3 Bedienung des Impedanzmessgerätes

3.1 LCD

3.1.1 Beschreibung LCD Ansteuerung

Anschlussbelegung des LCD

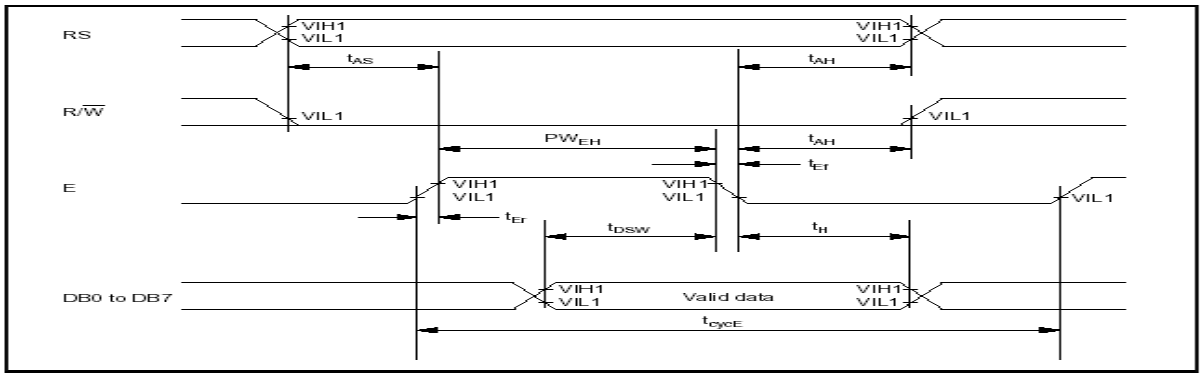
Port C PC7 = E
PC6 = RS
PC5 = R/W
PC04 bist PC7 = DB4 bis DB7

Initialisierung
Zuerst muss das Display vom Atmega 32 initialisiert werden. Der Displaycontroller kennt eine Power-on-Reset-Funktion. Diese läuft beim Anlegen der Betriebsspannung automatisch ab und bringt das Display in folgenden Zustand:

- 8-bit-Interface
- 1-zeiliges Display
- 5x8-Punkt-Matrix
- Display aus / Cursor: aus / Cursor-blinken: aus
- Displayshift: aus
- Cursor geht bei jedem neuen Zeichen nach rechts

Während der Reset-Prozedur kann das Display keine Befehle von aussen annehmen. Nach dem Reset ist das Display ausgeschaltet; es kann nicht benutzt werden. Eine Initialisierungsroutine sollte das Display aus diesem Zustand in den gewünschten Betriebsmode bringen.

Schreiben zum Display



Die Abbildung zeigt einen Schreibzugriff zum Display. Das Display überwacht den Pegel von ENABLE (E). Ändert sich dieser Pegel von Low nach High, wird die Leitungen RS und R/W abgefragt. Liegt R/W auf Low, dann ist es ein Schreibzugriff. Nun werden mit der High-Low Flanke von ENABLE (E) die Daten vom Datenbus (DB0..DB7) eingelesen. Liegt RS auf Low, werden die Daten als Kommando verstanden und in ein Steuerregister geschrieben. Liegt aber RS auf High, handelt es sich um Daten, die angezeigt werden.

Um eine sichere Verarbeitung einzuhalten, sind Abstände zwischen dem Einstellen von RS und R/W sowie der ENABLE-Low-High-Flanke einzuhalten. Der Datenbus muss während der High-Low-Flanke von ENABLE herum stabil sein, und RS und R/W dürfen erst nach ENABLE abgeschaltet werden. Diese Zeiten sind aber so kurz, dass man sie bei der Ansteuerung durch einen uC vergessen kann.

LCD 4 Bit Modus

Das oben beschriebene Interface mit seinem 8-Bit breiten Datenbus hat den Nachteil, dass zum Anschluss des Displays an den steuernden Prozessor 11 Leitungen nötig sind.
Wenn man einen Prozessor mit reichlich I/O-Pins benutzt, ist das nicht problematisch, aber beim Atmega32 mit nur 32 Pin muss gut überlegt werden wie man die Pin einsetzt.

Das LCD -Displays kennt ein Interface, bei dem nur die oberen 4-Bit des Datenbusses (D7..D4) benutzt werden. Das verringert die Zahl der benötigten I/O-Pins auf 7. Da immer noch 8-Bit Werte über den Datenbus übertragen werden müssen, geschieht dies in 2 Hälften. Bei jedem Lese- oder Schreibzugriff werden auf dem Datenbus also nacheinander erst die 4 oberen Bits und dann die 4 unteren Bits des 8-Bit-Datenworts übertragen. Dazu muss Enable zweimal ein- und wieder ausgeschaltet werden.

3.1.2 Software LCD Struktogramme my_lcd_4bit.c

Schreiben Kommando zu Initialisierung

LCD_Inst_data (unsigned char Komando)

Schreiben Komando zum Display

Daten auf auf Port schreiben
R/W ------> Low Schreibzugriff
RS ------> low Komando Daten
Enabel Low ----> Hi Abfrage von RS und R/W
Kurtz Warten
Enabel Hi ----> Low Einlesen der Daten als Kommando

Diese Unterprogramm wird zur Initialisierung im 8 Bit Interface gebraucht. Damit werden Kommandos im 8 Bit Modus zum LCD gesendet.

LCD_Inst_data_set (unsigned char Komando)

Schreiben Komando zum Display

Temp. Variable
lcd_4bit_Hi (Instration); Hi bit heraus Filtern
LCD_Inst_Data(Instration); Kommando auf Port schreiben HI
Wait (Zeit); Warten
lcd_4bit_LW (Instration); LW bit heraus Filtern
LCD_Inst_Data(Instration); Kommando auf Port schreiben LW
Wait (Zeit); Warten

Das Unterprogramm ist für die Initialisierung im 4 Bit Modus. Damit werden Steuerbefehle zu LCD gesendet.

void lcd_init (void)

Initalisierung

Port auf Ausgang schalten
Warten
LCD_Inst_data (0x30) Interface auf 8-bit setzen
Warten
LCD_Inst_data (0x30) Interface auf 8-bit setzen
Warten
LCD_Inst_data (0x30) Interface auf 8-bit setzen
LCD_Inst_data (0x20) Interface auf 4-bit setzen
LCD_Inst_data_set (0x08); Display off / Display löschen
LCD_Inst_data_set (0x01); Display löschen
Warten
LCD_Inst_data_set (0x06); +1increment,no shift Laufrichtung
LCD_Inst_data_set (0x02); cusor home
Warten
LCD_Inst_data_set(0x0C); display on, cursor on
LCD_Inst_data_set(0x80); 1st lin 1 column

Initialisierung LCD 4 Bit

Als erstes müssen die Pins wird als Ausgang geschaltet werden. Danach wird das LCD dreimal im 8-Bit Modus initialisiert. Nun wird das LCD auf 4-Bit initialisiert. Die Kommandos können nur noch mit der LCD_Inst_data zum LCD gesendet werden. Zum Schluss muss das LCD noch in den gewünschten Betriebsmodus gebracht werden.

Daten Aufteilen auf LW und HI

lcd_4bit_HI (unsigned char Daten);

Filtern von dem Hi Bit

Temp. Variable
High nibble nach High schieben
Rückgabew ert HI

lcd_4bit_LW (unsigned char Daten);

Filtern von dem Low Bit

Temp. Variable
High nibble nach High schieben
Rückgabew ert HI

Dieses kleine Unterprogramm wird gebraucht um die Daten oder Kommandos ins 4 Bit Interface zu dekodieren. Ein Buchstabe ist immer ein Charakter (char) mit 8 Bit. Die Aufteilung wird in diesen Routinen erledigt.

Schreiben Daten zum Display

LCD_data (unsigned char daten)

Schreiben Daten zum Display

Daten auf Port Legen
R/W ------> Low Schreibzugriff
RS ------> Hi Komando Daten
Enabel Low ----> Hi Abfrage von RS und R/W
Kurtz Warten
Enabel Hi ----> Low Einlesen der Daten als Kommando

Diese Routine ist für die Übernahmen der Daten, die am Display anliegen. Im 8 Bit-Modus kann so ein ASCII-Zeichen auf dem LCD angezeigt werden.

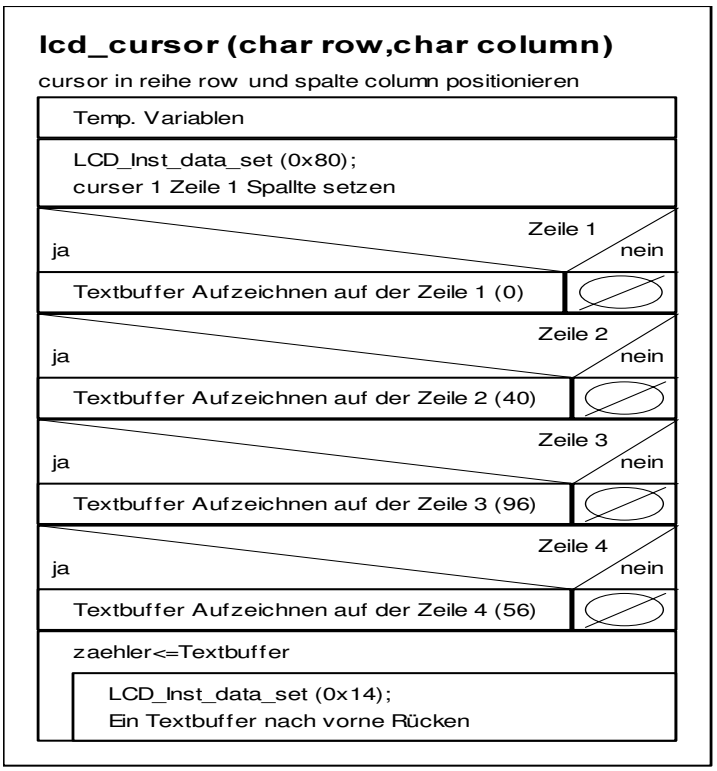
LCD_data_set (unsigned char Daten)

Schreiben Daten zum Display

Temp. Variable
lcd_4bit_Hi (Instration); Hi bit heraus Filtern
LCD_data (Instration); Daten auf Port schreiben HI
Wait (Zeit); Warten
lcd_4bit_LW (Instration); LW bit heraus Filtern
LCD_data (Instration); Daten auf Port schreiben LW
Wait (Zeit); Warten

Dieses Programm sendet Daten, die angezeigt werden müssen im 4 Bit Interface zum Display.
Das heisst die Daten werden im 8 Bit Modus der Funktion übergeben. In der Funktion werden die Daten in Hi und LW aufgeteilt und nacheinander zum LCD gesendet.

Auswahl der Zeile und Spalte auf dem LCD



Mit diesem Unterprogramm kann man die Zeile und Spalte auf dem Display anwählen. Diese Funktion zählt den Textbuffer hoch, bis der Cursor auf der ausgewählten Stelle steht.

Anzeigen von Datensätzen auf dem Display

Dieses Unterprogramm ist für die Positionierung des Textes und Anzeigen der Daten auf dem Display. Der Text wird in "Anführungszeichen" und eine Variable wird mit der Adresse (&VariableName) übergeben. Die Funktion schreibt auf das Display, bis der String fertig ist.