

```

1  /*
2  * C++ Quellcode fuer LCD-Anzeige (1602A)
3  * zur Ausgabe der Uhrzeit mit Zeitzone, des Wochentages & des Datums.
4  * Laufzeitumgebung: Linux auf Raspberry Pi 3 Model 8 V1.2
5  * Format:      hh:mm:ss Zone
6  *              Tag_ dd mon yyyy
7  */
8
9
10 #include <stdio.h>
11 #include <stdlib.h>
12 #include <time.h>
13 #include <lcd.h>
14 #include <wiringPi.h>
15
16
17
18 int disp;                                // Variable fuer Anzeige
19 char *buf1;                              // Puffer Zeile 1
20 char *buf2;                              // Puffer Zeile 2
21 char str [] = "  empty buffer  ";        // Zeichenfolge f. Platzhalter
22
23
24 PI_THREAD(refreshDisp)                   // Thread fuer Displayaktualisierung
25 {
26     int i;
27     while (TRUE)
28     {
29         for (i = 0; i < 16; i++)          // Iteration ueber Displayspalten
30         {
31             lcdPosition(disp, i, 0);      // Cursor-Position in Zeile 1 setzen
32             lcdPutchar(disp, *(buf1 + i)); // Inhalt des 1. Puffers schreiben
33                                             // & Pointer inkrementieren
34             lcdPosition(disp, i, 1);      // wie zuvor,
35             lcdPutchar(disp, *(buf2 + i)); // nur mit 2. Puffer fuer 2. Zeile
36         }
37         delay(50);                        // refresh-Zeit der Anzeige (50 ms)
38     }
39 }
40
41
42 PI_THREAD(writeTimeDateBuffer)            // Thread zum fuellen des Puffers
43                                           // mit Datum- & Zeit-String
44 {
45     lcdClear(disp);                      // Anzeige loeschen
46     buf1 = (char*)"  lcdTimeDate  ";    // Appname bei Initialisierung
47     delay(1000);                         // Darstellungszeit
48
49     time_t tnow;                         // Datentyp d. Zeit (time.h)
50     char tme[17];                       // Puffer Zeit
51     char dte[17];                       // ... Datum
52
53     while (TRUE)
54     {
55         time(&tnow);                    // Zeitabfrage
56         strftime(tme, 17, " %H:%M:%S %Z ", localtime(&tnow)); // Formatierung Zeit & Zeitzone
57         buf1 = &tme[0];                 // Pointer 1. Zeile auf Zeitpuffer
58         strftime(dte, 17, "%a %e %b %G ", localtime(&tnow)); // Datumsformatierung
59         buf2 = &dte[0];                 // Pointer 2. Zeile auf Datumpuffer
60         delay(50);
61     }
62 }
63

```

```

64 int initDisplay()
65 {
66     buf1 = &str[0];
67     buf2 = &str[0];
68
69     disp = lcdInit(
70         2, // Initialisierung LCD-Anzeige 1602A
71         16, // Zeilen
72         4, // Spalten
73         26, // Bandbreite Daten (bit)
74         27, // rs-pin (register select)
75         25, 23, 22, 21, // strb-pin (enable)
76         0, 0, 0, 0 // 4-bit Daten (D0, D1, D2, D3)
77     ); // 8-bit Daten +(D4, D5, D6, D7)
78 // config-digits for wiring Pi-pins
79 // : wirPi(GPIO_GEN)->BCM(GPIO) :
80 // (26~GPIO12, 27~GPIO16, 25~GPIO26)
81
82
83     printf("display: %d\n", disp);
84
85     if (disp == -1) // Initialisierung fehlerfrei?
86     {
87         printf("lcdInit 1 failed\n");
88         return -1;
89     }
90
91     lcdClear(disp); // LCD-Anzeige loeschen
92     delay(50);
93
94     return 0;
95 }
96
97
98 int main(int argc, char** args)
99 {
100     printf("application: lcdTimeDate\n");
101
102     if (wiringPiSetup() == -1) // Setup der wiringPi Bibliothek
103     {
104         printf("setup wiringPi failed\n");
105         return -1;
106     }
107
108
109     initDisplay(); // Anzeige initialisieren
110
111     piThreadCreate(refreshDisp); // Aktual. der Anzeige starten
112
113     piThreadCreate(writeTimeDateBuffer); // Aktualisierung
114 // des Zeit-/Datumpuffers starten
115
116     while (TRUE) // Laufzeit der Threads sichern
117     {
118         delay(10000);
119     }
120
121     return 0; // (Hauptprogramm beenden)
122 }

```

