

LCD-Modul 2x8 - 5.01mm

INCL. CONTROLLER ST7066

*no more mounting
required*



Dimension 40 x 20 mm

FEATURES

- * SUPER SMALL LCD MODULE
- * HIGH CONTRAST LCD-SUPERTWIST DISPLAY (BLUE/NEUTRAL)
- * OPTIONAL LED-BACKLIGHT YELLOW/GREEN
- * HD 44780 COMPATIBLE
- * 4- OR 8-BIT DATA BUS INTERFACE
- * ASCII CHARACTER SET BUILT IN
- * ALSO AVAILABLE: 1x8 LCD WITH SAME DIMENSIONS
- * POWER SUPPLY +5V OR $\pm 2.7V$ OR $\pm 3.3V$ @2mA and 50mA for LED-B/L (5V)
- * OPERATING TEMPERATURE RANGE EA DIPS082: -20...+70°C
- * OPERATING TEMPERATURE RANGE EA 8081-A3N: 0...+50°C
- * NO MORE MOUNTING REQUIRED: SIMPLY PLUG INTO PCB

ORDERING INFORMATION

LCD-MODULE 2x8 - 5.01mm
WITH LED BACKLIGHT YELLOW/GREEN
LCD-MODULE 1x8 - 7.15 mm

EA DIPS082-HN
EA DIPS082-HNLED
EA 8081-A3N

Pinout

Pin	Symbol	Level	Function	Pin	Symbol	Level	Function
1	VSS	L	Power Supply 0V (GND)	8	D1	H/L	Display Data
2	VDD	H	Power Supply +5V	9	D2	H/L	Display Data
3	VEE	-	Contrast (about 0.3V / 1.2V)	10	D3	H/L	Display Data
4	RS	H/L	H=Data / L=Command	11	D4 (D0)	H/L	Display Data
5	R/W	H/L	H=Read / L=Write	12	D5 (D1)	H/L	Display Data
6	E	H	Enable (falling edge)	13	D6 (D2)	H/L	Display Data
7	D0	H/L	Display Data / Anode LED-B/L	14	D7 (D3)	H/L	Display Data, MSB

LED Backlight

Standard display EA DIPS082-HN is reflective, non-backlighted version. Module with part number EA DIPS082-HNLED comes with yellow/green LED backlight. Power consumption for backlight is 50mA typ. and 80mA max. Backlight is permanent switched on.

For individual use LED backlight can be switched on and off after doing the following modification: Remove series resistor R5 and change resistor from R6 to R7. Now a positive voltage at pin 7 (D0) powers Anode of backlight direct. To limit LED-current an external series resistor is required ($R_{Ext.} = 0,8V / I_{LED}$). Please note that in this case display interface is 4-bit mode only !

Table of commands

Instruction	Code										Description	Execute Time (max.)
	RS	R/W	DB 7	DB 6	DB 5	DB 4	DB 3	DB 2	DB 1	DB 0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears all display and returns the cursor to the home position (Address 0).	1.64ms
Cursor At Home	0	0	0	0	0	0	0	0	1	*	Returns the Cursor to the home position (Address 0). Also returns the display being shifted to the original position. DD RAM contents remain unchanged.	1.64ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets the Cursor move direction and specifies or not to shift the display. These operation are performed during data write and read.	40µs
Display On/Off Control	0	0	0	0	0	0	1	D	C	B	Sets ON/OFF of all display (D) cursor ON/OFF (C), and blink of cursor position character (B).	40µs
Cursor / Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the Cursor and shifts the display without changing DD RAM contents.	40µs
Function Set	0	0	0	0	1	DL	N	F	*	*	Sets interface data length (DL) number of display lines (N) and character font (F).	40µs
CG RAM Address Set	0	0	0	1	ACG					Sets the CG RAM address. CG RAM data is sent and received after this setting.		40µs
DD RAM Address Set	0	0	1	ADD					Sets the DD RAM address. DD RAM data is sent and received after this setting.		40µs	
Busy Flag / Address Read	0	1	BF	AC					Reads Busy flag (BF) indicating internal operation is being performed and reads address counter contents.		-	
CG RAM / DD RAM Data write	1	0	Write Data					Writes data into DD RAM or CG RAM		40µs		
CG RAM / DD RAM Data Read	1	1	Read Data					Reads data from DD RAM or CG RAM		40µs		

2x8 - 5.01mm

Character set

Below shown character set is already built in. Additionally 8 self defined characters can be attached.

Lower 4 bit \ Upper 4 bit	0000 (\$0x)	0010 (\$2x)	0011 (\$3x)	0100 (\$4x)	0101 (\$5x)	0110 (\$6x)	0111 (\$7x)	1010 (\$Ax)	1011 (\$Bx)	1100 (\$Cx)	1101 (\$Dx)	1110 (\$Ex)	1111 (\$Fx)
xxxx0000 (\$x0)	CG RAM (0)	0	@	P	`	F		-	9	ε	α	ρ	
xxxx0001 (\$x1)	(1)	!	1	A	Q	a	q	μ	7	φ	Δ	ä	q
xxxx0010 (\$x2)	(2)	"	2	B	R	b	r	Γ	ι	ψ	χ	β	θ
xxxx0011 (\$x3)	(3)	#	3	C	S	c	s	∟	υ	τ	ε	ε	∞
xxxx0100 (\$x4)	(4)	\$	4	D	T	d	t	∖	I	†	‡	μ	Ω
xxxx0101 (\$x5)	(5)	%	5	E	U	e	u	·	†	†	∫	ε	Ü
xxxx0110 (\$x6)	(6)	&	6	F	V	f	v	☞	カ	ニ	ヨ	ρ	Σ
xxxx0111 (\$x7)	(7)	'	7	G	W	g	w	ア	キ	ヌ	ウ	g	π
xxxx1000 (\$x8)	CG RAM (0)	(8	H	X	h	x	イ	ウ	ネ	リ	フ	×
xxxx1001 (\$x9)	(1))	9	I	Y	i	y	ウ	ケ	ノ	ル	フ	γ
xxxx1010 (\$xA)	(2)	*	:	J	Z	j	z	エ	コ	ハ	レ	j	≠
xxxx1011 (\$xB)	(3)	+	;	K	[k	[ス	サ	ヒ	ロ	*	≠
xxxx1100 (\$xC)	(4)	,	<	L	¥	l	l	ヤ	シ	フ	ワ	φ	≠
xxxx1101 (\$xD)	(5)	-	=	M	∫	m	}	ユ	ズ	ハ	ン	ε	÷
xxxx1110 (\$xE)	(6)	.	>	N	^	n	→	ヨ	セ	ホ	ハ	ñ	
xxxx1111 (\$xF)	(7)	/	?	O	_	o	←	ウ	リ	マ	"	ö	■

CREATING YOUR OWN CHARACTERS

All character display modules offered in this catalogue, are able to create 8 own characters (ASCII Codes 0..7) in addition to the 192 ROM fixed codes.

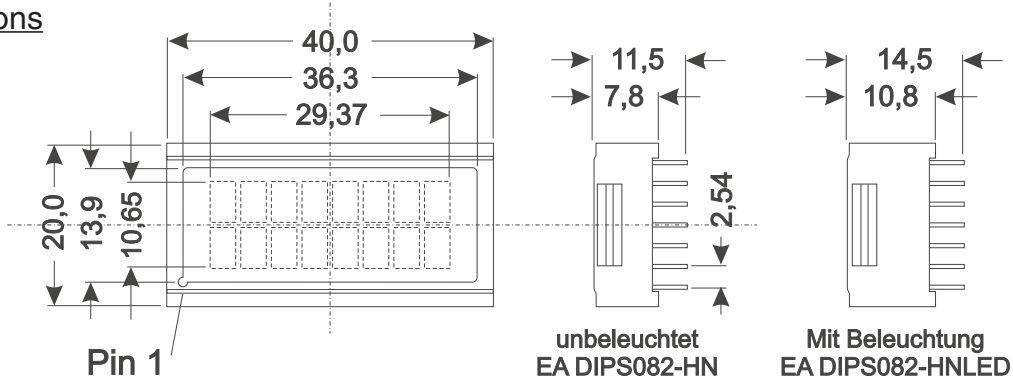
- 1.) The command "CG RAM Address Set" defines the ASCII code (Bit 3,4,5) and the dot line (Bit 0,1,2) of the new character. Example demonstrates creating ASCII code \$00.
- 2.) Doing 8 times the write command "Data Write" defines line by line the new character. 8th. byte stands for the cursor line.
- 3.) The new defined character can be used as a "normal" ASCII code (0..7); use with "DD RAM Address Set" and "Data Write".

Adresse im CG RAM setzen					Daten des Zeichens												
Adresse		Hex	Bit														
7	6	5	4		3	2	1	0	Hex								
0	1	0	0	0	0	0	0	\$40	X	X	X	0	0	1	0	0	\$04
				0	0	1	0	0				\$04					
				0	1	0	0	\$42				0	0	1	0	0	\$04
				0	1	1	0	\$43				0	0	1	0	0	\$04
				1	0	0	0	\$44				1	0	1	0	1	\$15
				1	0	1	0	\$45				0	1	1	1	0	\$0E
				1	1	0	0	\$46				0	0	1	0	0	\$04
				1	1	1	0	\$47				0	0	0	0	0	\$00

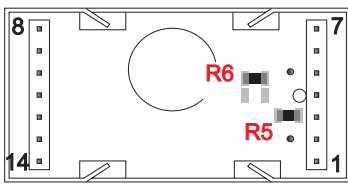
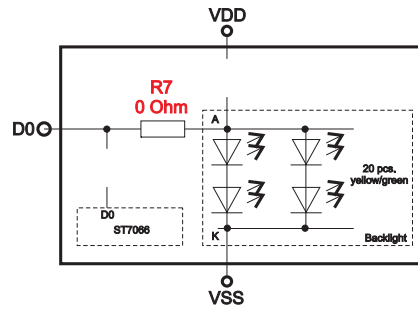
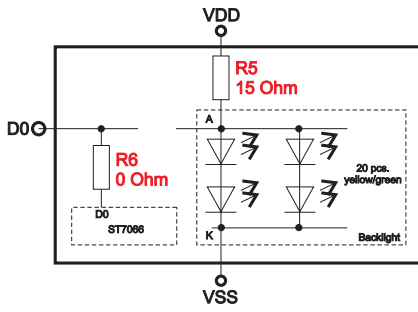
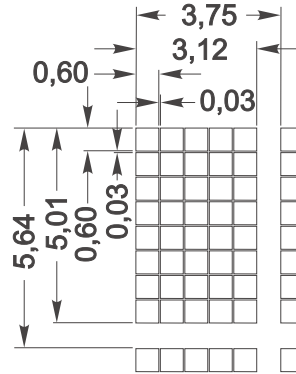
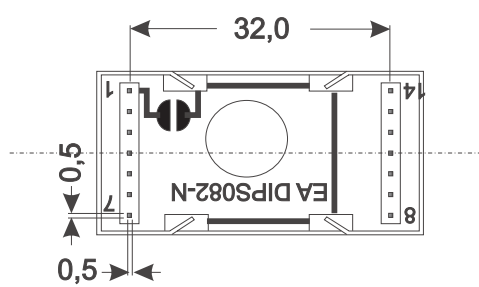
INITIALISATION FOR A 2 LINE DISPLAY / 8-BIT MODE										
Command	RS	R/	DB	DB	DB	DB	DB	DB	DB	Remark
Function Set	0	0	0	0	1	1	1	0	0	8-Bit Data Length, 2/4 lines, 5x7 Font
Display ON/OFF	0	0	0	0	0	0	1	1	1	Display on, Cursor visible, Cursor blink
Clear Display	0	0	0	0	0	0	0	0	1	Clear Display, Cursor Home
Entry Mode Set	0	0	0	0	0	0	1	1	0	Cursor Auto-Increment

2x8 - 5.01mm

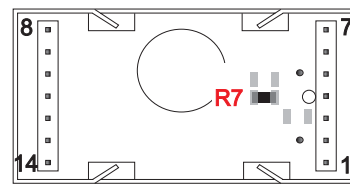
Dimensions



all dimensions are in mm



Delivery state
Intern, LED permanent on



Extern, LED controllable,
only possible in 4-bit mode

1-line Display EA 8081-A3N

