

LC DESIGN

*Analogue Data Display  
Unit  
LC240240ADD  
V3.0  
short*

© LC Design

Inh. Dipl. Ing. Hartmut Putzig

Johann-Knecht-Straße 28 • 63785 Obernburg a.M.

Telefon 0 60 22/61 44 30 • Fax 0 60 22/61 44 31

Internet <http://www.LC-Design.de>

e-mail [Support@LC-Design.de](mailto:Support@LC-Design.de)

This document and the information given in connection with, so far is based on LC Design creations, are and remain the property of LC Design. This document and all information are confidential and it is not allowed to make them accessible to third parties without having the previous written consent of LC Design. All rights especially in relation to inventions are reserved.

## Directory

<b>Directory</b>	<b>2</b>
<b>Description</b>	<b>3</b>
<b>Views</b>	<b>4</b>
<b>front view:</b>	<b>4</b>
<b>view from the back:</b>	<b>4</b>
<b>Dimensions</b>	<b>5</b>
<b>Optical characteristics</b>	<b>6</b>
<b>Power supply</b>	<b>7</b>
<b>Pin assignments</b>	<b>7</b>
<b>SPI- interface</b>	<b>7</b>
<b>USB- interface</b>	<b>7</b>
<b>RS232- interface</b>	<b>8</b>
<b>I2C- interface</b>	<b>8</b>
<b>keyboard interface</b>	<b>8</b>
<b>Analogue input</b>	<b>9</b>
<b>USB Controller</b>	<b>9</b>
<b>Keyboard reading</b>	<b>10</b>
<b>Command set</b>	<b>11</b>
<b>Composition of command set</b>	<b>12</b>
<b>Transparency function</b>	<b>13</b>
<b>Loading of pictures</b>	<b>13</b>
<b>Storing and loading pictures into/from flash</b>	<b>13</b>
<b>Display needle indicators</b>	<b>15</b>
<b>Analogue display mode</b>	<b>15</b>
<b>Video display</b>	<b>16</b>
<b>Blend functions</b>	<b>16</b>
<b>Default settings</b>	<b>16</b>
<b>Colour conversion</b>	<b>17</b>
<b>Character fonts</b>	<b>18</b>
<b>Interface selection</b>	<b>19</b>
<b>Notes for UART (RS232), SPI and I<sup>2</sup>C interfaces</b>	<b>20</b>
<b>SPI timing</b>	<b>21</b>
<b>Technical data</b>	<b>24</b>
<b>Precautions</b>	<b>25</b>
<b>Revisions</b>	<b>26</b>

This document and the information given in connection with, so far is based on LC Design creations, are and remain the property of LC Design. This document and all information are confidential and it is not allowed to make them accessible to third parties without having the previous written consent of LC Design. All rights especially in relation to inventions are reserved.

## Description



The analogue data display unit (ADD) consists of a TFT-display with a resolution of 240xRGBx240 dots and a processing unit which can handle different interfaces. USB, RS232, SPI and I2C can be selected.

The unit is powered either via an USB-2.0 – interface or from another 5V source, depending on the selected interface.

The display content can be programmed with a simple command structure to keep the data transfer in the application as low as possible and save. Especially the simulation of an analogue instrument is made easy due to the possibility to load dial and pointer as a picture, while the picture of the position of the pointer in relation to the dial is calculated in the unit without the need to transfer the final picture. Only the value for the desired pointer position and angle is needed.

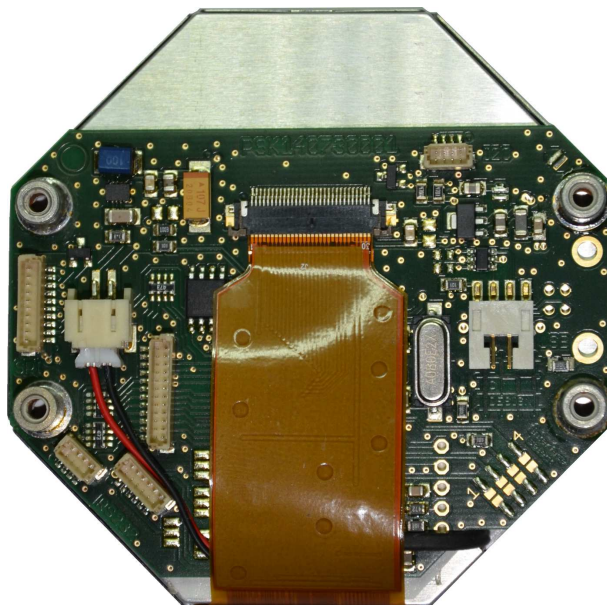
## Views

### front view:

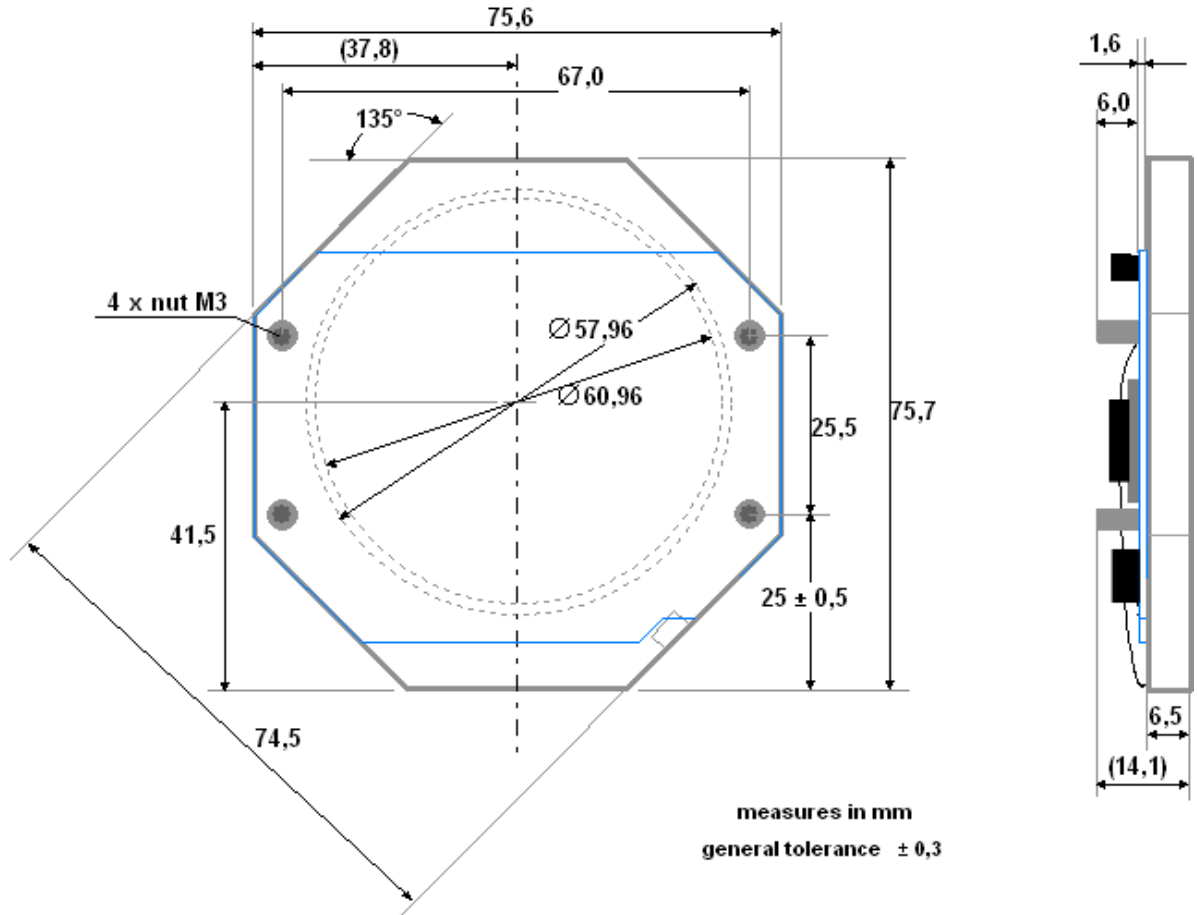


The size of the unit is mainly defined by the size of the display.

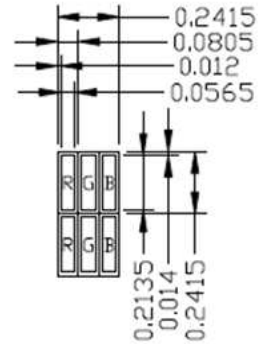
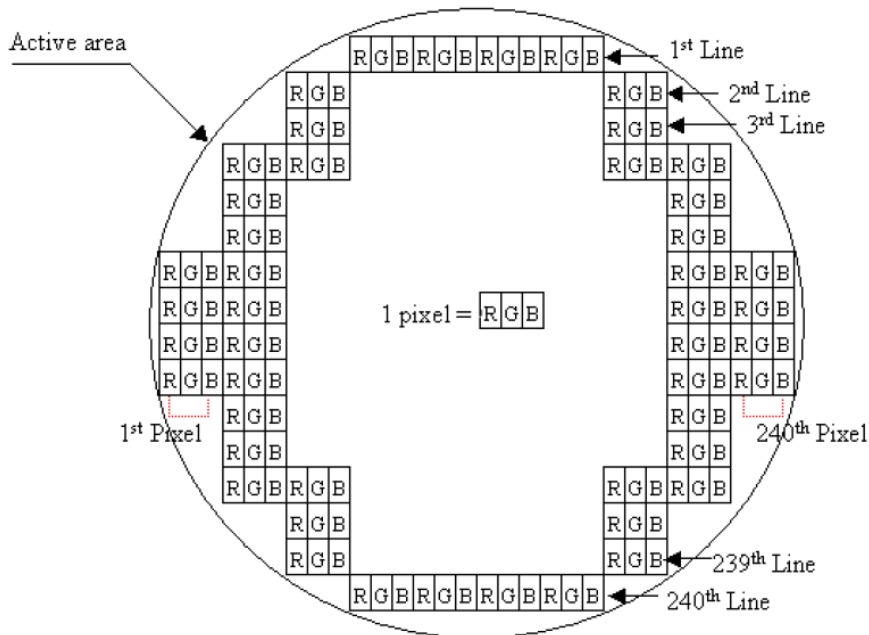
### view from the back:



# Dimensions



Pixel pitch: 0,2235 mm x 0,2235mm

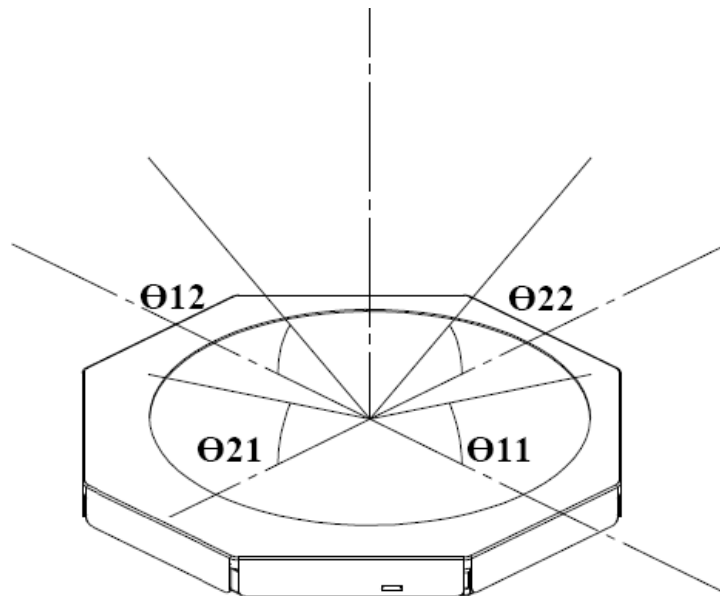


## Optical characteristics

Ta=25°C

Parameter		Symbol	Condition	min.	typ.	max	Unit
Viewing Angle	Horizontal	$\Theta_{11}$ , $\Theta_{12}$	$CR \geq 10$	70	75	-	deg
	Vertical	$\Theta_{21}$		45	50	-	deg
		$\Theta_{22}$		65	70	-	deg
Contrast Ratio		CR	At optimized viewing angle	200	400	-	-
Brightness		L	$\theta = 0^\circ$	800	1.000		cd/m <sup>2</sup>
Uniformity		U%	$\theta = 0^\circ$	70	75	-	%
Response	Rise	Tr	$\theta = 0^\circ$	-	15	20	ms
	Fall	Tf		-	20	30	ms
White Chromaticity		x	$\theta = 0^\circ$	0.29	0.33	0.37	-
		y		0.31	0.35	0.39	-
LED Life Time		-	+25°C	20.000	30.000	-	hrs

The definitions of viewing angles are as follows:



## Power supply

The unit has to be powered with 5V, even all logic levels are on a 3.3V base.

There is no special connector for power supply. For easy use every interface connection includes pins for power supply.

## Pin assignments

### SPI- interface

CON 2 10 pin single row header, 1.00 mm pitch, JST BM10B-SRSS

Pin	Signal	Function
1	V+	5V supply
2	SDI	Serial data input 3.3V level (5V tolerant)*
3	GND	Reference level 0V
4	SDO	Serial data output 3.3V level
5	GND	Reference level 0V
6	SCI	Serial clock input 3.3V level (5V tolerant)*
7	GND	Reference level 0V
8	SS	Slave select input 3.3V level (5V tolerant)*
9	GND	Reference level 0V
10	BUSY	Busy indication output 3.3V level

### USB- interface

CON 4, 4 pin connector, 2.00mm pitch, FCI 95000-004LF  
(CON 7, 4 pin connector, USB-B)

Pin	Signal	Function
1	V+	5V supply
2	USB D -	USB differential signal D -
3	USB D+	USB differential signal D +
4	GND	Reference level 0V

**RS232- interface**

CON 8, 6 pin single row header, 1.00 mm pitch, JST BM06B-SRSS

Pin	Signal	Function
1	V+	5V supply
2	TXD	Output RS232 or 3.3V level
3	RTS	Output RS232 or 3.3V level
4	RXD	Input RS232 or 3.3V level (5V tolerant)*
5	CTS	Input, currently not used RS232 or 3.3V level (5V tolerant)*
6	GND	Reference level 0V

Despite CTS being currently being not used, it should be driven low to indicate data can be transmitted from module to host for compatibility with future versions.

**I2C- interface**

CON 9, 4 pin single row header, 1.00 mm pitch, JST BM04B-SRSS

Pin	Signal	Function
1	V+	5V supply
2	SCL	Serial clock no pull up, 3.3V level (5V tolerant)*
3	SDA	Serial data no pull up, 3.3V level (5V tolerant)*
4	GND	Reference level 0V

**keyboard interface**

CON 5, 12 pin single row header, 1.00 mm pitch, JST BM12B-SRSS

Pin	Signal	Function
1	1	Button top left
2	2	Button top centre
3	4	Button top right
4	8	Button middle left
5	16	Button middle centre
6	32	Button middle right
7	64	Button bottom left
8	128	Button bottom centre
9	256	Button bottom right
10	512	Underline left
11	1024	Underline right
12	GND	Reference level 0V

All buttons switch to ground if pressed

\* Note on 5V tolerance: 5V tolerance is only allowed when powering the display unit. Otherwise the display unit may be damaged!



## Analogue input

CON 10, 4 pin single row header, 1.00 mm pitch, JST BM04B-SRSS

Pin	Signal	Function
1	V+	5V supply
2	GND	Reference level 0V (power GND)
3	input	0-3.3V analogue input
4	GND	Reference level 0V (signal GND)

Note: Despite having a high input impedance ( $>1\text{M}\Omega$ ), it should be driven by rather low impedance source ( $<10\text{k}\Omega$ ) to avoid gain and offset errors.

## USB Controller

The connection to the host will automatically be detected by the host. Afterwards a driver file has to be installed. This will create a virtual COM port with the next available address.

Windows Server 2008  
 Windows Server 2008 x64  
 Windows Vista  
 Windows Vista x64  
 Windows 7  
 Windows 7 x64  
 Windows XP  
 Windows XP x64  
 Windows 2000  
 Windows Server 2003  
 Windows Server 2003 x64 will be supported.

Suspend mode:

The unit will not comply to the suspend mode for USB devices. The backlight will be switched off, but the microcontroller and the TFT display module will remain fully powered. Therefore the current in that mode will be around 50mA to 100mA. For most systems this will not be a problem, except for some battery powered systems. If the powering system (PC) allows a higher current, then, by setting a jumper (JP3), the display backlight can also be enabled to show a logo or anything else at power up, even before USB drivers are running and grant the higher current.

## Keyboard reading

The keyboard reading sends a code packet any time a key is pressed or released or on request depending on the mode, which has been set upfront.

The packet starts with start byte (211), length (4), command (50) followed by the acknowledge code 0 and low and high byte of the actual pressed keys. (Details see command set and command structure.)

0	no key pressed			
1	key 1 pressed	1	2	3
2	key 2 pressed	4	5	6
4	key 3 pressed	7	8	9
8	key 4 pressed			
$2^{(K-1)}$	key K pressed	xx	,	

If more than one key is pressed, the values are added.

e.g. key 1 and key 3 pressed  $\Rightarrow 1 + 4 = 5$

# Command set

command	short form
1	WRITECHAR
2	LINE FEED
3	CARRIAGE RETURN
4	GOTOXY
5	FONT
6	TEXTCOLOR
8	WRITECHARROT
10	SETPIXEL
11	LINE
12	CIRCLE
13	BLOCK
14	LOADPIC
20	BRIGHTNESS
21	CLEAR
22	TRANS
23	INVERT
24	INVERTAREA
30	REINIT
31	RESTART
33	LOADPICFLASH
34	STORESTARTUP
34	STORESTARTUP
35	STOREPICFLASH
36	READPIC
37	COPYPIC
38	LOADPICFLASHROT
39	STOREINDICATOR
40	LOADINDICATOR
41	SOFTINFO
42	HARDINFO
43	LCDINFO
44	LCDSTRING

command	short form
50	GETKEYS
51	AUTOKEYS
52	CLEARKEYS
58	READADC
60	UPDATE
61	AUTOUPDATE
70	ANSWER
75	PICINFO
80	LOADPICFLASHBLEND
81	LOADPICFLASHBLENDBACK
82	BLENDAREA
97	STANDALONEMODE
98	SHOWVIDEO
100	----

In all cases, except picture load, the colour has to be specified as 16bpp, even only less different colours are needed. See down low calculations for colour conversion

\* key scan and display activities cannot be controlled at the same time.

If a command is detected (start byte received), no key stroke is send to the host. The key strokes will be monitored, stored in a buffer of max. 256 entries and

send to the host after the command has been executed. This will typically delay the key stroke by a time, depending of the actual command between a few

usec and a few msec in normal case. If a picture is transferred the delay may be longer.

## Composition of command set

- start byte (211) indicator of communication
- length of the string, command and data, without CRC (max. 64 bytes)
- command see list above
- data
- CRC lowbyte
- CRC highbyte

The CRC has to be calculated including length, command and data.

The CRC-CCITT (CRC-16) with the start value of 0xFFFF and the polynomial  $x^{16} + x^{12} + x^5 + 1$  has to be used.

The response has the same structure.

- start byte (211)
- length of the string, command and data, without CRC
- command returned
- failure code or data
- CRC lowbyte
- CRC highbyte

The code of the command will always be returned followed by a failure code of 1 Byte:

- |     |                                       |
|-----|---------------------------------------|
| 0   | command executed successfully         |
| 1   | timeout                               |
| 2   | size invalid (that means >64 Bytes)   |
| 3   | CRC failure                           |
| 4   | coordinates invalid                   |
| 5   | invalid parameters                    |
| 6   | no picture available                  |
| 7   | number of parameters incorrect        |
| 8   | unknown command                       |
| 128 | picture data successfully transferred |

If a failure occurs which prevents the reception of a valid command code, 0 will be send back as command code (e.g. timeout error, CRC error, packet length error).

After reception of the start byte of a command a timeout of 10 seconds will be started.

Within this 10 seconds the command or all picture data must be fully transferred.

If the transfer is not finished within this time, the transfer will be aborted and timeout error will be send.

## Transparency function

If this function is active, a comparison between colour to be drawn and transparency colour will be carried out with all drawing operations, draw line, draw rectangle, etc.

At identity on colour on any position of the display, the former colour of that pixel will be kept.

This allows e.g. to write a text with a transparent fill-up colour over a picture. The remaining pixels will keep the colour of the former display content. This will also work with pictures.

The comparison is time consuming, because it is made pixel by pixel. This lengthens the drawing operations. Therefore it is recommended to switch off this function if not needed.

## Loading of pictures

Different colour depth can be selected: 16bpp, 8bpp, 4bpp or 1bpp.

- 16bpp uses the standard RGB 5:6:5 bit format. 5bit for red, 6bit for green, 5 bit for blue. Within this format all colour values will be transferred. 2 bytes per pixel have to be transferred.
- 8bpp reduces the amount of data to 1 byte per pixel. The data will be transferred in the 3:3:2 format.
- 4bpp uses the standard 16 colour palette, which means 1 bit per red, green and blue and 1 bit for the intensity.
- 1bpp transfers only 2 colours which are taken from text colour and background colour. With 1 byte, data for 8 pixels can be transferred.

If the amount of bits to be transferred in 4bpp or 1 bpp mode not fits exactly to the number of bytes, which will be transferred, the not used bits per row will be neglected.

e.g. to transfer the data for 60 pixel in the 1bpp format  $60/8 = 7,5$  byte would be needed.

8 bytes will be transferred and 4 bits neglected.

The data in 4bpp and 1bpp format have to be transferred in LSB first configuration. Bit 0 will be the first pixel set in the data string.

## Storing and loading pictures into/from flash

Flash memory has not an unlimited number of write cycles. Therefore the user should avoid unnecessary write cycles and distribute the pictures over all available picture storage locations evenly. Pictures 0-5 are stored in internal flash memory, pictures 6-21 or more, depending on memory size (larger flash optional) are stored in external flash memory.

The internal flash memory of the microcontroller has at least 1000 write cycles per storage location, external flash memory more then 100000. Whenever possible the external memory should be preferred, because it is more durable. The load time of a picture from the external memory is slightly higher than from the internal memory (~50msec instead of ~ 10 msec for a 240x240 picture).

Generally, all picture locations can be used to store images, but some of them have special features available:

- Pictures 6-21 (or more, depending on optional external flash installed) are stored in external flash. This memory can store a large number of images. But the loading time is higher and no special features like image rotation are available. It is best suited for background images.

- Pictures 0-5 are stored in internal flash. The loading time is faster and the pictures can be rotated. Because of limited memory available the images have different sizes: Pictures 0-1 allow fullscreen images, pictures 2-5 only smaller ones.

All pictures stored in internal flash can be used for normal operation. Pictures are stored using the STOREPICFLASH command.

In addition, picture number 0 can also be used for start screen or for standalone indicator mode. For use as start screen a special command is available (STORESTARTUP) that stores special attributes to the picture to make the firmware automatically load it during boot. Since the image is used to fill the background the size is fixed to fullscreen resolution of 240x240. Another mode available to this picture is the analogue display standalone mode. This mode is entered using the same STORESTARTUP command, but with modified parameters. In this mode the image is loaded automatically, but in addition a needle indicator stored in a separate location is added. All remaining picture locations can be used as needle indicators using the STOREINDICATOR command. This stores additional parameters used for needle display using the LOADINDICATOR command or analogue standalone mode.

This results in the following restrictions for loading images:

- All pictures, regardless of the command they were stored with, can be loaded using the LOADPICFLASH command. Addition parameters are ignored and only the image data is used.
- All pictures in internal flash memory can be used with the LOADPICFLASHROT command.
- Only pictures stored with the STOREINDICATOR command can be used with LOADINDICATOR command. For all other ones the LOADPICFLASHROT command has to be used. The main difference between these two commands is the source of the rotation parameters: LOADINDICATOR uses the data stored together with the picture, LOADPICFLASHROT receives the data together with the command.
- Only picture nr 0 can be used for bootscreen or standalone mode

Recommended picture storage location usage:

- start up picture / standalone mode: location 0
- background images: locations 6-21 (or more, depending on optional flash installed)
- small rotated images/needles: locations 2-5
- large rotated images/needles: locations 0-1
- icons: locations 2-21

It is possible to retrieve information of every picture using the PICINFO command:

It not only informs if a picture is present and its resolution but also what additional information is attached to it:

- Pictures that were saved using the STOREPICFLASH command have a type value of 0 if they are saved in internal flash or 4 if they are saved in external.
- Pictures that were saved using the STOREINDICATOR command have a type value of 2.
- Pictures that were saved using the STORESTARTUP command have a type value of 1 if start screen is enabled or 3 if standalone mode is enabled.

This attribute allows the software to decide which LOAD commands the image can be used with.

All images have an ID number attached. This number can be freely assigned by the user during storage. It is not used by the firmware, its only purpose is to identify the image being stored in memory. The user has to assign a unique ID number to each image. The software can read this ID to check if the correct image is available in memory as expected. If the user does not give an ID number, it is set to 0.

## Display needle indicators

Displaying analogue needle indicators (and other rotated objects) is simple and fast because of an extra command which does all the necessary work.

To use this command the user must first set a background image (optional) and the needle picture and parameters and store them into the flash memory like normal pictures. In fact, needle indicators are stored like normal pictures, except that additional parameters are stored together with the picture data and they can not put in external memory. These pictures can be loaded afterwards just like normal pictures, ignoring the additional data.

To display the needle indicator, the software needs to know the following parameters:

- position of needle in display
- rotation axis of needle image
- transparent colour of needle image
- rotation angle for minimum input value
- rotation angle for maximum input value
- minimum input value
- maximum input value

The allowed input value range is 0 to 65535 for maximum resolution of rotation. The user has not to use the full range. He can use any part of it and the needle angle is scaled to fit the given range. Normally the needle moves clockwise with increasing input values, even if maximum angle is lower than minimum angle. For example if the minimum angle is given is 16384 (3 o'clock) and the maximum angle is 0 (12 o'clock), it will move clockwise from 3 o'clock to 12 o'clock. If the angle values are reversed, the needle will move from 12 o'clock to 3 o'clock. To reverse the movement, the input values must be swapped, thus the minimum input value must be larger than the maximum value. The angle values must also be swapped because the maximum input value always corresponds with maximum angle. If the user sets e.g. 100 as minimum value, 16384 as minimum angle and 50 as maximum value, 49152 as maximum angle, the needle will start at 9 o'clock at an input value of 50 and move counter clockwise to 3 o'clock with the input value increasing towards 100.

If the input value is outside the given input range, the needle will stop at the minimum or maximum angle position.

As an alternative to the user supplied input value, the build in analogue input can be used giving an ADC output value in the range of 0-65535 for an analogue input voltage of approximately 0-3.3V.

The needle indicator command uses all these stored parameters and the data given in the command and first loads the background image (if given) with previous transparency settings. Then it calculates the needle angle and draws it with transparency settings stored together with the needle picture and updates the display if autoupdate is activated. Afterwards the old transparency settings are restored.

## Analogue display mode

To simplify analogue data display further, a standalone mode is integrated. Using this mode the user can setup an analogue needle indicator using the supplied PC software and enable the analogue display mode while loading the start screen image. On the next reboot the display unit will automatically load the startup image as background and the given indicator image and use the analogue input voltage to position the needle. The unit will continuously query the ADC value and update the needle position. The normal command interface is still operational, but all changes to the image will be overwritten at the next needle update. Therefore only non image commands should be send like setting backlight brightness, reading keys or storing new needle settings.

You can exit standalone mode by sending the STANDALONE MODE command. This can be useful to start with a default dial and change it later to more specific one using an external microcontroller. You can always reenable the standalone mode.

Note: Display updating is prioritized. Command execution may be slower than in normal mode.

## Video display

It is possible to display a short video sequence using multiple images.

The SHOWVIDEO command loads a series of images using a given time interval. This avoids the need of sending repetitive image load commands at a high rate. In addition to displaying the image sequence it is also possible to loop the sequence a given number of times (1-256x). The framerate is determined by the frametime value. The delay between images is settable in multiples of 10ms. This allows framerates in the range of 100fps (1 for 10ms) down to 0.39fps (0 for 2.56s). Using a 240x240 full image from external flash allows up to 16fps, therefore the minimum possible frame time is 70ms → 7.

The display will be updated with the video data regardless of autoupdate setting.

During video display no other command will be executed neither key values will be sent. But they will be internally buffered and send afterwards. There is no possibility to abort this command.

Note: Using low framerates and a high number of repetitions the execution time of this command can be very long: 16 frames x 2.56s per frame x 256 repetitions = 2.9 hours.

## Blend functions

The blend functions merge two pictures together by using a given percentage of information from the first and the remaining fraction of the second image. This allows to display half transparent images or fading between two images.

When blending the display contents many times in small steps it should be kept in mind that the display has a limited number of displayable colours. This will result in rounding errors and very dark colours will even not fade at all, because they will stick at black because of rounding errors. On the other hand, this can give a nice looking effect if the adequate combination of picture and fading values is used, particularly together with the transparency function.

## Default settings

After power-on the following settings are valid.

- character font	2 (12x22)
- text colour	white
- background colour	black
- transparency	off
- autokey	on
- answers	on
- picture updates	auto
- brightness	50 (=20%) or depending on start screen setting



**Colour conversion**

All colour values have to be transmitted in 16bpp mode, except a picture is transferred.

- Conversion 3x 8bit RGB data to 16bpp:  $((b \& 248) / 8) | ((g \& 252) * 8) | ((r \& 248) * 256)$

- Conversion 3x 8bit RGB data to 8bpp:  $((b \& 192) / 64) | ((g \& 224) / 8) | (r \& 224)$

- Colour table for 4bpp mode (based on 3x8 bit RGB):

( 0, 0, 0)  
 ( 0, 0,170)  
 (0,170, 0)  
 ( 0,170,170)  
 (170, 0, 0)  
 (170, 0,170)  
 (170, 85, 0)  
 (170,170,170)  
 ( 85, 85, 85)  
 ( 85, 85,255)  
 ( 85,255, 85)  
 ( 85,255,255)  
 (255, 85, 85)  
 (255, 85,255)  
 (255,255, 85)  
 (255,255,255)

### Character fonts

Eight different font sizes are available, based on the font in the table down low.

- Font 0: size 6 x 8
- Font 1: size 8 x 14
- Font 2: size 12 x 22
- Font 3: size 16 x 30
- Font 4: size 24 x 42
- Font 5: size 32 x 58
- Font 6: size 48 x 84
- Font 7: size 64 x 116

The first 32 characters are not available at all font sizes. For small sizes not all symbols are displayable. For font size 4 and larger they are removed to save memory.

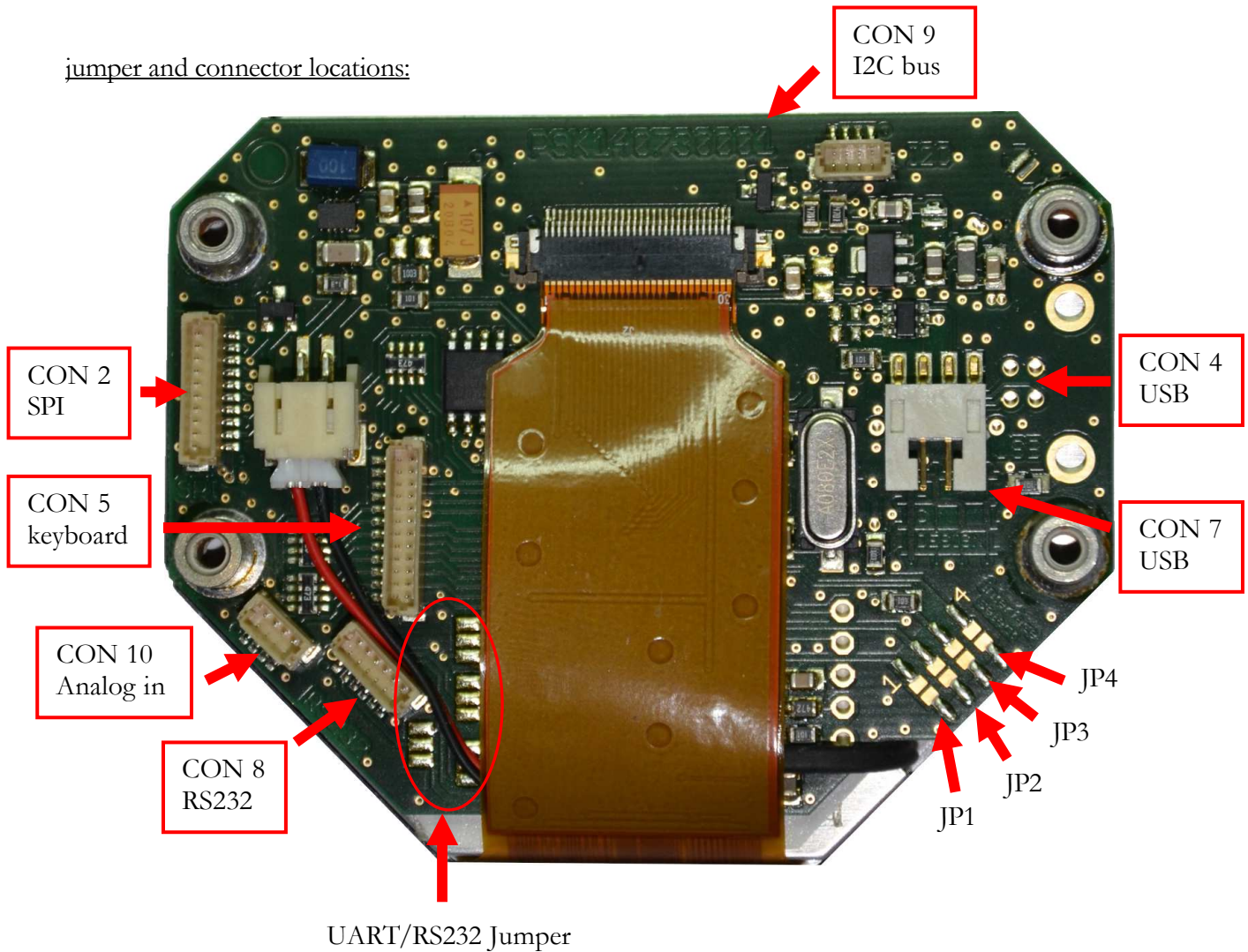
	☺	☹	♥	♦	♣	♠	○	●	◻	◼	♂	♀	♪	♫	☀
▶	◀	↕	!!	¶	§	—	↓	↑	↓	→	←	└	↔	▲	▼
	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{		}	~	⊠
Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	∅	×	f
á	í	ó	ú	ñ	Ñ	ª	º	¿	®	¬	½	¼	¡	«	»
☼	☽	☾		†	Á	Â	À	©	¶		⌈	⌋	ç	¥	⌋
L	⊥	⊤	†	—	†	ã	Ã	ℓ	⌈	≡	⌈	⌋	=	⌈	⊠
ð	Ð	Ê	Ë	È	ı	Í	Î	Ï	⌋	⌈	■	■	ı	Ì	■
Ó	β	Ô	Ò	Õ	Õ	μ	ρ	ρ	Ú	Û	Ù	ý	Ý	—	´
-	±	=	¾	√	Σ	÷	,	°	¨	▪	1	3	2	■	Ω

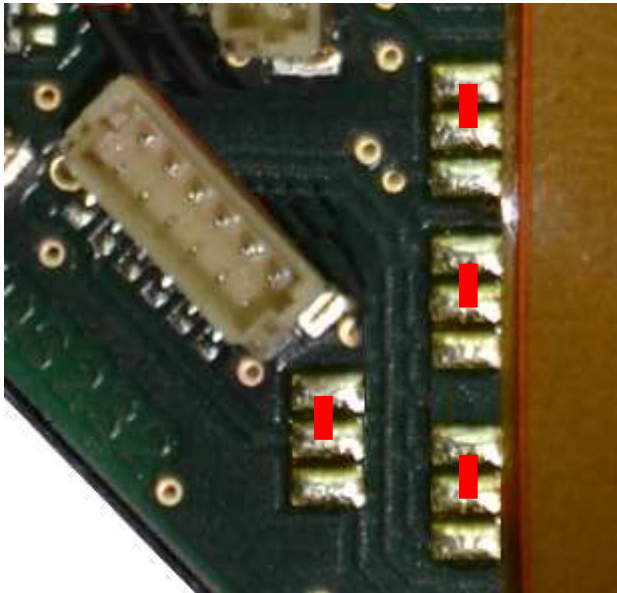
## Interface selection

The module pcb carries four solder jumpers. The first two (JP1, JP2) selects the interface, the second two (JP3, JP4) set different options on each interface.

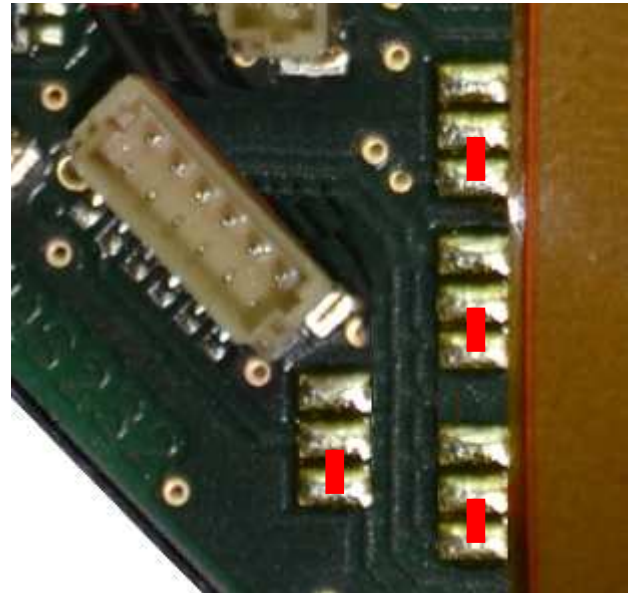
JP1	JP2	JP3	JP4	Interface	Option
open	open	open	open	USB	normal operation
		closed	open		backlight always on
		open	closed		reserved (do not use)
		closed	closed		reserved (do not use)
closed	open	open	open	UART/RS232	19200 Baud
		closed	open		57600 Baud
		open	closed		115200 Baud
		closed	closed		500000 Baud
open	closed	open	open	SPI	SPI mode 0
		closed	open		SPI mode 1
		open	closed		SPI mode 2
		closed	closed		SPI mode 3
closed	closed	open	open	I <sup>2</sup> C	I <sup>2</sup> C address 0x70
		closed	open		I <sup>2</sup> C address 0x72
		open	closed		I <sup>2</sup> C address 0x78
		closed	closed		I <sup>2</sup> C address 0x7A

jumper and connector locations:



UART/RS232 jumper settings:

Jumper setting for RS232 levels



Jumper setting for 3,3V logic levels

**Notes for UART (RS232), SPI and I<sup>2</sup>C interfaces**

If the input buffer is filled by more than 75%, the module indicates to stop data transfer.

The user should stop data transmission to avoid a buffer overflow but can finish currently transferred command if its string is not longer than 256 bytes. Handshaking is done via different ways dependent on selected interface:

- UART/RS232: RTS is set to high
- SPI: BUSY is set to high
- I<sup>2</sup>C: BUSY (from SPI) is set to high and module address is not acknowledged. Note: During a transmission it is not possible to determine buffer state without checking BUSY from SPI, because address acknowledge only affects start of transmission. This should be kept in mind when using I<sup>2</sup>C.

Storing pictures into flash (STOREPICFLASH, STORESTARTUP and STOREINDICATOR commands) is possible via all interfaces, but should only be done using USB. During programming of flash no reception is possible, even if there is space in the input buffer. Therefore the module sets the BUSY line and RTS to high and stops acknowledging of I<sup>2</sup>C data 1ms prior of starting programming. If data transmission continues, data loss can occur.

When transferring picture data to the module, a special case occurs when SPI interface is selected: Because data is read by shifting data in, it is not possible to read the answer without sending (dummy) data to the module. In normal command mode, characters other than the starting character are ignored. For picture data all values are interpreted as data. It would distort picture when reading the command answer. To avoid this problem in SPI mode an additional start byte is necessary to start picture transfer:

After acknowledge of picture transfer a loading command is send and the module ignores all received characters until 0x55 is received. All data are interpreted as picture data until picture transfer is finished.

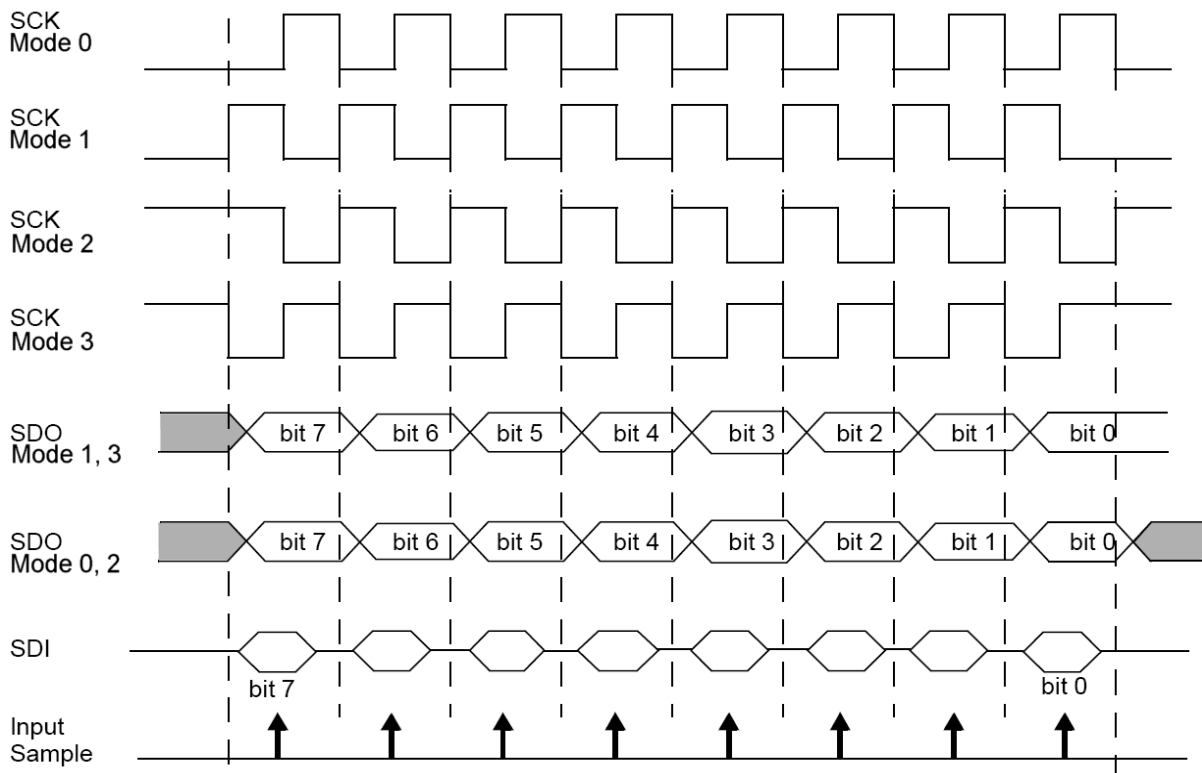
When sending dummy data to read an answer or to finish an unfinished command, send 0 as data.

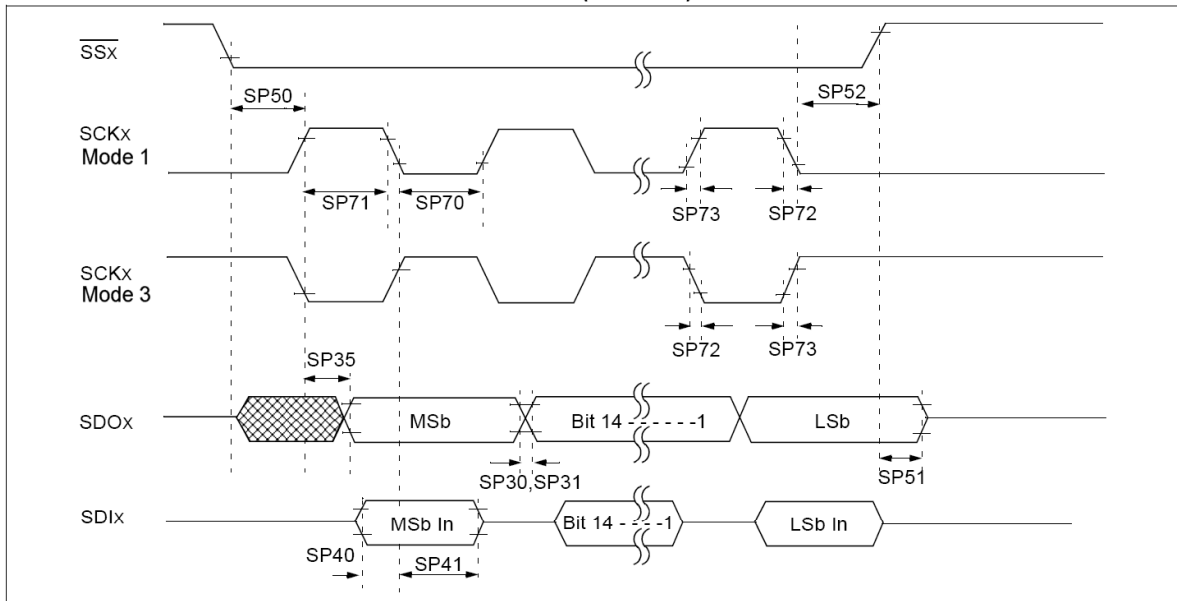
In SPI and I<sup>2</sup>C mode the user initiates a read operation. Therefore the module always has to send some data even no valid data are available. The module sends 0 as dummy data. When waiting for an answer the user should read data and discard all data until the start byte (211) is received. After that the remaining answer should be read as valid data.

The start of an answer can occur any time, so it is not necessary to deselect the module in SPI mode or restart a read operation in I<sup>2</sup>C.

Answer data are collected in a transmit buffer, therefore the user can send several commands before reading answer data. About 100 answers can be stored until the buffer overflows. If this occurs all further answers (or parts of them) will get lost. This should be avoided.

**SPI timing**

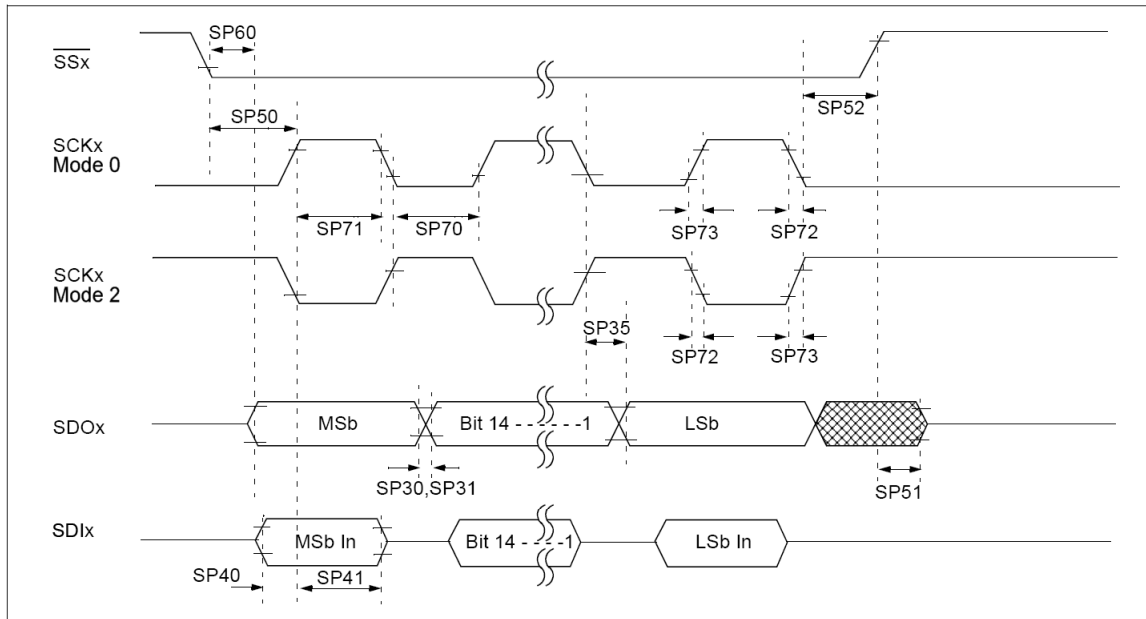




**SPIx MODULE SLAVE MODE ( 1 or 3 ) TIMING REQUIREMENTS**

AC CHARACTERISTICS			Standard Operating Conditions: 2.3V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics <sup>(1)</sup>	Min.	Typ. <sup>(2)</sup>	Max.	Units	Conditions
SP70	TscL	SCKx Input Low Time <b>(Note 3)</b>	Tsck/2	—	—	ns	—
SP71	Tsch	SCKx Input High Time <b>(Note 3)</b>	Tsck/2	—	—	ns	—
SP72	TscF	SCKx Input Fall Time	—	—	—	ns	See parameter DO32
SP73	TscR	SCKx Input Rise Time	—	—	—	ns	See parameter DO31
SP30	TdoF	SDOx Data Output Fall Time <b>(Note 4)</b>	—	—	—	ns	See parameter DO32
SP31	TdoR	SDOx Data Output Rise Time <b>(Note 4)</b>	—	—	—	ns	See parameter DO31
SP35	Tsch2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	—	15	ns	VDD > 2.7V
			—	—	20	ns	VDD < 2.7V
SP40	TdI2scH, TdI2scL	Setup Time of SDIx Data Input to SCKx Edge	10	—	—	ns	—
SP41	Tsch2dIL, TscL2dIL	Hold Time of SDIx Data Input to SCKx Edge	10	—	—	ns	—
SP50	TssL2scH, TssL2scL	SSx ↓ to SCKx ↑ or SCKx Input	175	—	—	ns	—
SP51	TssH2doZ	SSx ↑ to SDOx Output High-Impedance <b>(Note 3)</b>	5	—	25	ns	—
SP52	Tsch2ssH TscL2ssH	SSx after SCKx Edge	Tsck + 20	—	—	ns	—

- Note 1:** These parameters are characterized, but not tested in manufacturing.  
**Note 2:** Data in “Typical” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.  
**Note 3:** The minimum clock period for SCKx is 40 ns.  
**Note 4:** Assumes 50 pF load on all SPIx pins.



**SPIx MODULE SLAVE MODE ( 0 or 2 ) TIMING REQUIREMENTS**

AC CHARACTERISTICS			Standard Operating Conditions: 2.3V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics <sup>(1)</sup>	Min.	Typical <sup>(2)</sup>	Max.	Units	Conditions
SP70	TscL	SCKx Input Low Time <b>(Note 3)</b>	Tsck/2	—	—	ns	—
SP71	TscH	SCKx Input High Time <b>(Note 3)</b>	Tsck/2	—	—	ns	—
SP72	TscF	SCKx Input Fall Time	—	5	10	ns	—
SP73	TscR	SCKx Input Rise Time	—	5	10	ns	—
SP30	TdoF	SDOx Data Output Fall Time <b>(Note 4)</b>	—	—	—	ns	See parameter DO32
SP31	TdoR	SDOx Data Output Rise Time <b>(Note 4)</b>	—	—	—	ns	See parameter DO31
SP35	Tsch2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	—	20	ns	VDD > 2.7V
			—	—	30	ns	VDD < 2.7V
SP40	TdIV2scH, TdIV2scL	Setup Time of SDIx Data Input to SCKx Edge	10	—	—	ns	—
SP41	Tsch2dIL, TscL2dIL	Hold Time of SDIx Data Input to SCKx Edge	10	—	—	ns	—
SP50	TssL2scH, TssL2scL	SSx ↓ to SCKx ↓ or SCKx ↑ Input	175	—	—	ns	—

- Note 1:** These parameters are characterized, but not tested in manufacturing.  
**Note 2:** Data in “Typical” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.  
**Note 3:** The minimum clock period for SCKx is 40 ns.  
**Note 4:** Assumes 50 pF load on all SPIx pins.

SPI frequency can be as high as 25MHz but the average data rate (8 or more bytes) must not be higher than 250kByte/s. Otherwise data loss can occur.

## Technical data

Logic supply voltage:  $5V \pm 5\%$

Supply current (at 5.0V):

- backlight 0%, typ.:  
idle: 55mA, busy: 105mA
- backlight 100%, typ.:  
idle 300mA, busy: 350mA
- max. 500 mA

Operating temperature:  $-25^{\circ}\text{C}$  to  $+75^{\circ}\text{C}$

Storage temperature:  $-30^{\circ}\text{C}$  to  $+80^{\circ}\text{C}$



## Precautions

- The display unit is an electrostatic sensitive component. To prevent destruction of the elements by static electricity, be careful to maintain an optimum work environment.
  - a. Be sure to ground the body when handling the display units.
  - b. Tools required for assembly, such as soldering irons, must be properly ground.
  - c. To reduce the amount of static electricity generated, do not carry out assembly and other work under dry conditions.
  - d. The LCD is coated with a film to protect the display surface during assembly and transportation. Be careful when peeling off this protective film since static electricity may be generated. Pull it off slowly.
  
- The display panel is made of glass. Do not subject it to a mechanical shock by dropping it from a high place, etc.
  
- The surface of the display is covered with a polarizer, which easily can be scratched with spiky and sharp materials. Handle the polarizer carefully.  
 if the display surface is contaminated, breathe on the surface and gently wipe it with a soft dry cloth. If still not completely clear, moisten cloth with one of the following solvents:
  - Isopropyl alcohol or Ethyl alcohol
 Solvents other than those may damage the polarizer. Especially, do not use the following fluids:
  - Water, Ketone, Aromatic solvents
 Remove any fluids immediately.
  
- Condensation on the display surface is not allowed. This will damage the polarizer in short term.
  
- The display is pressure sensitive. Uneven or high pressure cause change in optical appearance or may destroy the display. Be careful while mounting the unit. The mounting must not cause any mechanical stress in the display or its interconnection.
  
- Do not expose the display to direct sunlight or intensive ultraviolet rays for many hours.  
 The polarizer and the liquid crystal are deteriorated by ultraviolet rays.
- Don't activate the same area of the display with the same text or picture for longer than 5 minutes.  
 This may cause an image sticking on the display.
- The display is backlit with white LEDs. The lifetime of these LEDs is not as long as of green or red LEDs. Therefore it is recommend to reduce the brightness to a level which is just sufficient for the application and to reduce the brightness to a low level or switch the backlight off, if the display is not used.
- The write cycle count of flash memory is limited. Use store picture command as rarely as possible.
- Do not apply signals without applying power to the display unit. This may damage the whole unit.
- Never exceed 5.5V on any signal pin (except RS232), even for short time or with current limited.
- Analogue input value shall not exceed 0-3.3V and should only be applied when power is present.

## Revisions

23/07/2014      V3.0-Short