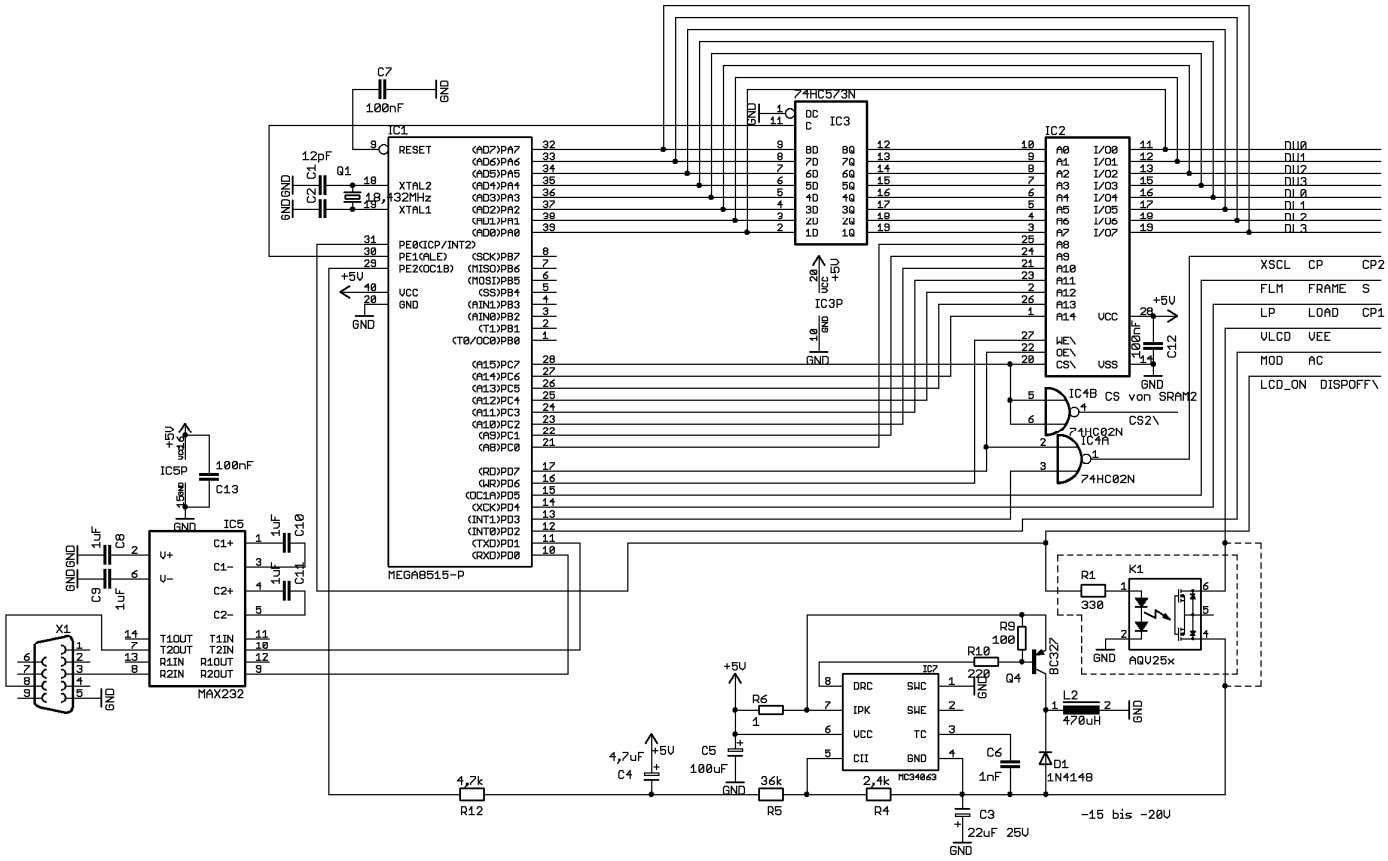
640x480 LCD Controller

- LCD Controller für ein 640x480 Dualscan LCD.
- Schnittstelle: UART, 115200Baud
- Optimiert für Textausgabe
- vollständiger 8x12 Zeichensatz
- 80x40 Zeichen
- Grafik möglich

Befehl	Parameter	Beschreibung
0		NOP (nichts machen)
1		Cursor auf (0,0) setzen
2		LCD (Kontrast Spannung) aus
3		LCD (Kontrast Spannung) an
8		Backspace
9	X Low X High Y Low Y High	Set Pixel (x,y)
10		Cursor eine Zeile nach unten
11		Zeichen an Cursorposition löschen
12		Komplettes Bild löschen
13		Cursor an Zeilenanfang setzen
14		Nicht verwenden !
15	Kontrast	Kontrast einstellen (0-100)
16	X Y Low Y High XS YS Low YS High Daten…	Bild an Position (x,y) mit Auflösung (xs,ys) laden. Auf diesen Befehl folgen xs*ys Bytes mit Bilddaten. x zählt die Nibbles und hat daher den Bereich 0-159, xs zählt Bytes und hat daher den Bereich 1-80.
17	X Y	Textcursor auf (x,y) setzen
18	X1 Low X1 High X2 Low X2 High	Linie von (x1,y1) nach (x2,y2) zeichnen
21		Nicht verwenden !
22		Nicht verwenden !
25	Nummer Zeile 1 Zeile 2 … Zeile 8	Benutzerdefiniertes Zeichen an die entsprechende Position laden laden. Jedes Zeichen besteht aus 8 Zeilen (Bytes) zu je 8 Pixeln (Bits).
28		Nicht verwenden !
30	Ziel Wert	Zeichen aus erweitertem Zeichensatz schreiben. Ziel = 0: Benutzerdefiniertes Zeichen 0-7 schreiben Ziel > 0: ASCII Zeichen 0-255 schreiben.
32-255		Buchstaben an Cursorposition zeichnen, Cursor erhöhen.



Der OptoMOSFET K1 (bzw. der gestrichelte Bereich) kann bei LCDs die über einen ONOFF oder DispOFF\ Pin verfügen entfallen und durch die gestrichelte Verbindung ersetzt werden. Falls der OptoMOSFET nicht erhältlich ist, kann dieser durch einen normalen Optokoppler ersetzt werden, wenn dieser genügend Strom für das LCD schalten kann (Stromverstärkung/Übertragungsverhältnis beachten!).

R5/R4 stellen die Spannung für das Display und somit den Kontrast ein. Der Kontrast lässt sich über einen kleinen Bereich per Software über PWM regeln. Sollte dieser Bereich für das LCD nicht passen, müssen R5/R4 angepasst werden.

IC4B generiert aus dem RD\ Impuls und dem Enable Signal (PD3) die Shiftclock Impulse für das LCD. IC4A invertiert das Chipselect Signal für den 2. 32kByte SRAM. Alternativ kann natürlich auch ein 64kByte SRAM verwendet werden, bei dem der Inverter entfallen kann, da A15 als normale Adressleitung genutzt wird. CS\ wird dann dauerhaft an GND gelegt.

Hinweise zur Ansteuerung:
Aufgrund der teilweise ziemlich rechenintensiven Grafikbefehle sollte der RTS/Busy Ausgang vor dem Senden eines Befehls abgefragt werden. Der Controller verfügt zwar über einen 256 Byte Befehlspuffer, aber auch dieser ist irgendwann voll, wenn mehrere aufwendige Befehle ohne Pause gesendet werden! Solange der RTS/Busy Pin Low ist, kann man gefahrlos mindestens 64 Bytes senden, da RTS/Busy aktiviert wird, wenn der Puffer zu ¾ gefüllt ist. Man braucht also nicht vor jedem Byte RTS/Busy zu prüfen, sondern es reicht einmal vor jedem Befehl (falls dieser kürzer als 64 Bytes ist.)
Bei Verwendung eines vollständigen UARTs reicht es dafür das CTS Handshaking zu aktivieren, die Schnittstelle erledigt dann den Rest.

Die Fusebits müssen auf externen Quarz gestellt werden. Und da die Frequenz >8MHz ist, CKOPT nicht vergessen zu setzen! Um einen sicheren Betrieb zu gewährleisten sollte man auch die BOD auf 4V einschalten. Dann schaltet der AVR nämlich ab wenn die Spannung zu klein wird, was wiederum die Displayspannung über den Optomosfet abschaltet. Das ganze sieht dann in etwa so aus:

Ein paar Worte zum SRAM Timing:

Das Timing hängt hauptsächlich von der "OE\ Access Time" des SRAMs ab:
Im Datenblatt vom ATmega8515, Seite 202, Tabelle 98, Punkt 10:
Read Low to Data valid: max 1.0*tclcl-50ns. Bei 16MHz ist tclcl=62,5ns.
Das SRAM darf daher maximal 12,5ns brauchen zwischen dem Anlegen des RD Impulses bis die Daten stabil sein müssen.
Das von mir gerne verwendete IS61C256 Cache SRAM hat in der langsamsten 25ns Ausführung hier nur 9ns. Es ist also ausreichend schnell.
Bei der Low Power Variante IS62C256 dagegen, hat selbst die 45ns Variante hier 25ns. Dies kann funktionieren, da die Werte jeweils die garantierten Maximalwerte sind, muss aber nicht.

Zusätzlich ist natürlich noch die Zugriffszeit wichtig:
Das wäre Punkt 5 im Datenblatt: Address valid to RD Low: 1.0*tctc-10ns.
Insgesamt ergibt sich dadurch eine Zeit von 2.0*tctc-20ns, also 105ns zwischen Adresse gültig bis Daten stabil, was eigentlich alle SRAMs die obige Bedingung erfüllen, auch erfüllen.

Mit viel Glück funktionieren aber selbst 100ns SRAMs ohne waitstates. Trotzdem sollte man möglichst SRAMs mit 55ns oder weniger verwenden.

Das Latch ist da meiner Meinung nach absolut unkritisch, da die Zugriffszeit ja im vergleich zu der OE Zeit des Speichers extrem groß ist. Ich verstehe bis heute nicht, warum Atmel da ein AC573 empfiehlt. Daher verwende ich auch nur HC02 und HC573.