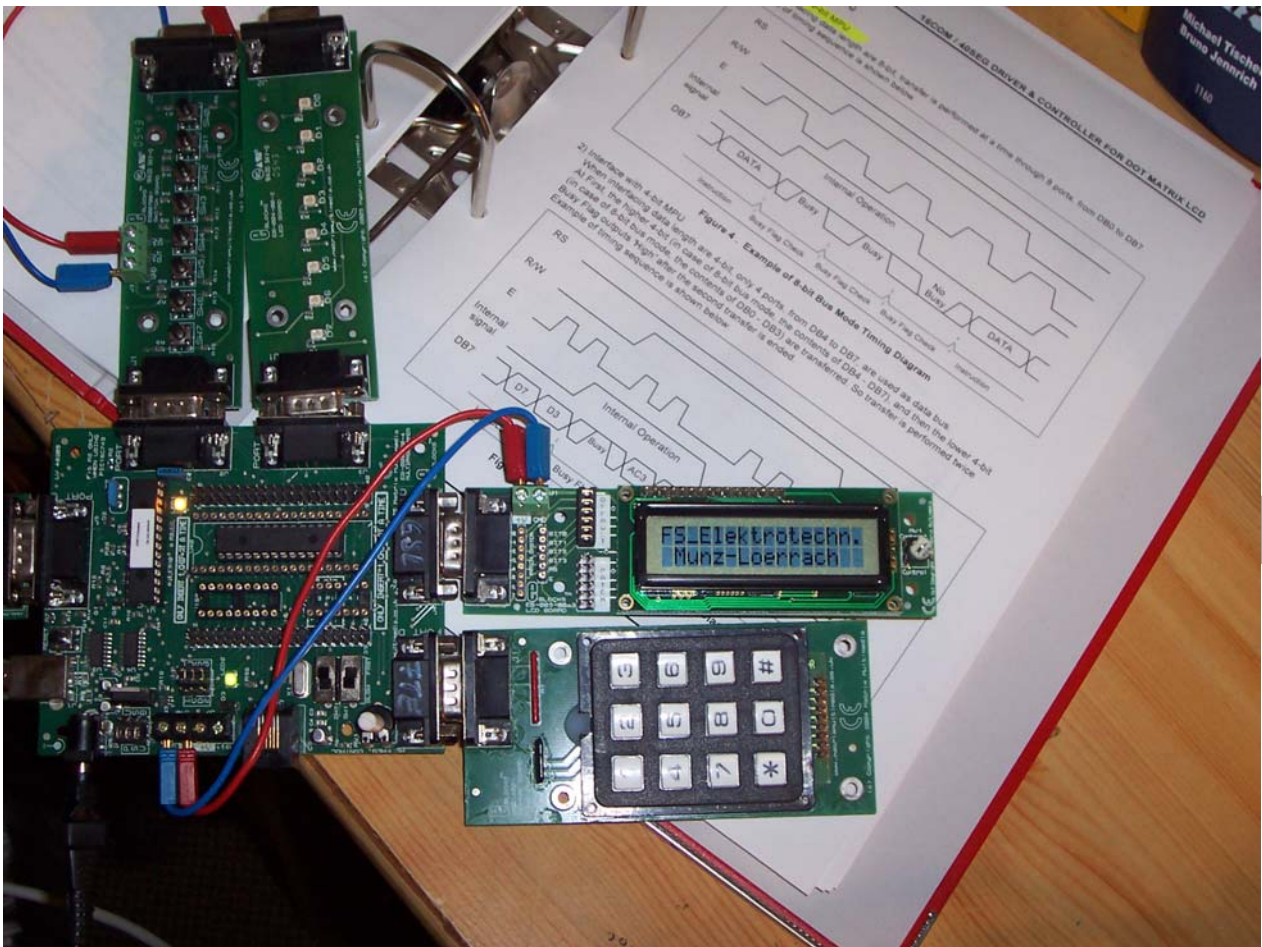


Ansteuern von LC-Displays - <http://de.wikipedia.org/wiki/FI%C3%BCssigkristallbildschirm>



Unter folgendem Link findet man ein 2x16 Display mit einem KS0066U Controller von Samsung

<http://www.femacorp.com/Products/LCD/Character%20Modules/Character%20Modules%20Index.htm>

• 16 Characters Per Line Display



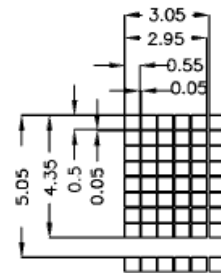
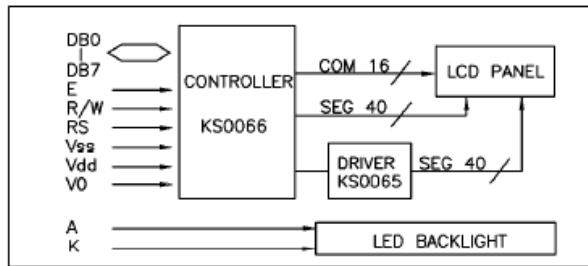
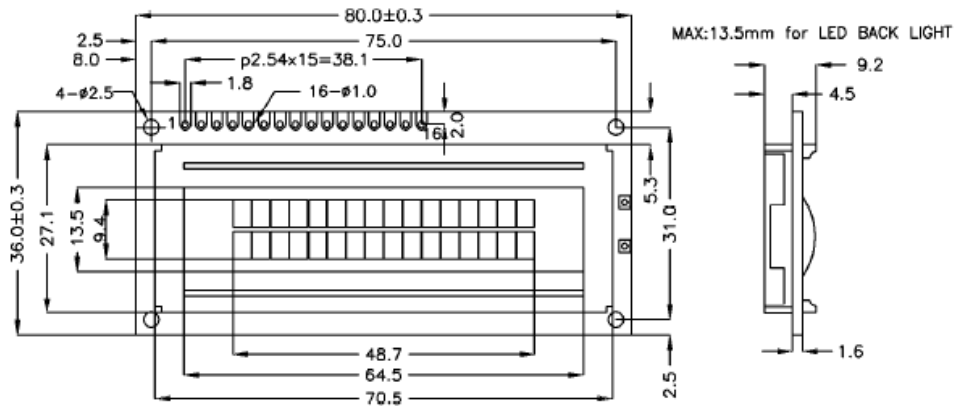
	Fema P/N	Module Size (mm)		Viewing Area (mm)		Digit Height	Controller
		(W)	(H)	(W)	(H)		
16 X 1	CM161B	61.0	30.0	45.0	12.0	5.41	KS0066 or Equiv.
	• CM161C	154.4	43.5	119.4	18.7	12.7	KS0066 or Equiv.
	• CM161D	80	36	64.5	13.5	6.56	KS0066U or Equiv.
16 X 2	• CM1616	122.0	33.0	99.0	13.0	9.22	KS0066 or Equiv.
	• CM1623	85.0	30	64.5	16.4	4.89	KS0066 or Equiv.
	• CM1628	84	44	66	16	5.55	KS0066 or Equiv.
	• CM162LO	80	36	64.5	13.5	4.35	KS0066U or Equiv.
	• CM162R	72	36	61	15.5	5.56	KS0066U or Equiv.
	• CM162W	122	44	99	24	9.66	KS0066U or Equiv.
16 X 4	• CM162P	59.0	29.3	52.0	15.0	4.67	KS0066 or Equiv.
	• CM1644	87	60	62	26	4.75	KS0066U or Equiv.

FEMA ELECTRONICS CORPORATION CM 162LO SERIES

**16 CHAR.
2 LINES**

OPTIONS

- * LCD PANEL
TN or STN
- * View Angle
12H or 6H
- * Character
Eng/Jan
Eng/Rus
Eng/Eur



All dimensions in mm. Tolerance +0.3mm unless otherwise specified.

Das Datenblatt des Controllers (wir brauchen es für die Ansteuerung) findet man hier.

<http://tools.asix.net/download/pvk40/ks0066u.pdf>

Table 5. Relationship between Character Code (DDRAM) and Character Pattern (CGRAM)

Character Code (DDRAM data)								CGRAM Address						CGRAM Data								Pattern number
D7	D6	D5	D4	D3	D2	D1	D0	A5	A4	A3	A2	A1	A0	P7	P6	P5	P4	P3	P2	P1	P0	
0	0	0	0	x	0	0	0	0	0	0	0	0	0	x	x	x	0	1	1	1	0	pattern 1
											0	0	1				1	0	0	0	1	
											0	1	0				1	0	0	0	1	
											0	1	1				1	1	1	1	1	
											1	0	0				1	0	0	0	1	
											1	0	1				1	0	0	0	1	
											1	1	0				1	0	0	0	1	
											1	1	1				0	0	0	0	0	

CGRAM = Character Generator RAM
DDRAM = Display Data RAM

M	u	n	z		A	K	-	T	I	-	N	W	T

INTERFACE WITH MPU

1) Interface with 8-bit MPU

When interfacing data length are 8-bit, transfer is performed at a time through 8 ports, from DB0 to DB7. Example of timing sequence is shown below.

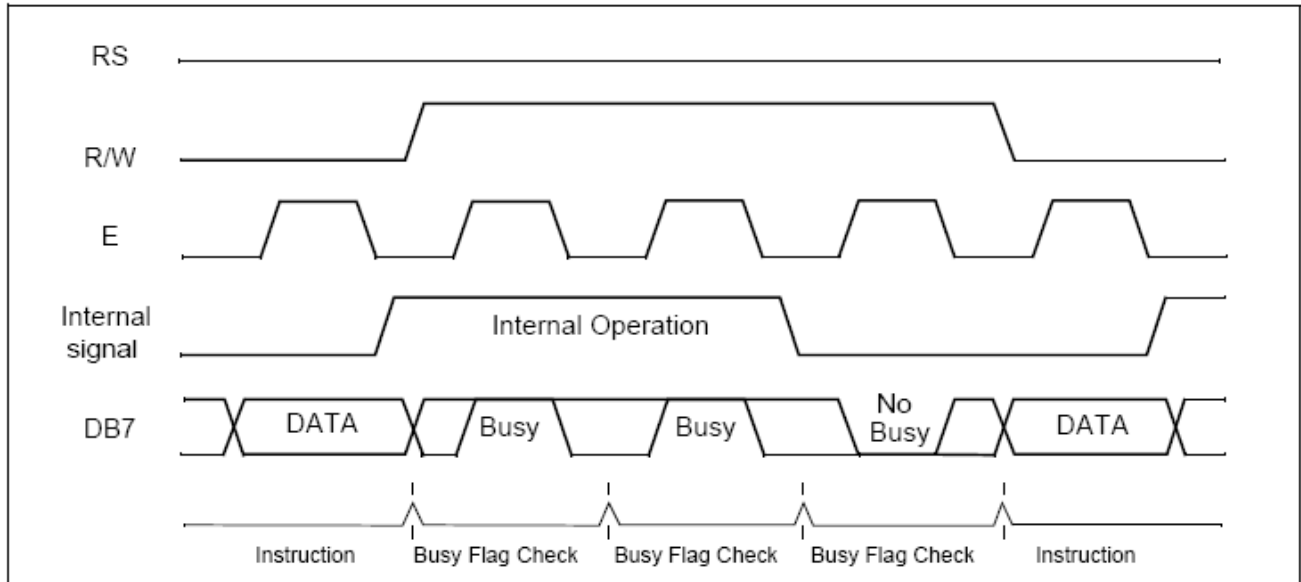
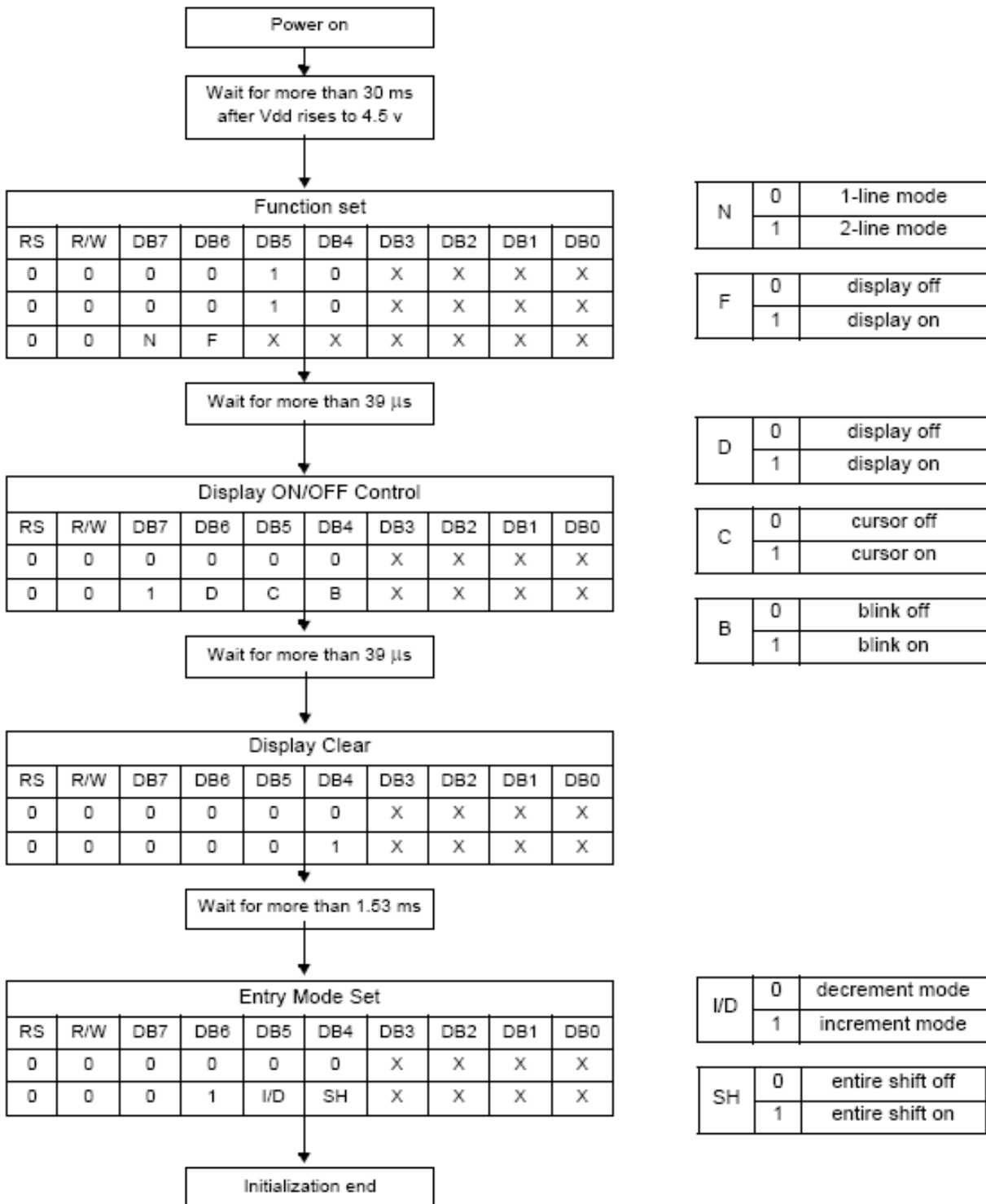


Figure 4 . Example of 8-bit Bus Mode Timing Diagram

2) 4-bit interface mode (Condition: fosc = 270KHZ)



M	u	n	z		A	K	-	T	I	-	N	W	T
---	---	---	---	--	---	---	---	---	---	---	---	---	---

zur Demonstration verwenden wir folgendes Display.



- 1) 9-poliger Sub-D-Stecker
- 2) Schraubklemmen für Stromversorgung
- 3) Patch-Feld (Default belegt die niedrigsten 5 Bit eines Ports)
- 4) 2 6fache Jumper-Feld
- 5) LC-Display
- 6) Potentiometer für LCD-Kontrast

Es hat auch ein KS0066U Controller, der aber weitgehend dem bekannten HD44780-Controller von Hitachi entspricht.

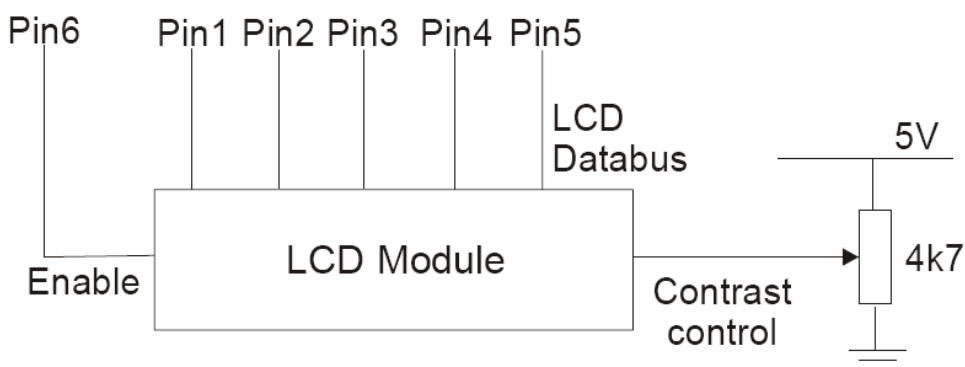
Programmieranleitung Das LCD-Modul ist vom Typ „16 Zeichen x 2 Zeilen“. „Intern“ verfügt es allerdings über 40 Zeichen in 2 Zeilen. Zeile 1 geht von H'00' bis H'27' und Zeile 2 geht von H'40' bis H'67'.

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67

Display Window [16x2] Internal [40x2]

Als Dateneingang werden die Pins 1 bis 6 des 9-poligen Sub-D-Steckverbinders genutzt (in DEFAULT-Einstellung, siehe Bild), um das LCD-Modul zu programmieren bzw. um Daten zu übertragen. **Nach dem Einschalten können Daten erst nach Ablauf von 30 ms übertragen werden – so lange braucht das Modul zur Initialisierung.** ContrastcontrolLCD ModulePin4Pin3Pin6Pin5Pin2Pin1LCD

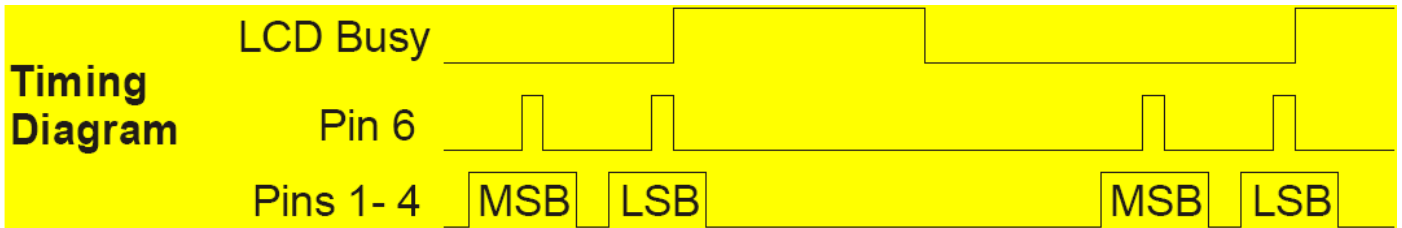
To 9-way D-type Connector



LCD Block Diagram

M	u	n	z		A	K	-	T	I	-	N	W	T

Ein Datum (1 Byte) wird in einem zweistufigen Verfahren an das LCD-Modul geschickt: Das MSB wird vom LSB gefolgt [die Daten liegen an den Pins 1 bis 4]. Nach der Übertragung jedes Nibbles muss B5 auf logisch 1 gelegt werden, um die Datenübertragung abzuschließen. Nach 2 Nibbles ist 1 Byte übertragen. Nach dem zweiten Nibble wird das Kommando ausgeführt. Das sendende Board muss nun mindestens die Zeit warten, bis das Kommando verarbeitet wurde. Erst dann kann es ein neues Datum schicken. Nachfolgend ein Zeit-Diagramm:



→ Also immer zuerst High-Nibble übertragen, dann Low-Nibble.

Das erste an das LCD zu sendende Kommando ist „**Function Set**“ [Setup des LCDs], **was üblicherweise von „Display Control“ und dann „Clear Display“ gefolgt wird.** Gemäß dem kommando „Entry Mode Set“ wird nach jedem an das LCD zu Darstellungszwecken gesendete Zeichen die Cursor-Position auf den neun Stand gebracht.

Displayfunktionen

Displayanschluss an Port C

C0-C3 = Datenbus

C4=RS-Register Select

C5=E=Enable

Initialisierung 1

- 1)0x22 -initialises the LC-display
- 2)0x2C -display in 2 lines
- 3)0x0C -display on, cursor off, blink off
- 4)0x01 -clear the display
- 5)0x06 -increment mode, shift off

Initialisierung 2 falls obige nicht funktioniert

- 1)0x33
- 2)0x32
- 3)0x2C
- 4)0x0C
- 5)0x01
- 6)0x06

//LCD-Kommunikation – d.h. - Befehlsübernahme

```
void lcd_com (void)
```

```
{
enable = 1;
Delay10KTCYx(10); //Verzoegerung 10ms
enable = 0; // PIN 6 – Bit 5 – Enable Toggeln wir – 1-0-1-0..... s.oben
Delay10KTCYx(10); //Verzoegerung 10ms
}
```

Das RS-Register Select Signal muss auf Low sein und das Ist es bei uns.

M	u	n	z		A	K	-	T	I	-	N	W	T

//LCD-Initialisierung

```
void lcd_ini (void)
{
Delay10KTCYx(5); //Verzoegerung 50ms - Selbstinit Controller s. flowchart
display = 0x03;
lcd_com();
display = 0x03;
lcd_com();

display = 0x03;
....
usw.
#####
```

//Zeilen ansprechen

```
void switch_z1 (void)
{
display = 0x08;
lcd_com();
display = 0x00;
lcd_com();
Delay10KTCYx(5); //Verzoegerung 50ms
}
```

```
void switch_z2 (void)
{
display = 0x0C;
lcd_com();
display = 0x00;
lcd_com();
Delay10KTCYx(5); //Verzoegerung 50ms
}
```

hier wird auch RS auf Null gezogen.

#####

```
display = 0x14; //Ausgabe eines Zeichens (H)
lcd_com();
display = 0x18;
lcd_com();
*/
```

Das Zeichen H hat folgenden CODE (s. nächste Seite) 0100 1000 → 0x48. Wir müssen immer zuerst das Higher-Nibble senden, dann das Lower-Nibble. D.h. zuerst die 0x04 und dann die 0x08.

Das RS (Register – Select) Signal PIN5 – Bit 4 müssen wir auf 1 ziehen, was wir hier auch tun. Deswegen 0x14 anstatt 0x04.

#####

// Text ausgeben

```
void show_text (char *ptr)
{
while (*ptr)
{
display = (*ptr/0x10) | 0x10; // Oberes Nibble des Zeichen-Bytes wird um
//4 Stellen nach rechts geschoben (geteilt durch 16) und anschließend
//oder-verknüpft, um das erforderliche Steuersignal (RS) zu erzeugen
lcd_com(); // Ausgabe
.....
```

http://en.wikipedia.org/wiki/Arithmetic_shift
http://irc.essex.ac.uk/www.iota-six.co.uk/c/e5_bitwise_shift_operators.asp

LCD Instruction Set

Instruction	Code					Description	Execution Time
	MSB LSB	B4	B3	B2	B1		
Clear Display	0	0	0	0	0	Clear all display data. Set DDRAM address to 0. Move cursor to home position. Entry mode set to increment.	1.53 ms
Return Home	0	0	0	0	0	Set DDRAM address to 0. Move cursor to home position.	1.53 ms
Entry Mode Set	0	0	0	0	0	Sets cursor move direction (I/D), specifies to shift the display (S). These operations are performed during data read/write.	39 us
		0	1	I/D	SH		
Display Control	0	0	0	0	0	D is Display ON/OFF bit, C is Cursor ON/OFF bit, B is Blink Cursor ON/OFF bit.	39 us
		1	D	C	B		
Cursor/Display Shift	0	0	0	0	1	Sets cursor-move or display-shift (S/C), shift direction (R/L). DDRAM contents remains unchanged.	39 us
		S/C	R/L	X	X		
Function Set	0	0	0	1	0	Configuration data for setting up LCD. [Send First]	39 us
		1	0	X	X		
Set CGRAM Address	0	0	1	A5	A4	Sets the CGRAM address, CGRAM data is sent and received after this setting.	39 us
		A3	A2	A1	A0		
Set DDRAM Address	0	1	A6	A5	A4	Sets the DDRAM address. DDRAM data is sent and received after this setting.	39 us
		A3	A2	A1	A0		
Write Data to RAM	1	D7	D6	D5	D4	Writes data to CGRAM or DDRAM.	43 us
		D3	D2	D1	D0		

DDRAM is Display Data RAM
 DDRAM address is location of cursor
 CGRAM is Character Generator RAM
 X is Don t Care

Bit Name	0	1
I/D	Decrement cursor position	Increment cursor position
SH	No display shift	Display shift
D	Display off	Display on
C	Cursor off	Cursor on
B	Cursor blink off	Cursor blink on
S/C	Move cursor	Shift display
R/L	Shift left	Shift right

LCD Character Set

Higher Lowest Addr	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	0	1	2	3	4	5	6	7	8	9	A	B	C
xxxx0001	!	!	!	!	!	!	!	!	!	!	!	!	!
xxxx0010	"	2	B	R	b	r	!	!	!	!	!	!	!
xxxx0011	#	3	C	S	c	s	!	!	!	!	!	!	!
xxxx0100	\$	4	D	T	d	t	!	!	!	!	!	!	!
xxxx0101	%	5	E	U	e	u	!	!	!	!	!	!	!
xxxx0110	&	6	F	V	f	v	!	!	!	!	!	!	!
xxxx0111	'	7	G	W	g	w	!	!	!	!	!	!	!
xxxx1000	<	8	H	X	h	x	!	!	!	!	!	!	!
xxxx1001	>	9	I	Y	i	y	!	!	!	!	!	!	!
xxxx1010	*	:	J	Z	j	z	!	!	!	!	!	!	!
xxxx1011	+	;	K	[k	[!	!	!	!	!	!	!
xxxx1100	,	<	L	¥	l	!	!	!	!	!	!	!	!
xxxx1101	-	=	M]	m]	!	!	!	!	!	!	!
xxxx1110	.	>	N	^	n	^	!	!	!	!	!	!	!
xxxx1111	/	?	0	_	o	_	!	!	!	!	!	!	!

