


RTC DS1307

Aus GNUBLIN

<i>Schwierigkeitsgrad</i>	<i>Voraussetzung</i>	<i>Gnublin Familie</i>
 Anfänger	Grundkenntnisse Konsole	Alle

Inhaltsverzeichnis

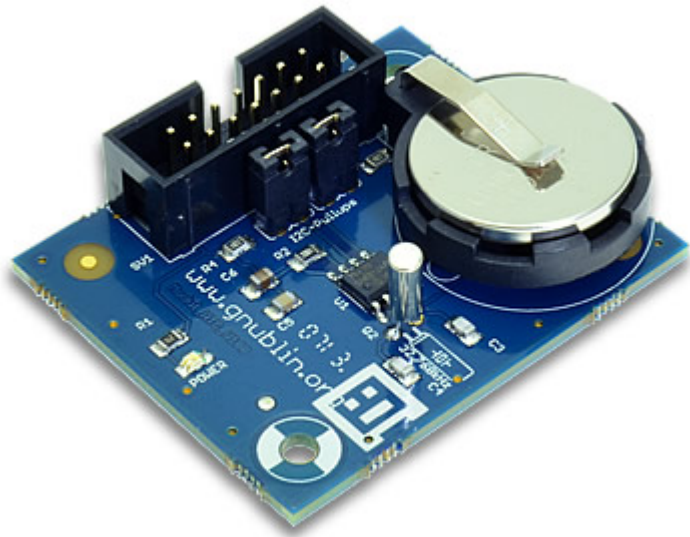
- 1 Quickstart
- 2 Für Anfänger
 - 2.1 RTC mit RV3029
 - 2.1.1 Systemzeit bei Systemstart stellen
 - 2.2 RTC mit MCP79410
 - 2.2.1 Systemzeit bei Systemstart stellen
- 3 Für Fortgeschrittene
 - 3.1 RTC mit DS1307
 - 3.1.1 Schaltplan
 - 3.1.2 Kernel-Anpassungen
 - 3.1.3 Linux Anpassungen
 - 3.2 RTC mit MCP79410
 - 3.2.1 Kernel-Anpassungen
 - 3.2.1.1 Kernel-Konfiguration
 - 3.2.1.2 Treiber patchen
 - 3.2.2 Linux-Anpassungen
- 4 Problembehebung inapropriate ioctl()
- 5 Problembehebung RTC_READ_TIME invalid argument
- 6 Uhrzeit temporär manuell einstellen

Quickstart

Um in einem Linux System sich die Uhrzeit zu merken, obwohl das System vom Strom getrennt war, benötigt man eine Echtzeit Uhr, die mit einer externen Batterie gepuffert ist.

Als Standard für die Gnublin Distribution verwenden wir: MCP79410 (<http://ww1.microchip.com/downloads/en/DeviceDoc/22266D.pdf>)

Den Baustein gibt es als fertiges Modul.



Für das Modul gibt es einen kurzen Befehl:

```
gnublin-rtc -s "2013/01/20 11:23:12"
```

Möchte man die Uhrzeit auslesen macht man dies so:

```
gnublin-rtc -g
```

Und will man, dass die Uhrzeit aus RTC in die Linux Uhr übernommen wird geht das wie folgt:

```
gnublin-rtc -x
```

So einfach hat man die Uhrzeit unter Kontrolle.

Für Anfänger

Dieser Abschnitt beschreibt wie man einen RTC IC (hier DS1307 (<http://datasheets.maximintegrated.com/en/ds/DS1307.pdf>) oder MCP79410 (<http://ww1.microchip.com/downloads/en/DeviceDoc/22266D.pdf>) oder RV-3029 (<http://www.microcrystal.com/CMSPages/GetFile.aspx?nodeguid=c85958d8-94ca-4325-87c5-e65a30ad4bd5>)) korrekt anschliesst und Linux so konfiguriert dass die RTC via *hwclock* angesprochen werden kann.

Hinweis: Ohne Gnublin Distribution sollte auf alle Fälle der an Gnublin angepasste I²C-Treiber aus dem Gnublin-Forum (<http://forum.gnublin.org/viewtopic.php?f=4&t=6>) verwendet werden (aktuell ist Version 5)! Mit dem originalen Treiber ließ sich der MCP79410 nicht ansprechen.

RTC mit RV3029

In der aktuellen Version der gnublin_defconfig und der Gnublin Distribution ist der Treiber bereits als Modul eingebaut.

In der menuconfig befindet sich der Treiber unter "Device Drivers -> Real Time Clock -> Micro Crystal RV-3029".

Nachdem man den Treiber aktiviert hat und die Module und das zImage kompiliert und auf die SD-Karte gebracht hat geht man wie folgt vor um die RTC anzusteuern.

Zuerst wird das Modul geladen (falls der Treiber nicht als builtin konfiguriert ist):

```
modprobe rtc-rv3029
```

Anschließend wird ein neues I2C Device mit folgendem Befehl erstellt und mit dem Treiber verknüpft:

```
echo rv3029 0x56 > /sys/bus/i2c/devices/i2c-1/new_device
```

Nun kann man die Zeit mit folgendem Befehl auslesen:

```
hwclock -r
```

Möchte man erweiterte Debug-Ausgaben so hängt man einfach ein -D an den Befehl an.

Falls bei diesem Schritt folgender Fehler auftritt:

```
RTC_RD_TIME: Inappropriate ioctl for device
ioctl() to /dev/rtc to read the time failed.
```

so sollte man sich den Abschnitt Problembehebung inapropriate ioctl() durchlesen und mithilfe der Anleitung in der Lage sein den Fehler zu beseitigen.

Die Zeit der RTC kann man mit folgendem Befehl setzen:

```
hwclock --set --date "2012/11/12 11:11:11"
```

Möchte man hier auch erweiterte Debug-Informationen haben, so muss man wieder ein -D an den Befehl anhängen.

Anschließend kann man die Systemzeit mit folgendem Befehl auf die Zeit der Hardwareclock stellen:

```
hwclock --hctosys
```

Systemzeit bei Systemstart stellen

Möchte man die Systemzeit bei jedem Start nach der RTC stellen, so sind ein paar kleine Veränderungen im System nötig. Zuerst wird der folgende Text in die Datei **/etc/rc.local** vor die Zeile *exit 0* hinzugefügt.

```
echo mcp7940 0x6f > /sys/bus/i2c/devices/i2c-1/new_device
echo "Now setting the date and time."
sleep 1
hwclock --hctosys
```

Anschließend wird ein Eintrag in der Datei `/etc/modules` hinzugefügt:

```
rtc-mcp7940
```

Zuletzt wird noch das `hwclock` Shellsript deaktiviert, da es mit dieser RTC nur zu Komplikationen führt.

```
update-rc.d hwclock remove && update-rc.d hwclock.sh remove
```

Nun dürfte das Board ohne weiteres bei jedem Start die Systemzeit nach der Hardwareclock stellen.

RTC mit MCP79410

In der aktuellen Version der `gnublin_defconfig` und der Gnublin Distribution ist der Treiber bereits als Modul eingebaut.

In der `menuconfig` befindet sich der Treiber unter "Device Drivers -> Real Time Clock -> Microchip 7940 (MCP7940)".

Nachdem man den Treiber aktiviert hat und die Module und das `zImage` kompiliert und auf die SD-Karte gebracht hat geht man wie folgt vor um die RTC anzusteuern.

Zuerst wird das Modul geladen (falls der Treiber nicht als `builtin` konfiguriert ist):

```
modprobe rtc-mcp7940
```

Anschließend wird ein neues I2C Device mit folgendem Befehl erstellt und mit dem Treiber verknüpft:

```
echo mcp7940 0x6f > /sys/bus/i2c/devices/i2c-1/new_device
```

Nun kann man die Zeit mit folgendem Befehl auslesen:

```
hwclock -r
```

Möchte man erweiterte Debug-Ausgaben so hängt man einfach ein `-D` an den Befehl an.

Falls bei diesem Schritt folgender Fehler auftritt:

```
Cannot access the Hardware Clock via any known method. Use the --debug option to see the details of our search for an access method. root@gnublin:~# hwclock -r -D hwclock from util-linux-ng 2.17.2 hwclock: Open of /dev/rtc failed, errno=6: No such device or address. No usable clock interface found. Cannot access the Hardware Clock via any known method.
```

so sollte man sich den Abschnitt Problembehebung inapropriate ioctl() durchlesen und mithilfe der Anleitung in der Lage sein den Fehler zu beseitigen.

Die Zeit der RTC kann man mit folgendem Befehl setzen:

```
hwclock --set --date "2012/11/13 11:11:11"
```

Möchte man hier auch erweiterte Debug-Informationen haben, so muss man wieder ein

-D an den Befehl anhängen.

Anschließend kann man die Systemzeit mit folgendem Befehl auf die Zeit der Hardwareclock stellen:

```
hwclock --hctosys
```

Systemzeit bei Systemstart stellen

Möchte man die Systemzeit bei jedem Start nach der RTC stellen, so sind ein paar kleine Veränderungen im System nötig. Zuerst wird der folgende Text in die Datei `/etc/rc.local` vor die Zeile `exit 0` hinzugefügt.

```
echo mcp7940 0x6f > /sys/bus/i2c/devices/i2c-1/new_device
echo "Now setting the date and time."
sleep 1
hwclock --hctosys
```

Anschließend wird ein Eintrag in der Datei `/etc/modules` hinzugefügt:

```
rtc-mcp7940
```

Zuletzt wird noch das `hwclock` Shellscript deaktiviert, da es mit dieser RTC nur zu Komplikationen führt.

```
update-rc.d hwclock remove && update-rc.d hwclock.sh remove
```

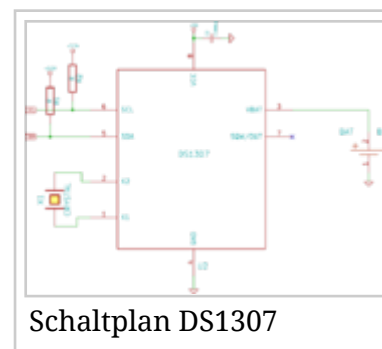
Nun dürfte das Board ohne weiteres bei jedem Start die Systemzeit nach der Hardwareclock stellen.

Für Fortgeschrittene

RTC mit DS1307

Schaltplan

Legende	Wert	Beschreibung
R1	1.2k	i2c Pull-Up SCL nach 3.3V
R2	1.2k	i2c Pull-Up SDA nach 3.3V
C2	100nF	Entstör-Kondensator
X1	32.768 Hz	RTC Quarz
B1 (optional)	3V Knopfzelle	Batterie für die RTC



Schaltplan DS1307

Wichtig: Der DS1307 ist ein 5V IC und braucht deshalb auch eine entsprechende Versorgung. Die I2C Ports des LPC arbeiten aber nur mit 3.3V. Dies ist kein Problem wenn man die I2C Pull-Up Widerstände ebenfalls an die 3.3V Versorgung hängt.

Kernel-Anpassungen

Folgende Kernel Optionen müssen aktiviert werden damit der DS1307 (als RTC) funktioniert:

Device Drivers:

- Realtime Clock (RTC_CLASS)
 - /sys/class/rtc/rtcN (sysfs) (RTC_INTF_SYSFS) *optional*
 - /proc/driver/rtc (procfs for rtc0) (RTC_INTF_PROC) *optional*
 - /dev/rtcN (character devices) (RTC_INTF_DEV)
 - Dallas/Maxim DS1307/37/38/39/40, ST M41T00, EPSON RX-8025 (RTC_DRV_DS1307)

Ich habe das ganze als Module compiliert, ich gehe aber davon aus dass statisch ebenfalls funktioniert

Linux Anpassungen

Dieser Teil dauerte tatsächlich am längsten. "Schuld" daran ist das *komische* i2c Interface.

Das Modul rtc_ds1307 erkennt die RTC nur wenn der Chip korrekt in den I2C Bus eingetragen wird. Dh.

1. dem bus die adresse und den typ des devices mitteilen
2. modul laden
3. device file anlegen

```
echo ds1307 0x68 > /sys/bus/i2c/devices/i2c-1/new_device modprobe rtc_ds1307 2>/dev/null 1>&2 mknod /dev/rtc0 c 253 0 2>/dev/null 1>&2
```

Das ganze sollte dann im dmesg/lsmmod wie folgt aussehen:

```
root@gnublin:/var/volatile/log# dmesg [ 6152.980000] device: 'rtc0': device_add [ 6152.980000] PM: Adding info for No Bus:rtc0 [ 6152.990000] rtc-ds1307 1-0068: rtc core: registered ds1307 as rtc0 [ 6153.000000] rtc-ds1307 1-0068: 56 bytes nvram [ 6153.000000] driver: '1-0068': driver_bound: bound to device 'rtc-ds1307' [ 6153.000000] bus: 'i2c': really_probe: bound device 1-0068 to driver rtc-ds1307 07 root@gnublin:/var/volatile/log# lsmod Module Size Used by rtc_ds1307 6464 0 rtc_core 11339 1 rtc_ds1307 pegasus 16772 0 mii 3416 1 pegasus
```

Damit sollte die RTC jetzt korrekt eingerichtet sein.

Um die RTC bei jedem Boot korrekt einzurichten habe ich folgende Zeilen im /etc/init.d /hwclock.sh hinzugefügt:

```
-----  
case "$1" in  
    start)  
#ab hier ists neu  
        if [ ! -d /sys/bus/i2c/devices/1-0068/ ]  
        then  
            echo ds1307 0x68 > /sys/bus/i2c/devices/i2c-1/new_device  
        fi  
        modprobe rtc_ds1307 2>/dev/null 1>&2  
        mknod /dev/rtc0 c 253 0 2>/dev/null 1>&2  
    *)  
    ;;  
esac  
-----
```

```
#bis hier
        if [ "$VERBOSE" != no ]
```

RTC mit MCP79410

Eine weitere Möglichkeit ist die Nutzung einer auf dem MCP79410 (<http://ww1.microchip.com/downloads/en/DeviceDoc/22266D.pdf>) basierenden RTC. Im Unterschied zum DS1307 arbeitet der Chip mit 3,3 V und kann somit ohne Pull-Up-Widerstände angeschlossen werden.

Es sind auch fertig bestückte Module in deutschen Webshops erhältlich (z. B. das PmodRTCC von Digilent (<http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,401,962&Prod=PMOD-RTCC>), die man mit den beigelegten Kabeln direkt an das Gnublin-Board anstecken kann.

Kernel-Anpassungen

Kernel-Konfiguration

Zum Ansteuern des MCP7941x kann der Treiber für den DS1307 verwendet werden. Zunächst müssen also die selben Einstellungen in der Kernel-Konfiguration vorgenommen werden, die bereits oben beschrieben wurden:

Device Drivers:

- Realtime Clock (RTC_CLASS)
 - /sys/class/rtc/rtcN (sysfs) (RTC_INTF_SYSFS) *optional*
 - /proc/driver/rtc (procfs for rtc0) (RTC_INTF_PROC) *optional*
 - /dev/rtcN (character devices) (RTC_INTF_DEV)
 - Dallas/Maxim DS1307/37/38/39/40, ST M41T00, EPSON RX-8025 (RTC_DRV_DS1307)

Treiber patchen

In der Kernel-Version 2.6.33 unterstützt der Treiber für den DS1307 die MCP7941x-Chipfamilie noch nicht. Es gibt einen Patch (<http://git.kernel.org/?p=linux/kernel/git/torvalds/linux.git;a=commitdiff;h=43fcb81550f7a16be192b19c77a379c9b27b1585>), der allerdings nicht 100%ig auf die von Gnublin verwendete Version passt. Die Unterschiede sind jedoch marginal. Der Patch konnte leicht an den Gnublin-Kernel angepasst werden und folgt hier:

```
----- a/drivers/rtc/rtc-ds1307.c 2012-08-31 07:51:50.000000000 +0200
+++ b/drivers/rtc/rtc-ds1307.c 2012-10-11 21:43:42.512435945 +0200
@@ -34,6 +34,7 @@
     ds_1388,
     ds_3231,
     m41t00,
+    mcp7941x,
     rx_8025,
     // rs5c372 too? different address...
```

```
};
@@ -43,6 +44,7 @@
#define DS1307_REG_SECS          0x00    /* 00-59 */
#define DS1307_BIT_CH           0x80
#define DS1340_BIT_nEOSC        0x80
+define MCP7941X_BIT_ST         0x80
#define DS1307_REG_MIN           0x01    /* 00-59 */
#define DS1307_REG_HOUR         0x02    /* 00-23, or 1-12{am,pm} */
#define DS1307_BIT_12HR        0x40    /* in REG_HOUR */
@@ -54,6 +56,7 @@
#define DS1307_REG_MONTH        0x05    /* 01-12 */
#define DS1337_BIT_CENTURY      0x80    /* in REG_MONTH */
#define DS1307_REG_YEAR         0x06    /* 00-99 */
+define MCP7941X_BIT_VBATEN     0x08

/* Other registers (control, status, alarms, trickle charge, NVRAM, etc)
 * start at 7, and they differ a LOT. Only control and status matter for
@@ -137,6 +140,8 @@
},
[m41t00] = {
},
+{mcp7941x} = {
+},
[rx_8025] = {
}, };

@@ -149,6 +154,7 @@
{ "ds1340", ds_1340 },
{ "ds3231", ds_3231 },
{ "m41t00", m41t00 },
+{ "mcp7941x", mcp7941x },
{ "rx8025", rx_8025 },
{ }
};
@@ -364,6 +370,10 @@
    buf[DS1307_REG_HOUR] |= DS1340_BIT_CENTURY_EN
                        | DS1340_BIT_CENTURY;

    break;
+ case mcp7941x:
+     buf[DS1307_REG_SECS] |= MCP7941X_BIT_ST;
+     buf[DS1307_REG_WDAY] |= MCP7941X_BIT_VBATEN;
+     break;
    default:
        break;
}
@@ -829,6 +839,22 @@
        dev_warn(&client->dev, "SET TIME!\n");
    }
    break;
+ case mcp7941x:
+     /* make sure that the backup battery is enabled */
+     if (!(ds1307->regs[DS1307_REG_WDAY] & MCP7941X_BIT_VBATEN)) {
+         i2c_smbus_write_byte_data(client, DS1307_REG_WDAY,
+             ds1307->regs[DS1307_REG_WDAY]
+             | MCP7941X_BIT_VBATEN);
+     }
+
+     /* clock halted? turn it on, so clock can tick. */
+     if (!(tmp & MCP7941X_BIT_ST)) {
+         i2c_smbus_write_byte_data(client, DS1307_REG_SECS,
+             MCP7941X_BIT_ST);
+         dev_warn(&client->dev, "SET TIME!\n");
+         goto read_rtc;
+     }
+     break;
    case rx_8025:
    case ds_1337:
    case ds_1339:
```

Der Code muss in einen Texteditor kopiert und als Datei (z. B. *rtc-ds1307_support_MCP7941x.patch*) im Hauptverzeichnis der Kernel-Quellen

abgespeichert werden. Anschließend kann man ihn mittels

```
user@home:/home/user/kernel# patch -p1 < rtc-ds1307_support_MCP7941x.patch
```

anwenden. Anschließend muss der Kernel kompiliert und (gemeinsam mit den Modulen) auf die SD-Karte übertragen werden. Nähere Infos dazu sind im Wiki unter Kernel kompilieren + Module installieren zu finden.

Linux-Anpassungen

Von nun an funktioniert alles fast identisch wie bereits für den DS1307 beschrieben. Es gibt nur zwei kleine Unterschiede:

- der Chip muss als *mcp7941x* (nicht *ds1307*) beim I²C-Bus registriert werden
- Die Adresse zur Kommunikation mit dem I²C-Bus muss dem Datenblatt zum RTC-Modul entnommen werden. Beim hier verwendeten Modul lautet sie *0x6f* (nicht *0x68*)

Insofern erfolgt die Registrierung mittels

```
root@gnublin:~# echo mcp7941x 0x6f > /sys/bus/i2c/devices/i2c-1/new_device
```

danach kann der Treiber dann wie oben beschrieben geladen werden

```
root@gnublin:~# modprobe rtc_ds1307
```

und gegebenenfalls muss noch das Device-File mit dem Befehl

```
root@gnublin:~# mknod /dev/rtc0 c 253 0
```

angelegt werden (i. d. R. nur einmal). Danach kann mittels *hwclock* auf die Uhr zugegriffen werden. Die Datei */etc/initd./hwclock.sh* habe ich dann wie folgt angepasst, damit die Uhrzeit nach jedem Booten wieder gesetzt wird:

```
-----  
case "$1" in  
    start)  
#ab hier ists neu  
        if [ ! -d /sys/bus/i2c/devices/1-006f/ ]; then  
            echo mcp7941x 0x6f > /sys/bus/i2c/devices/i2c-1/new_device  
        fi  
        modprobe rtc_ds1307 2>/dev/null 1>&2  
        if [ ! -c /dev/rtc0 ]; then  
            mknod /dev/rtc0 c 253 0 2>/dev/null 1>&2  
        fi  
#bis hier  
        if [ "$VERBOSE" != no ]  
-----
```

Problembhebung inapropriate ioctl()

Falls bei Aufruf von *hwclock* eine Fehlermeldung erscheint die **ioctl() inappropriate ...** enthält so stimmt etwas mit dem Devicefile */dev/rtc0* nicht. Zur Behebung des Problems muss man folgende Schritte durchgehen.

Überprüfen ob Devicefiles vorhanden sind:

```
ls -l /dev | grep rtc
```

Als Ausgabe sollte /dev/rtc0 und eventuell /dev/rtc erscheinen. Falls /dev/rtc erscheint muss man dieses Devicefile löschen, da hwclock sonst dieses File als default ansieht und unsere RTC auf /dev/rtc0 "liegt". Das Devicefile kann man einfach mit folgendem Befehl löschen:

```
rm /dev/rtc
```

Anschließend überprüfen wir ob die Major-Nummern der RTC und des Device-Files zusammenpassen. Hierfür geben wir folgenden Befehl ein:

```
cat /proc/devices | grep rtc
```

Als Ausgabe erscheint die Major-Nummer und rtc. Nun überprüfen wir die Major-Nummer von /dev/rtc0 indem wir wieder den folgenden Befehl eintippen:

```
ls -l /dev | grep rtc
```

Nach dem Benutzernamen und der Gruppe steht die Majornummer. Falls diese Zahl die selbe Zahl wie die aus dem cat Befehl ist so stimmt alles. Falls nicht muss man das Device-File löschen und ein Neues mit der richtigen Nummer eintragen.

```
rm /dev/rtc0  
mknod /dev/rtc0 c <MAJOR> 0
```

In obigem Befehl einfach MAJOR mit der Nummer ersetzen, die der Befehl "cat /proc/devices | grep rtc" ausgibt. Anschließend müsste hwclock die RTC lesen und beschreiben können.

Problembehebung RTC_READ_TIME invalid argument

Sollte man folgenden Fehler beim lesen/setzen der Uhrzeit haben,

```
RTC_RD_TIME: Invalid argument  
ioctl() to /dev/rtc0 to read the time failed.
```

so muss man einfach den nachfolgenden Befehl ausführen:

```
hwclock --systohc -D --noadjfile --utc
```

Uhrzeit temporär manuell einstellen

Hat man keine RTC und kein Netzwerk, braucht man aber eine Zeit im Linux System (z.B. wenn man das Passwort eines Benutzers einstellen möchte etc.) kann man dies schnell über die Kommandozeile machen:

```
date 072407032012
```

Nach einem weiteren Aufruf von

```
date
```

sieht man die neue Uhrzeit.

Von „http://wiki.gnublin.org/index.php?title=RTC_DS1307&oldid=6155“

Kategorie: Hardware

Diese Seite wurde zuletzt am 17. Mai 2013 um 18:06 Uhr geändert. - Diese Seite wurde bisher 12.194-mal abgerufen. - Impressum - Über GNUBLIN