Arduino Mega
Temperatur, Uhrzeit und Datum im zweizeiligen Display

Volker Schuhmaier

2023

```
/*
Copyright (c) 2023 by Volker Schuhmaier

 Permission is hereby granted, free of charge, to any person obtaining a copy
 of this software and associated documentation files (the "Software"), to deal
 in the Software without restriction, including without limitation the rights
 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
 copies of the Software, and to permit persons to whom the Software is
 furnished to do so, subject to the following conditions:

 The above copyright notice and this permission notice shall be included in all
 copies or substantial portions of the Software.

 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
 SOFTWARE.
```

```
21
22 Arduino Mega mit zweizeiligem LCD-Display, RTC und DHT11.
23 Temperatur und Luftfeuchtigkeit werden in der erten Zeile
      dargestellt.
24 Uhrzeit und Datum, im Wechel, werden in der zweiten Zeile
      dargestellt.
25 Es werden zwei Automaten (Timer_DHT11 und Diplaychange)
      verwendet,
26 dadurch wird die Uhrzeit jede Sekunde aktualisiert.
27 Die Aktualisierung von Temperatur und Luftfeuchte erfolgt pro
       Sekunde;
28 der DHT11 benoetigt 1 Sekunde zum Auslesen.
29 */
30
31
32 #include "RTClib.h"
33 #include <LiquidCrystal.h>
34 #include <SimpleDHT.h>
35
36 #define countof(a) (sizeof(a) / sizeof(a[0]))
37
38 char daysOfTheWeek[7][12] = { "Sunday", "Monday", "Tuesday",
      "Wednesday", "Thursday", "Friday", "Saturday" };
39 char daysOfTheWeek_DE[7][12] = { "Sonntag", "Montag", "
      Dienstag", "Mittwoch", "Donnerstag", "Freitag", "Samstag"
      };
40 char daysOfTheWeek_DE_kurz[7][12] = { "So", "Mo", "Di", "Mi",
       "Do", "Fr", "Sa" };
41
42 int pinDHT11 = 2; // Data Pin 2
43
44 // Generally, you should use "unsigned long" for variables
      that hold time
45 // The value will quickly become too large for an int to
      store
46 bool boolTimer_DHT11Read = 0;
47 bool boolDisplayChange = 0;
48 bool boolDisplayTime = 0;
49
50 byte byteTemperature = 0;
51 byte byteHumidity = 0;
52
53 int err = SimpleDHTErrSuccess;
54
55 const long longIntervalDHT11 = 500;   // (milliseconds)
      Interval fuer Anzeige des DHT11, allerdings dauert die
      Abfrage des DHT11 eine Sekunde
56 const long longDisplayChange = 5000;  // (milliseconds)
      Interval fuer Wechsel zwischen Uhrzeit und Datum
```

```
57
58 unsigned long ulongCurrMillis = millis(); //current
      Milliseconds
59 unsigned long ulongPrevMillisDHT11 = 0;    // previous
      Milliseconds for DHT read
60 unsigned long ulongPrevMillisDisplay = 0;  // previous
      Milliseconds for Display change
61
62 RTC_DS3231 rtc;
63
64 SimpleDHT11 dht11(pinDHT11);
65
66 LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
67
68 void setup()
69  {
70     lcd.begin(16, 2);
71     rtc.begin();
72  }
73
74 void loop()
75  {
76   DateTime now = rtc.now();
77
78   err = SimpleDHTErrSuccess;
79
80   if ((err = dht11.read(&byteTemperature, &byteHumidity, NULL
      )) != SimpleDHTErrSuccess)
81    {
82     return;
83    }
84
85   ulongCurrMillis = millis();
86
87   if (ulongCurrMillis - ulongPrevMillisDHT11 >=
      longIntervalDHT11)
88    {
89     ulongPrevMillisDHT11 = ulongCurrMillis;
90     boolTimer_DHT11Read = 1;  // refresh time and humidity on
      Display
91    }
92
93   if (ulongCurrMillis - ulongPrevMillisDisplay >=
      longDisplayChange)
94    {
95     ulongPrevMillisDisplay = ulongCurrMillis;
96     boolDisplayChange = 1;  // refresh Display with time or
      date
97    }
```

```
98
99  if (boolTimer_DHT11Read == 1)
100  {
101   lcd.setCursor(0, 0);
102   lcd.print("T/H:   C /   %");
103   lcd.setCursor(5, 0);
104   lcd.print(byteTemperature);
105   lcd.setCursor(11, 0);
106   lcd.print(byteHumidity);
107
108   boolTimer_DHT11Read = 0;  // Refresh done
109  }
110
111  // Display wird mit Uhrzeit und Datum getoggled
112  if (boolDisplayChange == 1)
113  {
114    boolDisplayTime = !boolDisplayTime;
115
116    boolDisplayChange = 0;
117  }
118
119  // Display-Auffrischung mit Datum
120  if (boolDisplayTime == 0)
121  {
122   char charDatestring[20];
123   snprintf_P(charDatestring,
124              countof(charDatestring),
125              PSTR("%02u.%02u.%04d"),
126              now.day(),
127              now.month(),
128              now.year());
129
130   lcd.setCursor(0, 1);
131   lcd.print(daysOfTheWeek_DE_kurz[now.dayOfTheWeek()]);
132   lcd.print(" ");
133   lcd.print(charDatestring);
134  }
135  // Display-Auffrischug mit Uhrzeit
136  if (boolDisplayTime == 1)
137  {
138   char charTimestring[20];
139   snprintf_P(charTimestring,
140              countof(charTimestring),
141              PSTR("%02u:%02u:%02u"),
142              now.hour(),
143              now.minute(),
144              now.second());
145
146   lcd.setCursor(0, 1);
```

```
147     lcd.print("                ");  // Loesche den alten
        Inhalt
148     lcd.setCursor(0, 1);
149     lcd.print(charTimestring);
150    }
151 }
```