Epson RX6110 SPI Linux Driver – January 9, 2014_ K3.8-v 1.0

The RX6110 SPI Linux device driver provides the means for an application running in user space to access the Epson RX6110 RTC. Note: The RX6110 supports both SPI and I2C interfaces, if you are configured for I2C, download the RX6110 I2C driver.

The SPI driver can be easily modified according to the user's requirements and rebuilt. The driver was tested using Linux kernel 3.8.x and was developed using the ARMhf Ubuntu 12.04LTS distribution on a BeagleBone Black. It is expected that changes and additions will be required for driver implementation on other platforms/interfaces based on the specific requirements of those platforms.

Installing the Driver

To install the driver source:

1. Copy the file rtc-rx6110-spi.c into the directory ./linux-3.8.x/drivers/rtc. "linux-3.8.x" refers to the base of the linux kernel source tree.

2. Add the following lines into the drivers/rtc/Kconfig file:

```
config RTC_DRV_RX6110_SPI
	tristate "Epson RX6110SA SPI"
	help
	  If you say yes here you get support for the Epson
	  RX6110SA SPI RTC chip.

	  This driver can also be built as a module. If so, the module
	  will be called rtc-rx6110-spi.
```

3. Add the following line to the drivers/rtc/Makefile:

```
obj-$(CONFIG_RTC_DRV_RX6110_SPI)	+= rtc-rx6110-spi.o
```

4. During the rebuild, make sure to include the 'Epson RX6110SA SPI' option in the Kernel Configuration window under 'Device Drivers' -> 'Real Time Clock'.

Hardware Considerations

The testing of the driver was done using an Epson RX6110 part mounted on a breadboard connected to the BeagleBone Black P9 header. The connections are as follows:

RX6110 pin	BeagleBone Black P9	Notes
#1 NC	--	
#2 NC	--	
#3 /IRQ2	P9_27	See Note
#4 GND	P9_01	
#5 SDA	P9_20	pull-up resistor recommended (see RX6110 spec)
#6 SCL	P9_19	pull-up resistor recommended (see RX6110 spec)
#7 /IRQ1	P9_23	See Note and pull-up resistor recommended (see RX6110spec)
#8 VDD	P9_03	

Testing Environment

The following assumes that the RX6110 driver has been modified as required, rebuilt, and included in the linux build as either a built-in driver or a module.

Also BB-BONELT-HDMI overlay needs to be disabled from loading to avoid the conflict as it uses same SPI. To do it, add the following line into boot/uEnv.txt file on SD card:

```
optargs=quiet capemgr.disable_partno=BB-BONELT-HDMI
```

To test the driver, the "RTC breadboard" was setup as a cape on the BeagleBone Black header P9. This requires adding the RX6110 RTC into the device tree using a device tree overlay (.dtbo). A sample device tree source file (.dts) and the compiled .dtbo file are included in the driver package. These files must be copied to the /lib/firmware directory on the BeagleBone Black system.

Once the system has booted, change to the /sys/devices/bone_capemgr.# directory (where # is the number). Type the following (you may need root access depending on your setup/configuration):

```
echo BBB-RX6110-SPI > slots
cat slots
```

The RX6110 cape should now be listed in one of the slots and the driver should be loaded at /dev/rtc1 (again depending on your configuration). To confirm that the driver has loaded, type:

```
dmesg | grep rx6110
```

To test the read/write capabilities of the RX6110, we can now use the hwclock command. To get the current time, type:

```
hwclock -r -f /dev/rtc1
```

If the time has not been set yet, the rtc should return with Jan 1, 1970. Next, we can set the system date with the following example:

```
date -s "Thurs Nov 07 11:33:00 PDT 2013"
```

Write the new date to the clock:

```
hwclock -w -f /dev/rtc1
```

And read it back:

```
hwclock -r -f /dev/rtc1
```

The RX6110 is now set with the new time.

The driver package also includes a small sample application (rtctest.c) that demonstrates how to read and write the time from/to the RX6110 using the ioctls RTC_RD_TIME and RTC_SET_TIME.

Note:
The driver uses one IRQ. The /IRQ1 is configured on P9_23 pin in BBB-RX6110-SPI-00A0.dts file.