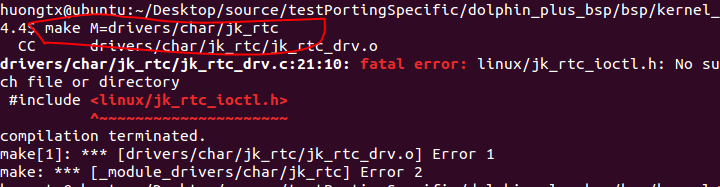
* Export Environment

$ 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/char/jk\_rtc*

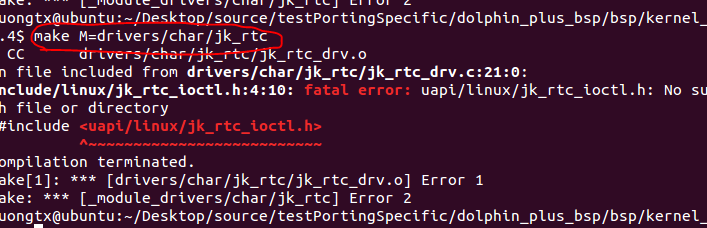
*$ make M=drivers/char/jk\_rtc*



* Thiếu file **linux/jk\_rtc\_ioctl.h**
* Copy **jk\_rtc\_ioctl.h** từ azalea

*cp azalea\_r1760/bsp/kernel/include/linux/jk\_rtc\_ioctl.h dolphin\_plus\_bsp/bsp/kernel\_4.4/include/linux/*

*Error occur* :



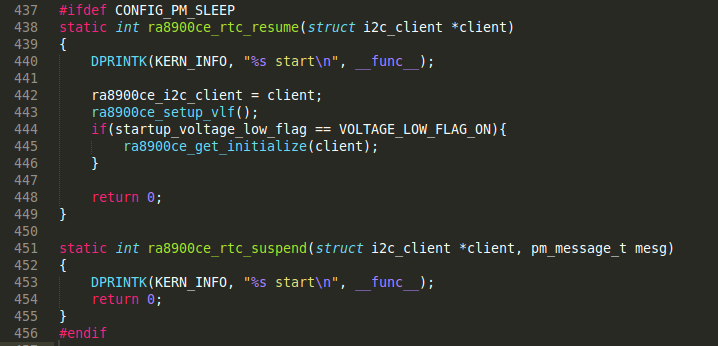
*cp azalea\_r1760/bsp/kernel/include/uapi/linux/jk\_rtc\_ioctl.h dolphin\_plus\_bsp/bsp/kernel\_4.4/include/uapi/linux/*



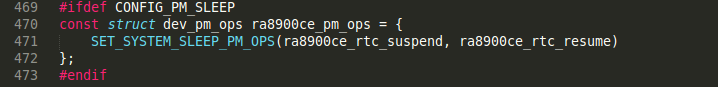
In source code kernel version 4.x, i2c\_driver do not have 2 functions resume and suspend, so fixed this error by move these functions to dev\_pm\_ops:

* Edit file *drivers/char/jk\_rtc/rtc-ra8900ce.c*

Add defined CONFIG\_PM\_SLEEP for two functions *suspend* and *resume*:



Create new struct *ra8900ce\_pm\_ops:*



#ifdef CONFIG\_PM\_SLEEP

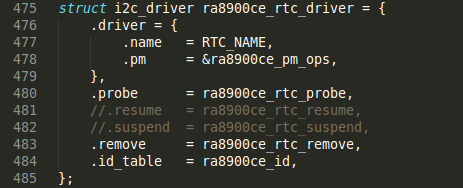
const struct dev\_pm\_ops ra8900ce\_pm\_ops= {

SET\_SYSTEM\_SLEEP\_PM\_OPS(ra8900ce\_rtc\_suspend,ra8900ce\_rtc\_resume)

};

#endif

* Add *ra8900ce\_pm\_ops* to *i2c\_driver* and comment unused functions in this struct:



struct i2c\_driver ra8900ce\_rtc\_driver = {

.driver = {

.name = RTC\_NAME,

.pm = &ra8900ce\_pm\_ops

},

.probe = ra8900ce\_rtc\_probe,

// .resume = ra8900ce\_rtc\_resume,

// .suspend = ra8900ce\_rtc\_suspend,

.remove = ra8900ce\_rtc\_remove,

.id\_table = ra8900ce\_id,

};

* Add new defined config of CONFIG\_PM\_SLEEP to *arch/arm/configs/tcc803x\_linux\_avn\_defconfig*



* Rebuild*:*

*$ make M=drivers/char/jk\_rtc*

* **© Porting is successful.**