RV1805 Linux driver documentation

Free Electrons

April 14, 2016

Contents

The current status of the functionalities :

Functionality	Status	Upstream
Read date/time	🟩	v4.5
Set date/time	🟩	v4.5
Alarm	🟩	v4.5
NVRAM	🟥	
OFIE	🟨	v4.6
Autocalib	🟨	v4.6

Building the driver

The driver for RTC Microcrystal RV-1805 is the same for Abracon ABX80X RTC as registers are similar.

out of tree

Simply use `make` in the driver directory.

```
$ make
make -C /lib/modules/‘uname -r‘/build M=$PWD
make[1] : Entering directory ’/usr/src/linux-headers-4.4.0-1-amd64’
  LD      /usr/src/rv1805/built-in.o
  CC [M]  /usr/src/rv1805/rtc-abx80x.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /usr/src/rv1805/rtc-abx80x.mod.o
  LD [M]  /usr/src/rv1805/rtc-abx80x.ko
make[1] : Leaving directory ’/usr/src/linux-headers-4.4.0-1-amd64’
```

If cross-compiling or if the kernel headers directory detection fails, pass `KDIR`:

```
$ make KDIR=/usr/src/linux
make -C /usr/src/linux/ M=$PWD
make[1]: Entering directory ’/usr/src/linux’
  LD      /usr/src/rv1805/built-in.o
  CC [M]  /usr/src/rv1805/rtc-abx80x.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /usr/src/rv1805/rtc-abx80x.mod.o
  LD [M]  /usr/src/rv1805/rtc-abx80x.ko
make[1]: Leaving directory ’/usr/src/linux’
```

in-tree

For Linux 4.4 :

```
$ git apply < 000*.patch
```

For Linux 4.6 :

These patches are already in this version so there is no need to apply patches.

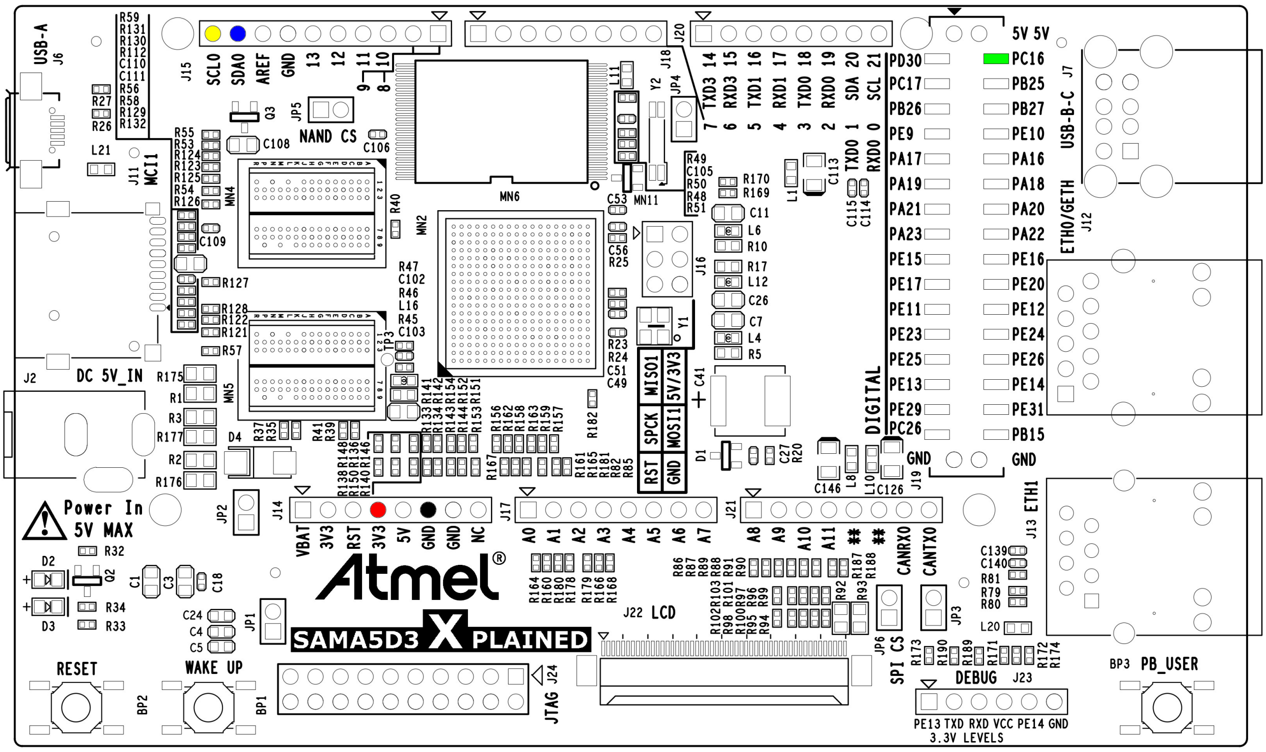
Then select `CONFIG_RTC_DRV_ABX80X` in your kernel configuration.

Connecting the RV-1805-C3 development board

Atmel Sama5d3 Xplained

The following configuration has been tested:

RV1805	SAMA5D3 Xplained	
VDD	J14 3V3	🟥
VSS	J14 GND	⬛
SDA	J15 SDA0	🟦
SCL	J15 SCL0	🟨
nINT	J19 PC16	🟩



Device Tree

In the SAMA5D3 Xplained devicetree (`arch/arm/boot/dts/at91-sama5d3_xplained.dts`), the rtc node should be added as a child of the existing i2c0 node. The end result should look like:

```
i2c0: i2c@f0014000 {
    pinctrl-0 = <&pinctrl_i2c0_pu>;
    status = "okay";

   rtc: rv1805@69 {
       compatible = "microcrystal,rv1805";
       reg = <0x69>;
       status = "okay";
       pinctrl-0 = <&pinctrl_rtc_int>;
       interrupts-extended = <&pioC 16 GPIO_ACTIVE_LOW>;
   };
};
```

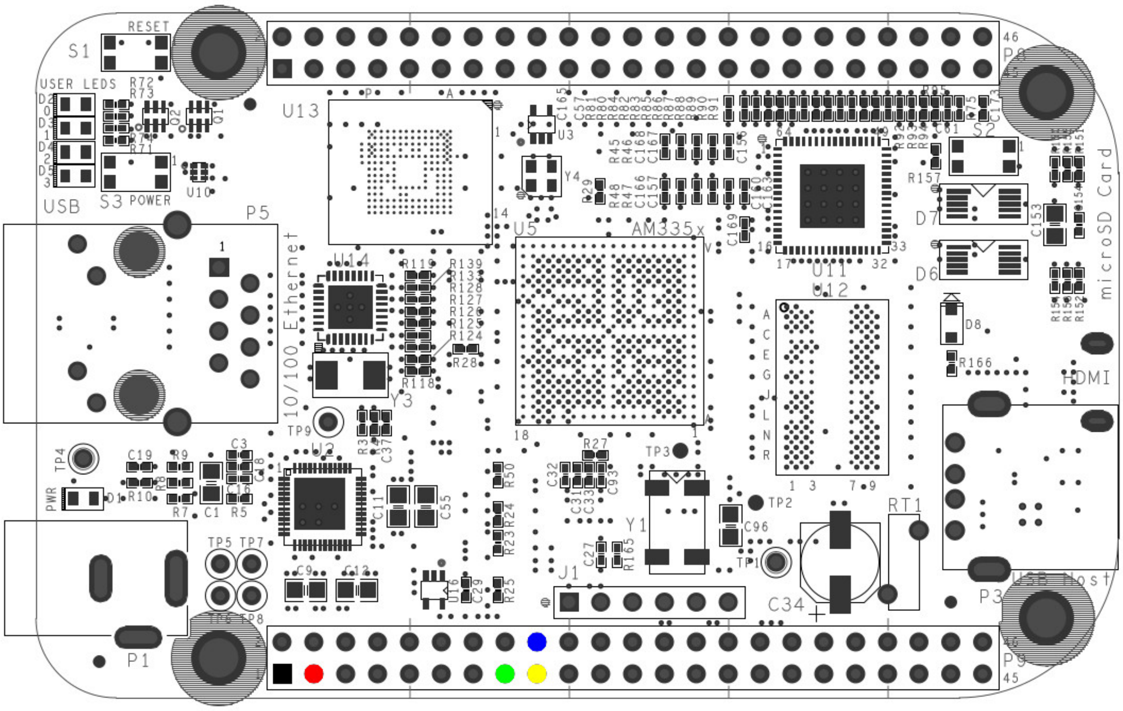
A pinctrl definition has to be added in the pinctrl node:

```
pinctrl@fffff200 {
    rtc {
        pinctrl_rtc_int: rtc_int {
            atmel,pins =
                <AT91_PIOC 16 AT91_PERIPH_GPIO AT91_PINCTRL_PULL_UP_DEGLITCH>;
        };
    };
```

Beaglebone Black

The following configuration has been tested:

RV1805	Beaglebone (P9)	
VDD	3	🟥
VSS	1	⬛
SDA	18	🟦
SCL	17	🟨
nINT	15	🟩



Device Tree

In the Beaglebone Black device tree (`arch/arm/boot/dts/am335x-boneblack.dts`), i2c1 should be enabled and the rtc node added as a child. The end result should look like:

```
&i2c1 {
    status = "okay";
    clock-frequency = <100000>;

    pinctrl-names = "default";
    pinctrl-0 = <&i2c1_pins>;

    rv1805: rtc@69 {
        compatible = "microcrystal,rv1805";
        reg = <0x69>;
        status = "okay";
        pinctrl-0 = <&rtc_nint_pins>;
        interrupts-extended = <&gpio1 16 GPIO_ACTIVE_LOW>;
    };
};
```

Pinctrl definitions have to be added in the pinctrl node:

```
&am33xx_pinmux {
    rtc_nint_pins: pinmux_rtc_nint_pins {
        pinctrl-single,pins = <
            0x040 (PIN_INPUT | MUX_MODE7)
        >;
    };
    /* Pins 17 (I2C1_SCL) and 18 (I2C1_SDA) of connector P9 */
    i2c1_pins: pinmux_i2c1_pins {
        pinctrl-single,pins = <
            0x158 (PIN_INPUT_PULLUP | MUX_MODE2)    /* spi0_d1.i2c1_sda */
            0x15c (PIN_INPUT_PULLUP | MUX_MODE2)    /* spi0_cs0.i2c1_scl */
        >;
    };
};
```

Using the RTC

Loading the driver

If the driver has been built statically in the kernel, nothing has to be done. If it is built as a module, then load it with:

```
$ modprobe rtc-abx80x
```

or if that is not working:

```
$ insmod /path/to/rtc-abx80x.ko
```

Finding the RTC device

On initialization, the driver will log the following message:

```
$ dmesg | grep abx80x
rtc-abx80x 1-0069: model 1805, revision 2.3, lot c, wafer 7, uid 5c7d
rtc-abx80x 1-0069: rtc core: registered abx8xx as rtc0
```

This means that `rtc0` is the RV1805 rtc.

Reading the date

```
$ hwclock -r -f /dev/rtc0
```

You'll get the following message when the date is invalid:

```
hwclock: RTC_RD_TIME: Invalid argument
```

You will also see the following message in your kernel log:

```
$ dmesg | grep abx80x
rtc-abx80x 1-0069: Oscillator failure, data is invalid.
```

Setting the date

Set the system date:

```
$ date -s "2016-04-11 09:30"
```

Write the date to the RTC:

```
$ hwclock -w -f /dev/rtc0
```

Testing functionalities

Oscillator Failure

To test the Oscillator Failure, disconnect the VDD pin. Plug it back and try to read the date, it should return an invalid data.

```
$ hwclock -r -f /dev/rtc0
hwclock: RTC_RD_TIME: Invalid argument
```

Updating the date removes the current error.

Alarm

To test the alarm, the `rtctest` tool is used. To compile it :

```
$ cd /path/to/kernel/source
$ arm-linux-gnueabi-gcc -s tools/testing/selftests/timers/rtctest.c
     -o rtctest
```

Copy the binary into the board and execute it. The test must be completed to validate the alarm with the final output :

```
*** Test complete ***
```

Autocalibration

The Autocalibration can be configured using sysfs entries :

- "oscillator" to select the oscillator type (XT or RC).
- "autocalibration" to select the autocalibration type.

For the oscillator, the values are "xtal" for XT oscillator and "rc" for RC oscillator. For the autocalibration, the value 0 disables the autocalibration cycle, 512 is for a 512 seconds autocalibration cycle and 1024 for a cycle of 1024 seconds.

```
Set to XT Oscillator
$ echo xtal > /sys/class/rtc/rtc0/device/oscillator
Activate an autocalibration every 512 seconds
$ echo 512 > /sys/class/rtc/rtc0/device/autocalibration
Read current values
$ cat /sys/class/rtc/rtc0/device/oscillator
$ cat /sys/class/rtc/rtc0/device/autocalibration
```