



Dokumentation Ansteuerung des LCD 4 Zeilen zu je 27 Zeichen

Das Display ist über ein 4-bit-Interface + Steuerzeichen an den PIC18F452 angebunden.

Die folgenden Funktionen sind für den Anwender sichtbar ausgeführt, d.h. sie sind in der display.h als Prototypen deklariert:

- void LcdInit(void);
- void LcdClear(void);
- void LcdClearLine(UBYTE Line);
- void LcdPutCharacter(int Line, int Position, char Value);
- int LcdPutCharacterHere(char Value);
- void LcdSetCursorPosition(int Line, int Position, int CursorMode);
- void LcdSetMode(int mode);
- void LcdPrintf(int Line, int Position, auto const MEM_MODEL rom char *fmt, ...);
- int LcdGetCurrentLine(void);
- int LcdGetCurrentPosition(void);

void LcdInit(void);

Das Display wird initialisiert. Abhängig vom Status des RCON-Registers werden entweder die Hardware komplett initialisiert, oder, falls kein Power-On Reset vorliegt, wird nur das Display gelöscht.

void LcdClear(void);

Das Display wird komplett gelöscht und der Cursor wird an die Position 0 der ersten Zeile gebracht. Der Cursormodus bleibt unverändert.

Achtung! Die Werte, die LcdGetCurrentLine() und LcdGetCurrentPosition() nach diesem Aufruf zurückgeben, sind undefiniert.



void LcdClearLine(UBYTE Line);

Die angegebene Zeile wird komplett gelöscht. Die Cursorposition ist danach undefiniert. Der Cursormodus bleibt unverändert.

Gültige Werte für Zeile sind 0 .. 3

Achtung! Die Werte, die LcdGetCurrentLine() und LcdGetCurrentPosition() nach diesem Aufruf zurückgeben, sind undefiniert.

void LcdPutCharacter(int Line, int Position, char Value);

Schreibt ein Zeichen an die angegebene Position in der Zeile die durch Line definiert ist. Gültige Werte für Zeile sind 0 .. 3,

für Position 0 .. 26,

für Value siehe die Zeichenliste des Display-Datenblattes.

Achtung! Die Werte, die LcdGetCurrentLine() und LcdGetCurrentPosition() nach diesem Aufruf zurückgeben, sind undefiniert.



void LcdSetCursorPosition(int Line, int Position, int CursorMode);

Setzt den Cursor an eine bestimmte Position in der durch Line definierten Zeile. Der Cursor wird in den Modus gesetzt, der durch CursorMode vorgegeben ist.

Gültige Werte für Zeile sind 0 .. 3,

für Position 0 .. 26,

für CursorMode → siehe Tabelle

	Wert = 0	Wert = 1	definierter Wert
Bit 0	Cursor unsichtbar	Cursor sichtbar	DISPLAY_ON
Bit 1	Cursor blinkt nicht	Cursor blinkt	CURSOR_VISIBLE
Bit 2	Display aus	Display ein	CURSOR_FLASH

In display.h sind zusätzlich folgende Werte definiert:

ALL_OFF für Display aus und Cursor unsichtbar.

UNCHANGED für unveränderten Cursor.

Die Werte, die LcdGetCurrentLine() und LcdGetCurrentPosition() zurückgeben, sind ab diesem Aufruf definiert.



int LcdPutCharacterHere(char Value);

Es wird ein Zeichen das durch Value definiert ist, an die momentante Cursorposition geschrieben.

Gültige Werte für Value siehe die Zeichenliste des Display-Datenblattes.

Die Position, an die das zeichen gesetzt wird, ist dieselbe wie die, die durch LcdGetCurrentLine() und LcdGetCurrentPosition() vor diesem Aufruf zurückgeben wird. Das heißt, vor einem oder mehreren aufeinanderfolgenden Aufrufen muß ein Aufruf von LcdSetCursorPosition (...) erfolgen.

Rückgabewerte: Wenn Zeile und Position intern undefiniert sind, kommt die Funktion mit -1 zurück, anderenfalls mit 0.

void LcdSetMode(int mode);

Der Cursor wird in den Modus gesetzt, der durch CursorMode vorgegeben ist.

Gültige Werte für Zeile sind 0 .. 3,

für Position 0 .. 26,

für CursorMode → siehe Tabelle

	Wert = 0	Wert = 1	definierter Wert
Bit 0	Cursor unsichtbar	Cursor sichtbar	DISPLAY_ON
Bit 1	Cursor blinkt nicht	Cursor blinkt	CURSOR_VISIBLE
Bit 2	Display aus	Display ein	CURSOR_FLASH

In display.h sind zusätzlich folgende Werte definiert:

ALL_OFF für Display aus und Cursor unsichtbar.

UNCHANGED für unveränderten Cursor.



void LcdPrintf(int Line, int Position, auto const MEM_MODEL rom char *fmt, ...);

Schreibt einen String an die Position in der Displayzeile, die durch Line definiert ist.

Für die Formatdefinition fmt siehe printf in der Standard-String-Bibliothek.

Achtung! Siehe andererseits auch die Kommentare im Internet über die Fehler, die printf in dem C18-Compiler noch hat! Nach meiner Erfahrung konnte man nur Werte ausgeben, die im unsigned long Format vorlagen, mit der Definition %ul. Bytes und %d arbeiten nicht korrekt. Bitte weiter gegentesten! Ein cast von char nach unsigned long kann hier helfen.

int LcdGetCurrentLine(void);

Gibt einen Wert zurück, der der momentanen Ausgabezeile entspricht. Gültig sind 0..3 oder 0xff für undefiniert. Der Wert ist immer dann gültig, wenn LcdSetCursorPosition (..) gerufen wurde und der Wert nicht durch Aufrufe von LcdPrintf, LcdClear oder LcdPutCharacter gelöscht wurde. Rückgabe 0..3, oder 0xff für ungültig.

int LcdGetCurrentPosition(void);

Gibt einen Wert zurück, der der momentanen Ausgabeposition entspricht. Gültig sind 0..26 oder 0xff für undefiniert. Der Wert ist immer dann gültig, wenn LcdSetCursorPosition (..) gerufen wurde und der Wert nicht durch Aufrufe von LcdPrintf, LcdClear oder LcdPutCharacter gelöscht wurde. Rückgabe 0..26, oder 0xff für ungültig.