## Real Time Clock (RTC)

CPU can accesses internal RTC registers and memories via **I2C #4** serial link. I2C address for accessing internal registers and SRAM memory is 6Fh while EEPROM access I2C address is 57h.

RTC alarm output is connected to CPU GPIO (GPIO1-18) and can be configured to interrupt CPU on specific time and date.

RTC battery back-up expected life time is 5 years. SW should provide user alarm for battery maintenance operation required.

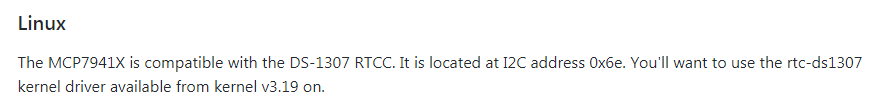
RTC has two I2C address spaces:

* I2C address = 57H: Read/Write into internal RTC EEPROM memory block
* I2C address = 6FH: Read/Write into internal RTC registers and SRAM.

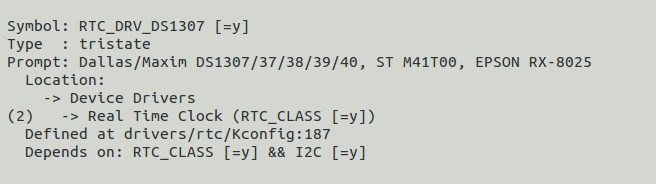
<http://www.microchip.com/wwwproducts/en/MCP79410>



Kernel config:

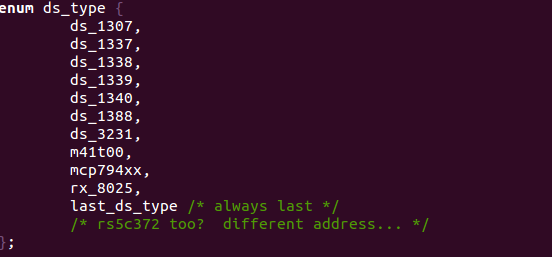


The RTC\_DRV\_DS1307 is enabled by default in the kernel

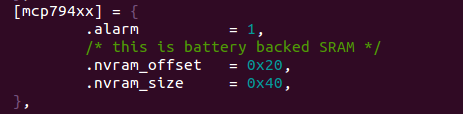


In the kernel driver:

Bsp/linux/drivers/rtc/rtc-ds1307.c



We can see that it is exist there:



**Device tree:**

In the i2c\_device\_id

static const struct i2c\_device\_id ds1307\_id[] = {

{ "ds1307", ds\_1307 },

{ "ds1337", ds\_1337 },

{ "ds1338", ds\_1338 },

{ "ds1339", ds\_1339 },

{ "ds1388", ds\_1388 },

{ "ds1340", ds\_1340 },

{ "ds3231", ds\_3231 },

{ "m41t00", m41t00 },

{ "mcp7940x", mcp794xx },

{ "mcp7941x", mcp794xx },

{ "pt7c4338", ds\_1307 },

{ "rx8025", rx\_8025 },

{ }

};

We have ID of: mcp7941x which is what we need.



The device have two i2c slave address in I2C4

&i2c4 {

clock-frequency = <100000>;

pinctrl-names = "default";

pinctrl-0 = <&pinctrl\_i2c4>;

clocks = <&clks 116>;

status = "okay";

eeprom: eeprom@52 {

compatible = "at,24c512";

reg = <0x52>;

};

mcp7941x@6f {

compatible = "microchip,mcp7941x";

reg = <0x6f>;

};

mcp7941x@57 {

compatible = "microchip,mcp7941x";

reg = <0x57>;

};

};

<https://raspberrypi.stackexchange.com/questions/61651/how-to-setup-a-mcp79410-real-time-clock>

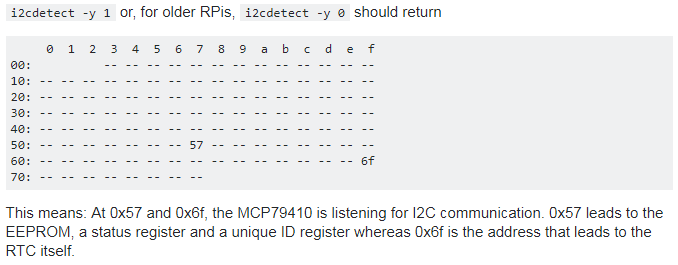
I2c-tools

We need to install the tool ( sudo apt-get install i2c-tools)

When the board will arrive, search the devices on the i2c 4 bus:

sudo i2cdetect -y 3

we should see the following:



The question is how the rtc-mcp7941x driver will be loaded?

Does only when the device is physically there? Or we need to modprobe it when it will be there.

When the device will be there we can use

https://www.cyberciti.biz/faq/howto-set-date-time-from-linux-command-prompt/

to set the date:

date -s "2 APR 2018 18:00:00"

write to the rtc

hwclock –w

Pull up the power and try to read it back.

From now on, the RTC will keep the time and resynchronize the RPi's system time automatically on startup or manually by entering

hwclock -r

Reading and writing

We can

Write an I2C c code or

<http://www.wiki.xilinx.com/Linux+I2C+Aardvark>

use the /sys/bus/i2c/devices/

We can also use i2cdump , i2cset , i2cget from the i2c tools

i2cdump -y -f 0x3 0x57

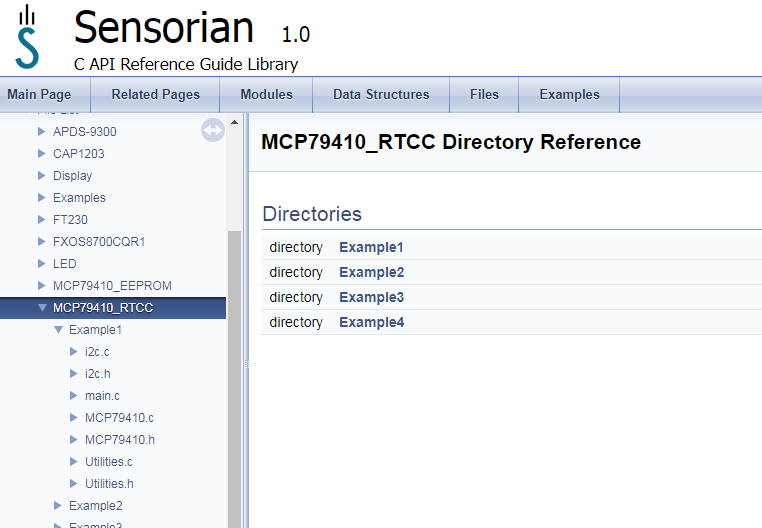
if we need other functionality that does not exists

we can port Arduino code:

<https://github.com/ichilton/mcp7941x_arduino/blob/master/MCP7941x.cpp>

<https://sensorian.github.io/c_docs/_example1_2_m_c_p79410_8c_source.html>

A full source code and examples:



<https://sensorian.github.io/c_docs/dir_33e4b38ea5e122bb4f74ed85e5eaad71.html>