

LK-RTC

Version vom 22. Februar 2017, 15:51 Uhr ([Quelltext anzeigen](#))

[Linkerkit-admin](#) ([Diskussion](#) | [Beiträge](#))

(→Raspberry Installation)

← [Zum vorherigen Versionsunterschied](#)

Zeile 171:

```
<pre class="code">echo ds1307 0x68 > /sys/class
/i2c-adapter/i2c-1/new_device
hwclock -w</pre>
```

–

Mit folgendem Befehl in der Konsole kann die aktuelle Uhrzeit ausgelesen werden:

–

```
<pre class="code">hwclock -r</pre>
```

==Downloads==

Aktuelle Version vom 7. August 2019, 15:11 Uhr ([Quelltext anzeigen](#))

[Linkerkit-admin](#) ([Diskussion](#) | [Beiträge](#))

(→Keine Zusammenfassung)

Zeile 171:

```
<pre class="code">echo ds1307 0x68 > /sys/class
/i2c-adapter/i2c-1/new_device
hwclock -w</pre>
```

Mit folgendem Befehl in der Konsole kann die aktuelle Uhrzeit ausgelesen werden:

+

```
<pre class="code">sudo hwclock -r</pre>
```

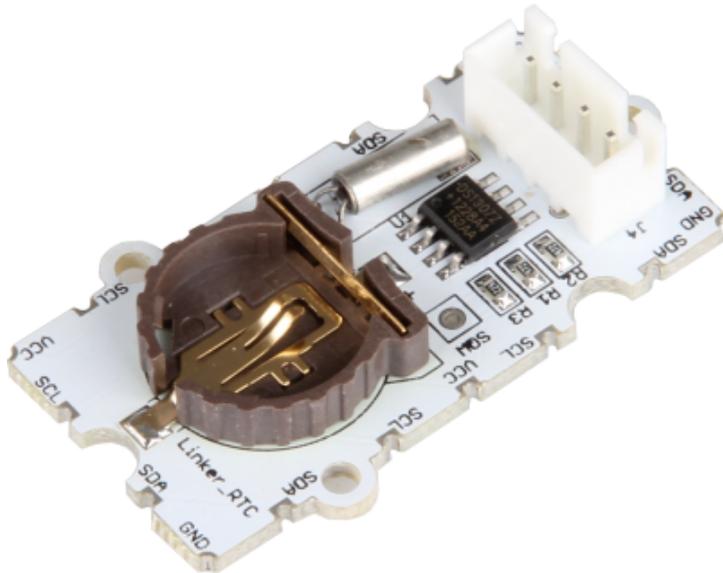
==Downloads==

Aktuelle Version vom 7. August 2019, 15:11 Uhr

Inhaltsverzeichnis

1 Bild	2
2 Technische Daten / Kurzbeschreibung	2
2.1 Spezifikation	2
3 Codebeispiel Arduino	2
4 Raspberry Installation	4
5 Downloads	5

Bild



Technische Daten / Kurzbeschreibung

Linker Kit Platine mit RTC (Echtzeituhr), ohne Batterie. Das RTC Modul basiert auf dem Uhr-Chip DS1307, welches das I2C Protokoll unterstützt. Für eine genaue Anzeige benötigt man eine Lithium Zellen Batterie (CR1225). Die Uhr liefert Sekunden, Minuten, Stunden, Tage, Monat und Jahr. Das Modul liefert das richtige Datum trotz Monate die weniger als 31 Tage haben, zusätzlich werden Schaltjahre berücksichtigt. Die Uhr arbeitet im 24 Stunden oder im 12 Stunden Format mit einer AM/PM Anzeige. Und sie ist gültig bis 2100. Wenn das Modul nur mit der primären Stromversorgung versorgt wird, funktioniert das Modul nicht richtig, weil der Kristall nicht schwingen kann.

Spezifikation

Maße: 42,1 × 24,2 × 10,6mm
 Gewicht: 3,8g
 IO Struktur: SCL,SDA,VCC,GND
 ROHS: Ja
 Vcc4,5 ~ 5,5V
 Logik High Level Eingang 2,2 - Vcc+0,3V
 Logik Low Level Eingang 0,3 - +0,8V
 Batterie Spannung 2,0 - 3,5V

Codebeispiel Arduino

```
#include "Wire.h"
#define DS1307_I2C_ADDRESS 0x68 // Das ist die I2C Adresse
// Globale Variablen
int command = 0; // Das ist der Befehl char, in ascii Form, aus dm seriellen Port gesendet
byte zero=0;
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
byte test;
```

```

byte decToBcd(byte val)
{
    return ( (val/10*16) + (val%10) );
}
// Konvertiert BCD-Code Zahlen zu normalen Dezimalzahlen
byte bcdToDec(byte val)
{
    return ( (val/16*10) + (val%16) );
}

void setDateDs1307(void)
{
    second = (byte) ((Serial.read() - 48) * 10 + (Serial.read() - 48));
    minute = (byte) ((Serial.read() - 48) * 10 + (Serial.read() - 48));
    hour = (byte) ((Serial.read() - 48) * 10 + (Serial.read() - 48));
    dayOfWeek = (byte) (Serial.read() - 48);
    dayOfMonth = (byte) ((Serial.read() - 48) * 10 + (Serial.read() - 48));
    month = (byte) ((Serial.read() - 48) * 10 + (Serial.read() - 48));
    year = (byte) ((Serial.read() - 48) * 10 + (Serial.read() - 48));
    Wire.beginTransmission(DS1307_I2C_ADDRESS);
    Wire.write(zero);
    Wire.write(decToBcd(second)); // 0 bis bit 7 startet die Uhr
    Wire.write(decToBcd(minute));
    Wire.write(decToBcd(hour)); // Wenn Sie eine 12 Stunden Uhr (am/pm) haben wollen, m
    // Bit 6 setzen (Es muss readDateDS1307 verändert werden)
    Wire.write(decToBcd(dayOfWeek));
    Wire.write(decToBcd(dayOfMonth));
    Wire.write(decToBcd(month));
    Wire.write(decToBcd(year));
    Wire.endTransmission();
}

// Ruft das Datum und die Uhrzeit aus dem DS1307
void getDateDs1307()
{
    // Registerzeiger wird zurückgesetzt
    Wire.beginTransmission(DS1307_I2C_ADDRESS);
    Wire.write(zero);
    Wire.endTransmission();
    Wire.requestFrom(DS1307_I2C_ADDRESS, 7);
    // Einige von diesen benötigen Masken, weil bestimmte Bits Steuerbits sind.
    second = bcdToDec(Wire.read() & 0x7f);
    minute = bcdToDec(Wire.read());
    hour = bcdToDec(Wire.read() & 0x3f); // Dies muss verändert werden,
    // wenn der 12Stunden Modus(am/pm) eingeschaltet ist
    dayOfWeek = bcdToDec(Wire.read());
    dayOfMonth = bcdToDec(Wire.read());
    month = bcdToDec(Wire.read());
    year = bcdToDec(Wire.read());

    Serial.print(hour, DEC);
    Serial.print(":");
    Serial.print(minute, DEC);
    Serial.print(":");
    Serial.print(second, DEC);
    Serial.print(" ");
    Serial.print(month, DEC);
    Serial.print("/");
    Serial.print(dayOfMonth, DEC);
    Serial.print("/");
    Serial.print(year, DEC);
    Serial.println(" ");
}

void setup()
{

```

LK-RTC

```

Wire.begin();
Serial.begin(9600);
}
void loop()
{
  delay(2000);
  /*T(00-59)(00-59)(00-23)(1-7)(01-31)(01-12)(00-99) - T(sec)(min)(hour)(dayOfWeek)(dayOfM
  - T stellt das Datum des RTC DS1307 Chip.
  Beispiel zum setzen der Zeit für 02-DEC-10 @ 19:57:11 für den dritten Wochentag, sende
  - T1157193021210
  */
  if (Serial.available())
  {
    // Wenn es festgestellt wurde, geben Sie char in der seriellen Warteschlange un
    command = Serial.read();
    if (command == 84)
    {
      //Wenn command = "T" setze Datum
      setDateDs1307();
      getDateDs1307();
      Serial.println(" ");
    }
    while(1)
    {
      getDateDs1307();
      delay(1000);
    }
  }
}
}

```

Raspberry Installation

Zunächst muss die Echtzeituhr an den I2C-Port des Raspberry-Linkerkit-Shields angeschlossen werden.

Als nächstes muss über die Kommandozeile das I2C-System installiert werden:

```
sudo apt-get install i2c-tools
```

Nun muss festgelegt werden, dass die benötigten Module auch beim Systemstart geladen werden:

```
sudo nano /etc/modules
```

Hier tragen wir folgende Module ein:

```
i2c-dev (Dieses Modul kann schon eingetragen sein)
snd-bcm2835
snd-bcm2837
spi-bcm2708
i2c-bcm2708
rtc-ds1307
```

Speichern Sie die Änderungen mit STRG + O ab. Anschließend Drücken Sie STRG + X um die Bearbeitung der Datei zu verlassen.

Nun müssen folgende Module aktiviert werden, geben Sie diese Befehle einfach in der Konsole ein:

LK-RTC

```
sudo modprobe i2c_bcm2708
sudo modprobe i2c_dev
sudo modprobe rtc_ds1307
```

Nun lassen wir das System nach einem I2C-Gerät suchen.

```
i2cdetect -y 1
```

Das Ergebnis sollte so aussehen:

```
0 1 2 3 4 5 6 7 8 9 a b c d e f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  ----  --  --  --  --  --  --  --  --  --  --  --  --  --  --  68  --  --  --  --  --  --
70:  ----  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Dies bedeutet, dass die Uhr unter der Adresse 68 gefunden wurde.

Nun müssen wir noch die Datei rc.local bearbeiten:

```
sudo nano /etc/rc.local
```

Damit die Uhrzeit bei jedem Neustart vom LK-RTC Modul ausgelesen wird, müssen wir folgende Zeilen in der rc.local Datei hinzufügen:

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
hwclock -w
```

Mit folgendem Befehl in der Konsole kann die aktuelle Uhrzeit ausgelesen werden:

```
sudo hwclock -r
```

Downloads

Infos:

Englisch: [001267860-da-01-en-linker_kit_platine_mit_rtc_zeituhr_.pdf](#)

Schaltung: [001267860-sp-01-en-linker_kit_platine_mit_rtc_zeituhr_.pdf](#)