DISTEC GMBH

IMAGELINK-50-USB2.0

5” TFT Display with USB 2.0 HighSpeed Interface

CONTENT

1. GENERAL DESCRIPTION

The ImageLink technology is providing an easy to use USB Interface for TFT displays.
The ImageLink-50-USB comes with a 5” QVGA (320x240) Display. For Display details see the Display Specification.
The ImageLink-50-USB is a self powered USB device with only needs a single 12V voltage supply.
ImageLink-50-USB also provides an integrated four wire resistive touch controller.

2. USB INTERFACE DESCRIPTION

The ImageLink-50-USB2.0 provides a USB 2.0 HighSpeed interface.

Vendor ID: ????
Product ID: ????

The ImageLink-50-USB is providing 3 USB endpoints (see figure 1).

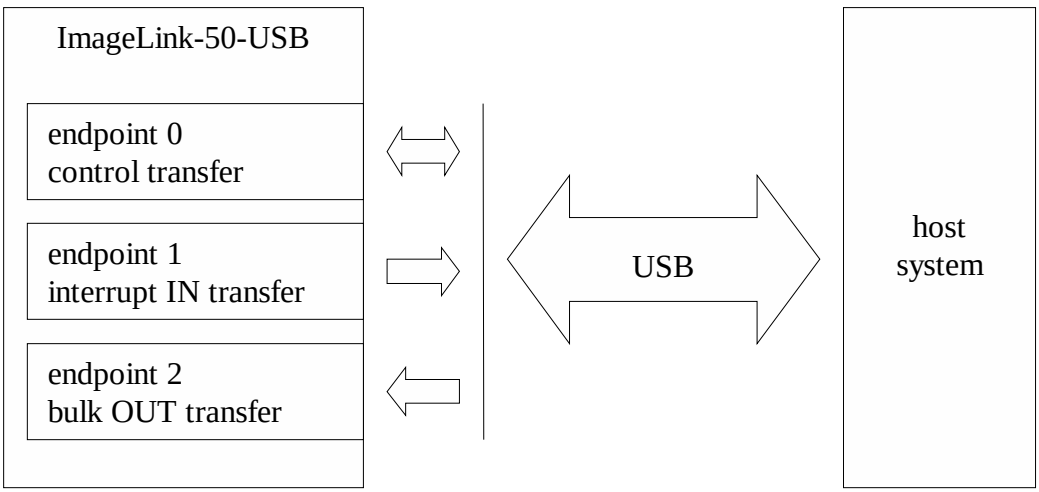


figure 1: ImageLink datapipes

2.1 Endpoint overview

The **endpoint 0** is the mandatory control endpoint. Over this endpoint you switch the backlight on/off, control the brightness and switch the display power on/off over USB vendor calls. There are also vendor calls for initalizing and restarting the ImageLink picture transmission. The USB vendor calls are described in chapter 3.

The **endpoint 1** is an interrupt endpoint. Over this endpoint you get a status code periodically and error codes if occur. Also the touch screen data comes over this pipe. The functionality of endpoint 1 is described in chapter 4.

The **endpoint 2** is a bulk out endpoint. Just send the image you want to show on the ImageLink display, over this pipe to the endpoint. There is no protocol overhead. The image format is a raw type with 320x240 points at 16 Bit color depth. The ImageLink is double buffered. You can transfer images on “low” Full- Speed USB 1.1 Interfaces without showing frame tears. A new Image is only showed after a buffer was fully filled. The functionality of endpoint 2 is described in chapter 5.

2.2 USB Descriptors

2.2.1 HIGH-SPEED

```
ImageLink-50-USB2.0  V01.26
Manufacturer: Data Display Group
Serial Number: xxxxxx
Speed: 480Mb/s (high)
USB Version:  2.00
Device Class: 00(>ifc )
Device Subclass: 00
Device Protocol: 00
Maximum Default Endpoint Size: 64
Number of Configurations: 1
Vendor Id: ????
Product Id: ????
Revision Number:  1.26

Config Number: 1
      Number of Interfaces: 1
      Attributes: c0
      MaxPower Needed:   0mA

      Interface Number: 0
            Name: dd_lou
            Alternate Number: 0
            Class: ff(vend.)
            Sub Class: 0
            Protocol: 0
            Number of Endpoints: 2

                  Endpoint Address: 81
                  Direction: in
                  Attribute: 3
                  Type: Int.
                  Max Packet Size: 64
                  Interval: 16ms

                  Endpoint Address: 02
                  Direction: out
                  Attribute: 2
                  Type: Bulk
                  Max Packet Size: 512
                  Interval: 0ms
```

2.2.2 FULL-SPEED

```
ImageLink-50-USB2.0  V01.26
Manufacturer: Data Display Group
Serial Number: xxxxxx
Speed: 12Mb/s (full)
USB Version:  2.00
Device Class: 00(>ifc )
Device Subclass: 00
Device Protocol: 00
Maximum Default Endpoint Size: 64
Number of Configurations: 1
Vendor Id: ????
Product Id: ????
Revision Number:  1.26

Config Number: 1
      Number of Interfaces: 1
      Attributes: c0
      MaxPower Needed:   0mA

      Interface Number: 0
            Name: dd_lou
            Alternate Number: 0
            Class: ff(vend.)
            Sub Class: 0
            Protocol: 0
            Number of Endpoints: 2

                  Endpoint Address: 81
                  Direction: in
                  Attribute: 3
                  Type: Int.
                  Max Packet Size: 64
                  Interval: 8ms

                  Endpoint Address: 02
                  Direction: out
                  Attribute: 2
                  Type: Bulk
                  Max Packet Size: 64
```

3. USB VENDOR CALLS OVER ENDPOINT 0

The ImageLink should be configured over the 8 Byte of setup data of the specified vendor calls over transfer endpoint 0.
For the USB standard requests see the USB Specification.

3.1 Backlight requests:

3.1.1 GET_BKL

The GET_BKL request is intended to determine the backlight status (switch ON/OFF).

Request	bmRequest Type	bRequest	wValue	wIndex	wLength	Data	
						1. Byte	2. Byte
GET_BKL	0xC0	0xB0	0x0000	0x0000	0x0002	bitmap	0xB0

bitmap:

b7	b6	b5	b4	b3	b2	b1	b0
reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	actual state 0= OFF 1= ON

3.1.2 SET_BKL

The SET_BKL request is intended to switch the backlight ON/OFF.

Request	bmRequest Type	bRequest	wValue		wIndex	wLength	Data
			1. Byte	2. Byte			
SET_BKL	0xC0	0xB1	bitmap	value	0x0000	0x0002	0x00B1

bitmap:

b7	b6	b5	b4	b3	b2	b1	b0
reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	set actual value

each bit can be set individually or together.

value:

0=OFF
1=ON

3.1.3 GET_BRIGHTNESS

The GET_BRIGHTNESS is intended to determine the brightness value. It can query the actual and the stored default values.

Request	bmRequest Type	bRequest	wValue	wIndex	wLength	Data	
						1. Byte	2. Byte
GET_BRIGHTNESS	0xC0	0xB2	mode	0x0000	0x0002	value	0xB2

mode:

0: get actual value
1: get stored default value

value:

0 to 255
0 is min brightness

3.1.4 SET_BRIGHTNESS

The SET_BRIGHNESS request is intended to set the actual and stored default brightness values. The decision which value should be changed is made by the bitmap in the first byte of wValue.

Request	bmRequest Type	bRequest	wValue		wIndex	wLength	Data
			1. Byte	2. Byte			
SET_ BRIGHTNESS	0xC0	0xB3	bitmap	value	0x0000	0x0002	0x00B3

bitmap:

b7	b6	b5	b4	b3	b2	b1	b0
reserved	reserved	reserved	reserved	reserved	set to stored default value (2)	store value as default (1)	set actual value (1)

(1) each bit can be set individually or together.
(2) has priority over b0

value:

0 to 255
0 is min brightness

3.2 Display requests

3.2.1 GET_DISPLAY_POWER

The GET_ DISPLAY_POWER request is intended to determine the display power status (switch display power ON/OFF and power good signal from power supply).

Request	bmRequest Type	bRequest	wValue	wIndex	wLength	Data	
						1. Byte	2. Byte
GET_ DISPLAY_POWER	0xC0	0xB4	0x0000	0x0000	0x0002	bitmap	0xB4

bitmap:

b7	b6	b5	b4	b3	b2	b1	b0
reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	0= no power 1= power good	reserved 0	actual state 0= OFF 1= ON

3.2.2 SET_ DISPLAY_POWER

The SET_DISPLAY_POWER request is intended to switch the display power ON/OFF. A return bitmap announces success or failure of power up or down.

Request	Bm Request Type	B Request	wValue		wIndex	wLength	Data	
			1. Byte	2. Byte			1. Byte	2. Byte
SET_ DISPLAY _POWER	0xC0	0xB5	bitmap	value	0x0000	0x0002	return bitmap	0xB5

bitmap:

b7	b6	b5	b4	b3	b2	b1	b0
reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	set actual value

each bit can be set individually or together.

value:

0=OFF
1=ON

return bitmap:

b7	b6	b5	b4	b3	b2	b1	b0
reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	0= no power 1= power good	reserved 0	actual state 0= OFF 1= ON

3.2.3 GET_TRANSFER_COUNTER

The GET_TRANSFER_COUNTER is intended for controlling the number of transferred pixels. At start of a transmission of one picture bitmap the counter has to be 320x240 pixels (1). For each transferred pixel the counter will decremented. After transferring a complete picture bitmap of 320x240 pixels the counter is automatically reinitialized and at a value of 320x240 pixels (1).

Request	bmRequest Type	bRequest	wValue	wIndex	wLength	Data	
						1.-4. Byte	5. Byte
GET_ TRANSFER_ COUNTER	0xC0	0xB6	mode	0x0000	0x0005	value[4]	0xB6

mode:

0: get actual value
1: get stored default value (1)

value:

value[0] = couter_byte_0
value[1] = couter_byte_1
value[2] = couter_byte_2
value[3] = couter_byte_3

(1) at present there is only a fixed value:

couter_byte_0 = 0x00
couter_byte_1 = 0xc2
couter_byte_2 = 0x01
couter_byte_3 = 0x00
this is corresponding to 320x240 pixels

3.2.4 SET_ TRANSFER_COUNTER

The SET_TRANSFER_COUNTER is used for initializing an image transfer. After some conditions which leave the ImageLink in an undefined state it has to be reinitialized with this request.

Request	bmRequest Type	bRequest	wValue		wIndex	wLength	Data	
			1. Byte	2. Byte			1. Byte	2. Byte
SET_ TRANSFER _COUNTER	0xC0	0xB7	bitmap	0x00	0x0000	0x0002	return bitmap	0xB7

bitmap:

b7	b6	b5	b4	b3	b2	b1	b0
reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	set to stored default value (1)	reserved 0	reserved 0

each bit can be set individually or together.

The return bitmap value of “Data” depends on display power state. If there is no switched ON or no display power good signal the SET_TRANSFER_COUNTER will fail and the return value is 0x00. Otherwise the return value is 0x01.

return bitmap:

b7	b6	b5	b4	b3	b2	b1	b0
reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	reserved 0	0= failure 1= success

(1) at present there is only a fixed value:

couter_byte_0 = 0x00
couter_byte_1 = 0xc2
couter_byte_2 = 0x01
couter_byte_3 = 0x00
this corresponds with 320x240 = 76800 pixel

4. INTERRUPT TRANSFER DATA OVER ENDPOINT 1

The interrupt endpoint 1 has several jobs. First it's the data pipe for status and error flags and second for the touch screen data. The data is transferred as block of word (2 bytes) items in little endian byte order which can be easily abstracted with a e.g. C programming language data structure.

title	**LSB**	**MSB**
status flags	value	*reserved = 0x00*
error flags	value	*reserved = 0x00*
touch screen state	value	*reserved = 0x00*
x- position	value	*reserved = 0x00*
y- position	value	*reserved = 0x00*
z1	value	*reserved = 0x00*
z2	value	*reserved = 0x00*
reserved	*reserved = 0x00*	*reserved = 0x00*
reserved	*reserved = 0x00*	*reserved = 0x00*
reserved	*reserved = 0x00*	*reserved = 0x00*
reserved	*reserved = 0x00*	*reserved = 0x00*
reserved	*reserved = 0x00*	*reserved = 0x00*

figure 2: interrupt transfer data block

```
typedef struct _IntTransfer {
   unsigned short state;
   unsigned short error;
   unsigned short touch_state;
   unsigned short x;
   unsigned short y;
   unsigned short z1;
   unsigned short z2;
 }IntTransfer;
```

figure 3: C data struct

4.1 Status flags

b7	b6	b5	b4	b3	b2	b1	b0
reserved	reserved	reserved	reserved	status flag was pending	RAMDAG switched to new picture	reserved	complete picture was transferred

4.2 Error flags

b7	b6	b5	b4	b3	b2	b1	b0
reserved	reserved	reserved	reserved	error flag was pending	reserved	counter expired but memory isn't fully filled	fully filled memory but counter not expired

4.3 Touch screen data

The ImageLink integrated touch screen controller provides its measurements periodically. The transmission of this data is announced with the "touch screen data" flag (see table 1: touch screen state flags). The measurement methods are shown in figure 4 and figure 5. For determining the pressure of a touch event the equation from figure 6 can be used.

b7	b6	b5	b4	b3	b2	b1	b0
reserved	reserved	reserved	reserved	reserved	reserved	reserved	touch screen data

table 1: touch screen state flags

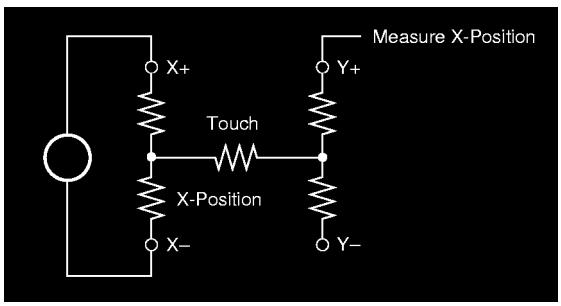
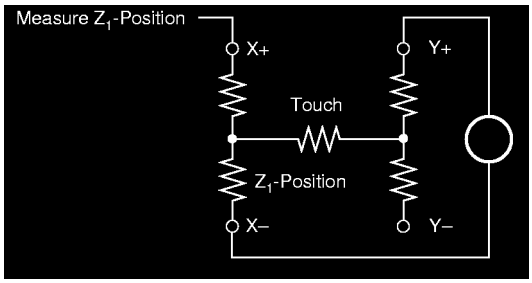


figure 4: X-position measurement (Y-position is correlative)



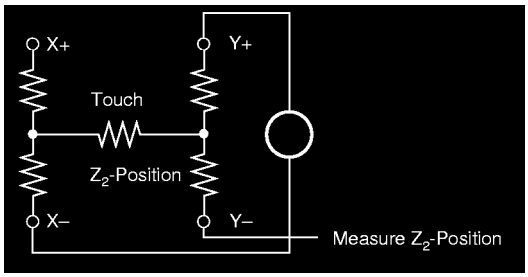


figure 5: Z1 and Z2 measurement

$$R_{TOUCH} = \frac{R_{X-Plate} \cdot X_{Position}}{4096} \cdot \left(\frac{Z_2}{Z_1} - 1 \right)$$

figure 6: pressure equation method

5. BULK TRANSFER OVER ENDPOINT 2

The bulk transfer endpoint 2 is the main data pipe for transmitting pictures to the ImageLink. From the programmer's point of view it's very easy to use. After you have initialized the ImageLink once by a SET_TRANSFER_COUNTER vendor call (see chapter 13), just send the picture you will show on the display as a bitmap starting with pixel '1' followed by the whole picture ending with pixel '76800' (see figure 7: pixel order). The picture will be stored in one half of the ImageLink video graphics memory. After a whole picture has been transferred the RAMDAC points to the new picture and shows it on the display. A status flag transmitted over the interrupt endpoint 1 announces the successful transfer of the picture (see chapter 14). Then the other half of the ImageLink video memory is free for receiving another picture and a new picture transfer can start. For higher performance there is no need to wait for the status flag. You only have to control the number of transferred bytes which must be the number of a whole picture. If the data is committed to the USB system is very likely that the transfer will go well because of the error controlled nature of the USB bulk transfer. So you can immediately start another picture transfer. The ImageLink is generously buffered. But you should always take care about the error flags transmitted over interrupt endpoint 1 (see chapter 14). In the unlikely event of an error you have to stop your current transmission and to reinitialize the ImageLink with the SET_TRANSVER_COUNTER vendor call over endpoint 0. Then you can resume the picture transmission.

Each pixel is represented with 16 Bits in BGR order. The red and blue color have a depth of 5 Bits the green color 6 Bits. So it's the common format BGR(565) (see figure 8: BGR color order) in little endian order (see figure 9: pixel format).

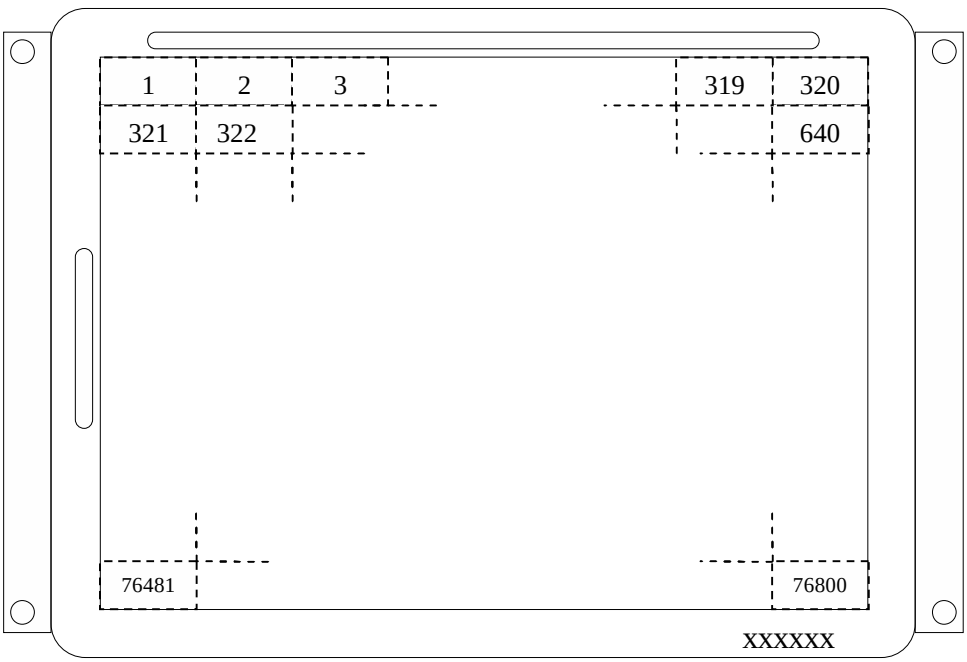


figure 7: pixel order

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

figure 8: BGR color order

LSB								MSB							
b8	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
G2	G1	G0	B4	B3	B2	B1	B0	R4	R3	R2	R1	R0	G5	G4	G3

figure 9: pixel format (little endian)

6. PROGRAMMFLOW OF IMAGELINK USAGE

In figure 10 is a typical program flow for using ImageLink-50-USB shown.

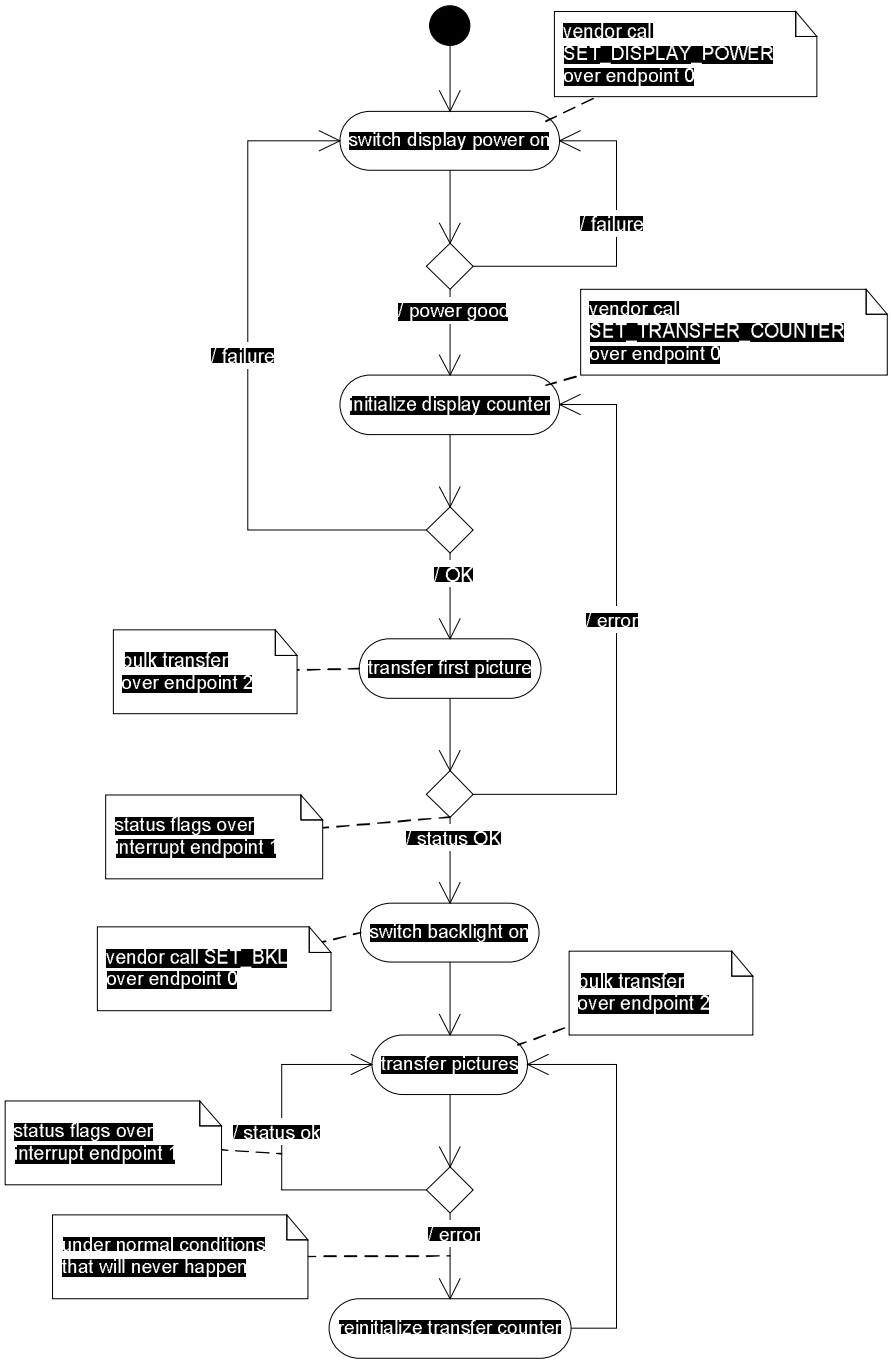


figure 10: typical program flow

7. ELECTRICAL CHARACTERISTICS

Operating conditions*

Item	Symbol	Min.	Typ.	Max.	Unit	Note
Supply Voltage	V_{IL}	10.8	12.0	13.2	V	
Supply Current	I_{IL}	-	115	215	mA	Display OFF BKL OFF
	I_{IL}	-	225	185	mA	Display ON BKL OFF
	I_{IL}	-	275	455	mA	Display ON BKL ON min. brightness
	I_{IL}	-	585	815	mA	Display ON BKL ON max. brightness

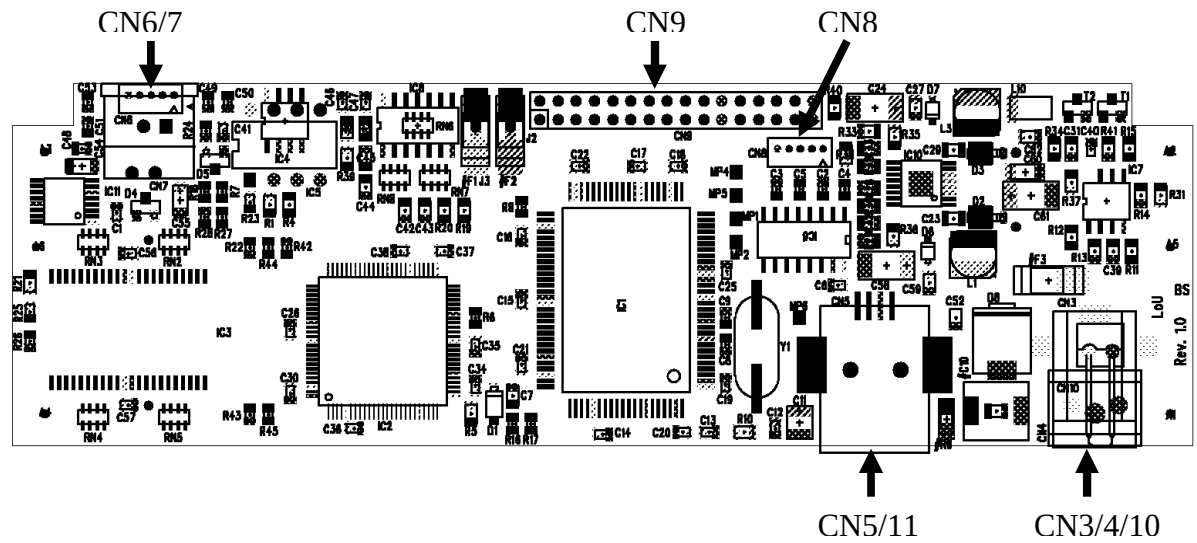
Electrical Absolute Ratings*

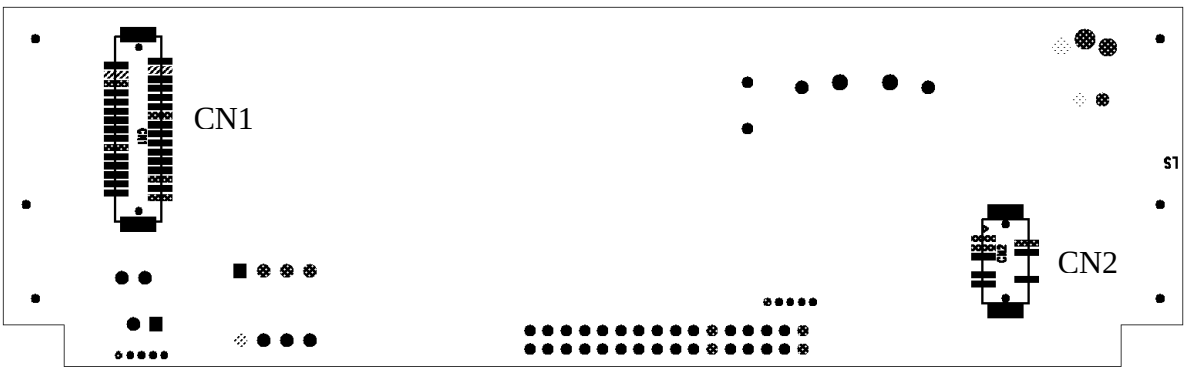
Item	Symbol	Min.	Max.	Unit	Note
Supply Voltage	V_{IL}	-0.3	15	V	(1),(2)

Note (1) Within operating temperature
Note (2) Permanent damage to the device may occur if maximum values are exceeded.
Functional operation should be restricted to the conditions described under normal operating conditions.
(*) preliminary

8. CONNECTOR PIN ASSIGNMENT

8.1 Connector overview





CN#	Description	Type	Manufacturer
CN10	power supply	2.5mm Pitch Header Right Angle alternative to CN4/3	Molex (5046 Series)
CN4	power supply	DC Power Jack 2.0mm	Kycon (KLD-SMT Series)
CN3	power supply	Screw type terminal 1,5 mm² alternative to CN4/10	Weco (95 Series)
CN5	USB	USB B-Type	Kycon (KUSB-BS-1-N)
CN11	USB	1.25mm pitch board to wire alternative to CN5	Hirose (DF13 Series)
CN9	GPIO/ prog. interface	Pin header (2mm / 2x16)	--
CN7	touch screen interface	1.25mm pitch FFC/FPC Connector	DMC (KCA-4)
CN6	touch screen interface	1.25mm pitch board to wire alternative to CN10	Hirose (DF13 Series)
CN11	serial data interface	RS232 / optional	Hirose (DF13 Series)
CN1	TTL interface (internal use)	1mm pitch board to board	Hirose (DF9 Series)
CN2	inverter supply and control (internal use)	1mm pitch board to board	Hirose (DF9 Series)

8.2 Pin assignments

CN10	Power supply connector	
Pin	**Signal**	**Description**
Center	+12V	+12V power supply
2	GND	Ground
3	-	Not connected

CN4	Power supply connector (alternative)	
Pin	**Signal**	**Description**
1	GND	Ground
2	+12V	+12V power supply

CN3	Power supply connector (alternative)	
Pin	**Signal**	**Description**
1	+5V / VCCEXT	+5V power supply / Ext. panel power supply
2	GND	Ground

CN11	serial data interface (optional)	
Pin	**Signal**	**Description**
1	TXD0	Data out
2	RXD0	Data in
3	TXD1	Data out
4	RXD1	Data in
5	GND	Ground

CN7	touch screen interface	
Pin	**Signal**	**Description**
1	X+	X+ position input
2	Y+	Y- position input
3	X-	X+ position input
4	Y+	Y- position input

CN6	touch screen interface (alternative)	
Pin	**Signal**	**Description**
1	X+	X+ position input
2	Y+	Y- position input
3	X-	X+ position input
4	Y+	Y- position input
5	GND	Ground

CN5	USB connector	
Pin	**Signal**	**Description**
1	VBUS	USB Bus Voltage (not connected)
2	D-	USB D-
3	D+	USB D+
4	GND	Ground

CN11	USB connector	
Pin	**Signal**	**Description**
1	VBUS	USB Bus Voltage (not connected)
2	GND	Ground
3	D-	USB D-
4	D+	USB D+
5	GND	Ground

CN1	TTL interface (internal use)	
Pin	**Signal**	**Description**
1	NC	(not connected)
2	NC	(not connected)
3	+5V	+5V display power
4	+5V	+5V display power
5	NC	(not connected) (DE)
6	GND	ground
7	B5	blue 5 (MSB)
8	B4	blue 4
9	B3	blue 3
10	B2	blue 2
11	B1	blue 1
12	B0	blue 0 (LSB)
13	GND	ground
14	G5	green 5 (MSB)
15	G4	green 4
16	G3	green 3
17	G2	green 2
18	G1	green 1
19	G0	green 0 (LSB)
20	GND	ground
21	R5	red 5 (MSB)
22	R4	red 4
23	R3	red 3
24	R2	red 2
25	R1	red 1
26	R0	red 0 (LSB)
27	GND	ground
28	VSYNC	display vertical sync
29	HSYNC	display horizontal sync
30	CLK	display clock
31	GND	ground

CN2	inverter supply and control	
Pin	**Signal**	**Description**
1	NC	(not connected)
2	NC	(not connected)
3	NC	(not connected)
4	+12V	+12V backlight power
5	+12V	+12V backlight power
6	+12V	+12V backlight power
7	BKL_EN	backlight enable
8	BR_CTRL	brightness control
9	GND	ground
10	GND	ground
11	GND	ground

CN9	TTL – Signal CONNECTOR 9 (*)					
Pin	**Signal**	**Description**		**Pin**	**Signal**	**Description**
1	+5V	fused with F2 (nc)		2	+3.3V	fused with F1 (nc)
3	VBAT1			4	VBAT2	
5	IN1			6	IN2	
7	PC0	GPIO (+3.3V,+5V)		8	PA4	GPIO (+3.3V,+5V)
9	PC1	GPIO (+3.3V,+5V)		10	PA5	GPIO (+3.3V,+5V)
11	PC2	GPIO (+3.3V,+5V)		12	PA6	GPIO (+3.3V,+5V)
13	PC3	GPIO (+3.3V,+5V)		14	PE3	GPIO (+3.3V,+5V)
15	INT4	Interrupt		16	PE4	GPIO (+3.3V,+5V)
17	PE7	GPIO (+3.3V,+5V)		18	INT5	Interrupt
19	RESET			20	WAKEUP	
21	GND			22	GND	
23	SCL	ext. I2C interface		24	+3.3V	fused with F1 (nc)
25	SDA	ext. I2C interface		26	TMS	prog. interface
27	DIN	prog. interface		28	CLK	prog. interface
29	CTRL	prog. interface		30	DONE	prog. interface
31	GND			32	GND	

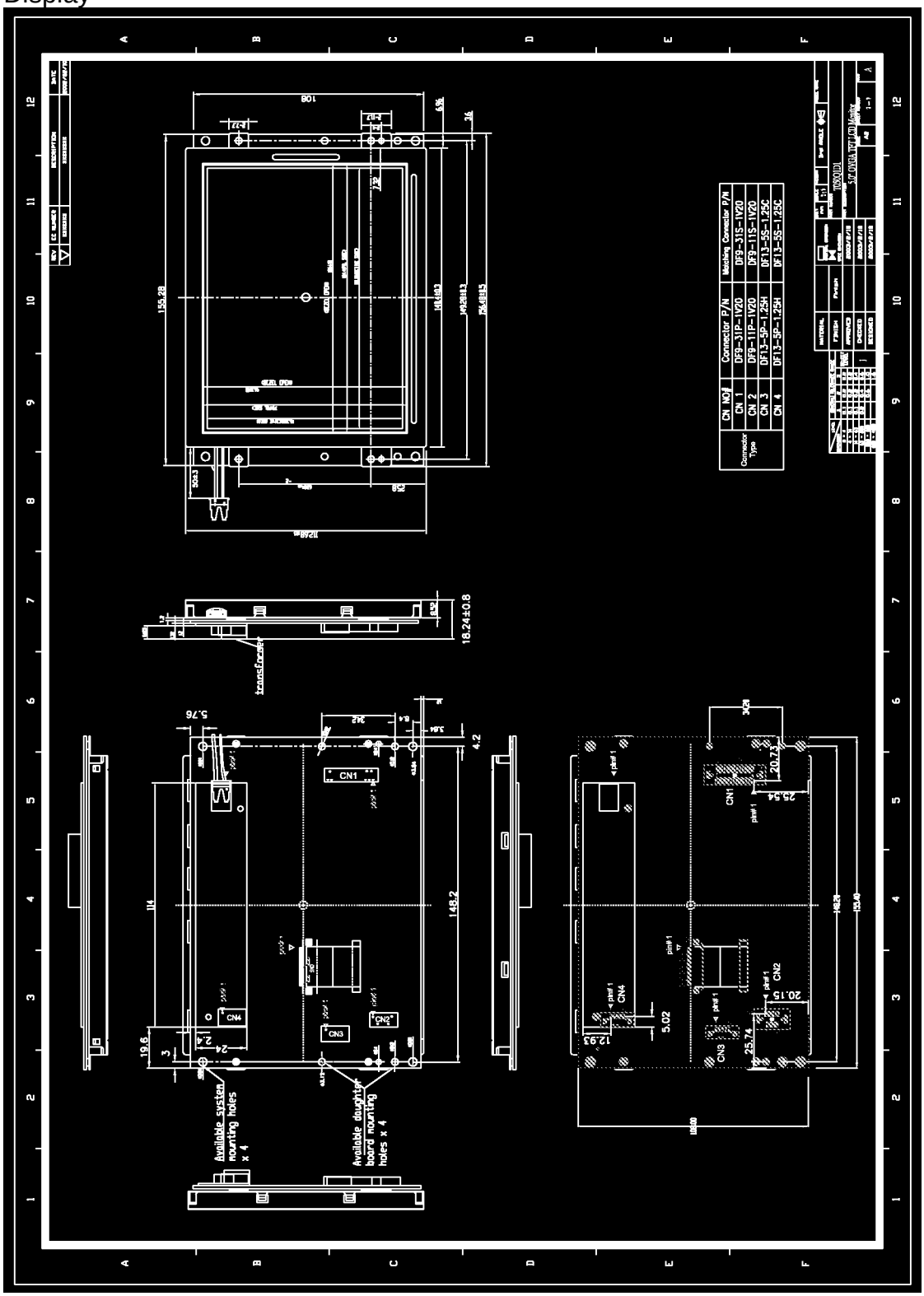
(*) optional GPIOs not implemented in firmware, maybe in future or on request

9. OUTSIDE DIMENSIONS

ImageLink

figure 11: ImageLink dimensions

Display


CN NO
Connector P/N
Mating Connector P/N
CN 1
DF9-31P-1V20
DF9-31S-1V20
CN 2
DF9-11P-1V20
DF9-11S-1V20
CN 3
DF13-5P-1.25H
DF13-5S-1.25C
CN 4
DF13-5P-1.25H
DF13-5S-1.25C
Connector Type
transformer
18.24±0.8
5.76
4.2
CN1
CN2
CN3
CN4
148.2
114
19.6
Available system mounting holes x 4
Available daughter board mounting holes x 4
20.73
25.54
20.15
12.93
5.02
25.74

10. MODIFICATIONS & IMPROVEMENTS

10.1 Version History

Rev.	Date	orig. of change	Description
1.0	31/03/2004	sw1	initial release