

```

1  /*
2   Program LCD DE 119, Jurgen Rieger, Januar 2021
3
4   Measure a DC-Value and show the Result at LCD-Modul
5   μC MSP430F477
6   LCD-Modul: DE 119, 4 Digits
7
8   first init the MSP430F477
9   than, capture a Voltage Value (Akku or Solar)
10  change the Measuretement in a 4 Digit Value
11  show it at 4 Digt LCD DE 119
12
13 μC Pins and its Value -> LCD Modul Segments
14 -----
15 LCDM1=1;      -> Segment 1a
16 LCDM1=0x10;    -> Segment 1b
17 LCDM2=1;      -> Segment 1c
18 LCDM2=0x10;    -> Segment 1d
19 LCDM3=1;      -> Segment 1e
20 LCDM3=0x10;    -> Segment 1f
21 LCDM4=1;      -> Segment 1g
22 LCDM4=0x10;    -> Segment 1dp
23 LCDM5=1;      -> Segment 2a
24 LCDM5=0x10;    -> Segment 2b
25 LCDM6=1;      -> Segment 2d
26 LCDM6=0x10;    -> Segment 2d
27 LCDM7=1;      -> Segment 2e
28 LCDM7=0x10;    -> Segment 2f
29 LCDM8=1;      -> Segment 2g
30 LCDM8=0x10;    -> Segment 2dp
31 LCDM9=1;      -> Segment 3a
32 LCDM9=0x10;    -> Segment 3b
33 LCDM10=1;     -> Segment 3c
34 LCDM10=0x10;   -> Segment 3d
35 LCDM11=1;     -> Segment 3e
36 LCDM11=0x10;   -> Segment 3f
37 LCDM12=1;     -> Segment 3g
38 LCDM12=0x10;   -> Segment 3dp
39 LCDM13=1;     -> Segment 4a
40 LCDM13=0x10;   -> Segment 4b
41 LCDM14=1;     -> Segment 4c
42 LCDM14=0x10;   -> Segment 4d
43 LCDM15=1;     -> Segment 4e
44 LCDM15=0x10;   -> Segment 4f
45 LCDM16=1;     -> Segment 4g
46 */
47 #include <msp430x47x.h> // Standard Definition header file for MSP430F477
48 #include <stdio.h>
49 typedef unsigned int word; // Type definition for 'word'
50
51 // Function prototypes.
52 void init_sys(void); // System initialization.
53 void voltage(void); // Millivoltmeter mode.
54 void delay(int K); // Delay
55 void Digit_Value(char *point, const char Digits[8]);
56 void Select_LCD_Digit(const char Digit[4],char Digit);
57
58 // Define Segments at Digits
59 #define a 1
60 #define b 0x10
61 #define c 1
62 #define d 0x10
63 #define e 1
64 #define f 0x10
65 #define g 1
66
67
68 const char zero[8]      ={a,b,c,d,e,f,0,0};
69 const char one[8]        ={0,b,c,0,0,0,0,0};
70 const char two[8]        ={a,b,0,d,e,0,g,0};
71 const char three[8]       ={a,b,c,d,0,0,g,0};
72 const char four[8]        ={0,b,c,0,0,f,g,0};
73 const char five[8]        ={a,0,c,d,0,f,g,0};
74 const char six[8]         ={a,0,c,d,e,f,g,0};
75 const char seven[8]       ={a,b,c,0,0,0,0,0};
76 const char eight[8]       ={a,b,c,d,e,f,g,0};

```

```

77 const char nine[8]      = {a,b,c,d,0,f,g,0};
78 const char AH[8]         = {a,b,c,0,e,f,g,0};
79 const char BH[8]         = {0,0,c,d,e,f,g,0};
80 const char CH[8]         = {a,0,0,d,e,f,0,0};
81 const char DH[8]         = {0,b,c,d,e,0,g,0};
82 const char EH[8]         = {a,0,0,d,e,f,g,0};
83 const char FH[8]         = {a,0,0,0,e,f,g,0};
84 const char blank[8]       = {0,0,0,0,0,0,0,0};
85
86 const char Digit1[4]     = {0x91,0x92,0x93,0x94};
87 const char Digit2[4]     = {0x95,0x96,0x97,0x98};
88 const char Digit3[4]     = {0x99,0x9a,0x9b,0x9c};
89 const char Digit4[4]     = {0x9d,0x9e,0x9f,0xa0};
90
91 char Buffer [5];
92 char Digit;
93
94 //-----
95 // Sub-Routine init
96 void init_sys(void)
97 {
98     int i;
99     // Configure all unused port pins as outputs.
100    P1DIR = 0xff;
101    P1OUT = 0;
102    P2DIR = 0x71;
103    P2OUT = 0;
104    P3DIR = 0xff;
105    P3OUT = 0;
106    P4DIR = 0xff;
107    P4OUT = 0;
108    P5DIR = 0xff; // P5.2 LED
109    P5OUT = 0;
110    P6DIR = 0xff;
111    P6OUT = 0;
112    P6SEL = BIT0 | BIT1; // P6.0 A0 In +, P6.1 A0 In -
113    P5OUT = 4; // LED ein
114    delay(500);
115    P5OUT = 0; // LED aus
116
117 // Init Sigma Delta SD16_A
118 SD16CTL = SD16SSEL_2; // ACLK, externe Referenz
119 SD16CCTL0 = SD16UNI | SD16SNGL;
120
121
122 // handle LCD memory
123 LCDACTL = LCDFREQ_64 | LCDSTATIC | LCDSON | LCDON; // LCD Freq / 64, Static, LCD ON
124 LCDAPCTL0 = 0xff;
125 LCDAPCTL1 = LCDS28; // segments = 0 - 35.
126 for (i = 0; i < 20; LCDMEM[i++] = 0); // Clear LCD memory.
127
128
129 // _EINT(); // Enable interrupts.
130 } // sys_init()
131
132 //-----
133 // subroutine, delay
134 void delay(int K)
135 {
136     int i,j;
137     for (i=1;i<1000;i++) for (j=1;j<K;j++);
138 }
139
140 //-----
141 // subroutine, write Segment at Segmentbuffer (LCD Memmory)
142 void Digit_Value(char *point, const char Digits[8])
143 {
144     *point++=Digits[0]+Digits[1];
145     *point++=Digits[2]+Digits[3];
146     *point++=Digits[4]+Digits[5];
147     *point++=Digits[6]+Digits[7];
148 }
149
150 //-----
151 // subroutine, change Digit in a Segment
152 void Select_LCD_Digit(const char Digit[4],char Digit)

```

```

153     {
154     char *point;
155     point = (char*)Digit[0];
156
157     switch (Digit)
158     {
159         case '0': Digit_Value(point,zero);
160         break;
161         case '1': Digit_Value(point,one);
162         break;
163         case '2': Digit_Value(point,two);
164         break;
165         case '3': Digit_Value(point,three);
166         break;
167         case '4': Digit_Value(point,four);
168         break;
169         case '5': Digit_Value(point,five);
170         break;
171         case '6': Digit_Value(point,six);
172         break;
173         case '7': Digit_Value(point,seven);
174         break;
175         case '8': Digit_Value(point,eight);
176         break;
177         case '9': Digit_Value(point,nine);
178         break;
179     default: Digit_Value(point,zero);
180         break;
181     }
182 }
183
184
185 // Sigma Delata Converter
186 //-----
187
188 // subroutine, Measure Voltage an show it
189 void voltage(void)
190 {
191     int i;
192 /*
193     conversion DC Value
194 -----
195     change Inkremnts in Voltage: U=3,48V / 4096 * ADC12MEMO (Reference extern: 3,48V)
196     change Voltage Value in BCD Digits and write at LCD
197
198     Start SDC16
199     - Analog Channel 0
200     - SD16AE0: negativ inputs connected to Vss
201     - SD16UUNI: Unipolar-Mode
202     - SD16SNGL: Single Conversion
203     - SD16OSRX: 256
204 */
205     SD16CCTL0 &= ~SD16IFG;
206     SD16CCTL0 &= ~SD16OVIFG;
207     SD16CCTL0 |= SD16SNGL;
208     SD16CCTL0 |= SD16UNI;
209     SD16CCTL0 |= SD16OSR0;
210     SD16CCTL0 |= SD16SC;           // Start Conversion
211     // wait, Convertierung is finish
212     while (!(SD16CCTL0 & SD16IFG));
213     // clear Buffer
214     for (i=0;i<10;i++) Buffer[i] = 0;
215     // change SD16MEMO in BCD and write to Buffer
216     sprintf(Buffer,"%4d",SD16MEM0);
217     delay(1000);
218     // write Buffer to LCD (Measure Value at LCD)
219     Select_LCD_Digit(Digit1,Buffer[0]); // write Digit 0
220     Select_LCD_Digit(Digit2,Buffer[1]); // write Digit 1
221     Select_LCD_Digit(Digit3,Buffer[2]); // write Digit 2
222     Select_LCD_Digit(Digit4,Buffer[3]); // write Digit 3
223 } // voltage()
224
225
226 //
-----
```

```
227 void main(void)
228 {
229     WDTCTL = WDTPW+WDTHOLD;           // Stop watchdog timer
230     init_sys();
231     /*
232     IE2 = 0x00;                      // Disable Basic Timer interrupt.
233     while (!(IFG2 & (char) BTIFG)); // Poll for Basic Timer interrupt.
234     IE2 |= BTIFG;                  // Enable Basic Timer interrupt.
235     */
236     while (1) voltage();
237 } // main()
238
239
```