Friedrich-Ebert-Schule Esslingen	MIKROCONTROLLER	Name:
4.6.4.1	Verwenden der Display-Bibliotheksfunktionen	Datum:

Zur Nutzung der Bibliotheksfunktionen muss die Datei **lcd8.a51** ins Projekt eingebunden werden. Weiterhin ist ins Projektverzeichnis (z.B. C:\controller\c5131) die Datei **LCD8.inc** zu kopieren. Dort werden globale Symbole zur Displaysteuerung definiert. In jedem Assemblermodul das diese benötigt, muss die Datei inkludiert werden.

LCD8.inc

```
;*******************************************************************************************
;** Symboldefinitionen für LC-Display, R.Rahm, 13.12.06
;** Vor Inklusion dieser Datei muss die Umgebungsvariable LCD16 oder LCD17 gesetzt werden
;*******************************************************************************************
        ;$Set (LCD20)            ; Hier die Umgebungsvariable für das verwendete Display
        $Set (LCD16)             ; setzen! Unbedingt vor Inclusion von LCD.inc

        backlight       bit p1.7        ; backlight = 0 --> Hintergrundbeleuchtung ein!
        busyflag        bit acc.7       ; Busy-Flag wird in Akkubit 7 zurücklesen!

        WriteLCDCommand equ 00h         ; 8 Bit Schreib- und Leseadressen für Display
        ReadBusy        equ 01h         ; Lesen mit MOVX a,@Rn
        WriteLCDData    equ 02h         ; Schreiben mit MOVX @Rn,a
        ReadLCDData     equ 03h         ; !!! Keinen Datenpointer verwenden !!!
                                        ;   (16 Bit Adressierung benötigt P2)
        ;Steuerbefehle
        DisplayClear            equ     00000001b       ; LCD-Befehle
        ReturnHome              equ     00000010b
        CursorOn                equ     00001110b
        CursorOff               equ     00001100b
        CursorShiftRight        equ     00010100b
        CursorShiftLeft         equ     00010000b
        DisplayShiftRight       equ     00011100b
        DisplayShiftLeft        equ     00011000b

        Z1_Start_Adress         equ     0h      ; DDRAM-Adressen für Zeilenanfänge
        Z2_Start_Adress         equ     40h
$if defined (LCD20)
        Z3_Start_Adress         equ     14h     ; Abhängig von der Displaygröße
        Z4_Start_Adress         equ     54h
$elseif defined (LCD16)
        Z3_Start_Adress         equ     10h
        Z4_Start_Adress         equ     50h
$endif
```

Die Bibliotheksfunktionen der Datei **lcd8.a51** sind als Unterprogramme definiert. Dabei sind immer der Akku und/oder der Datenpointer das Übergaberegister:

Displaysteuerung	
init_disp	Initialisierung muss vor jeder Nutzung des Displays durchgeführt werden!
define_char	Definition von eigenen Sonderzeichen
WaitWhileBusy	Das Busy-Flag sollte vor jedem Schreib-/Lesevorgang abgefragt werden!
clear_disp	Gesamtes Display löschen
clear_zeile1	Zeile 1 löschen
clear_zeile2	Zeile 2 löschen
clear_zeile3	Zeile 3 löschen
clear_zeile4	Zeile 4 löschen

Cursorsteuerung	
cursor_home	Cursor auf DDRAM Adresse 00h
cursor_rechts	Cursor eine Position nach rechts
cursor_links	Cursor eine Position nach links
set_cursor	Cursor auf beliebige Adresse setzen
cursor_on	Cursor Einschalten
cursor_off	Cursor Ausschalten

Zeichenausgabe	
out_char	Ausgabe eines Zeichens auf der aktuellen Cursorposition
out_disp	Ausgabe einer mit 0 terminierten konstanten Zeichenkette
out_var	Ausgabe einer mit 0 terminierten variablen Zeichenkette
text_zeile1	Ausgabe einer 0 terminierten Zeichenkette in Zeile 1
text_zeile2	Ausgabe einer 0 terminierten Zeichenkette in Zeile 2
text_zeile3	Ausgabe einer 0 terminierten Zeichenkette in Zeile 3
text_zeile4	Ausgabe einer 0 terminierten Zeichenkette in Zeile 4

Friedrich-Ebert-Schule Esslingen	MIKROCONTROLLER	Name:
4.6.4.2	Verwenden der Display-Bibliotheksfunktionen	Datum:

Vor dem Aufruf der Bibliotheksfunktionen müssen die benötigten Unterprogramme als extern deklariert werden:

```
extern code init_disp
extern code define_char, clear_disp, clear_zeile1, clear_zeile2, clear_zeile3, clear_zeile4
extern code cursor_home, cursor_rechts, cursor_links, set_cursor, cursor_on, cursor_off
extern code out_char, out_disp, out_var, text_zeile1, text_zeile2, text_zeile3, text_zeile4
```

Funktionsaufrufe

- Beispiele für parameterlose Funktionen

```
        lcall   init_disp

        lcall   clear_disp

        lcall   cursor_home
```

- Beispiele für Funktionen mit Übergabeparameter im Akku

```
        mov     a,#'A'
        lcall   out_char

        mov     a,#47h
        lcall   set_cursor
```

- Beispiel für Funktion mit Übergabeparameter im Datenpointer

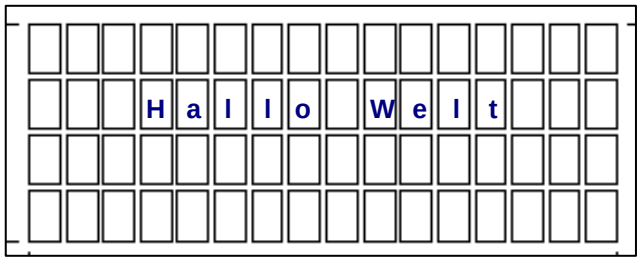
```
        mov     dptr,#text1
        lcall   text_zeile2
          ...
text1:  db "Hallo",0
```

- Beispiel für Funktion mit Übergabeparameter in Akku und Datenpointer

```
        mov     dptr,#zeichen1
        mov     a,#001000b
        lcall   define_char
```

Übungen:

1. Schreiben Sie ein Assemblerprogramm, welches den Text „**Hallo Welt**“ auf Zeile 2 des LC-Displays ausgibt. Verwenden Sie die Funktion **text_zeile2**:



2. Die an Port 1 (DIP-Schalter) eingestellte Dualzahl soll als 7 Bit ASCII-Zeichen auf Zeile 2 am Display ausgegeben werden. (Verwenden Sie die Funktion **set_cursor** und **out_char**)

3. Geben Sie in der gleichen Zeile die Dualzahl als 2-stelligen Hexwert aus! (Hinweis: Der ASCII-Offset für die Ziffern 0..9 beträgt **30h** und für die Buchstaben A..F **37h**).