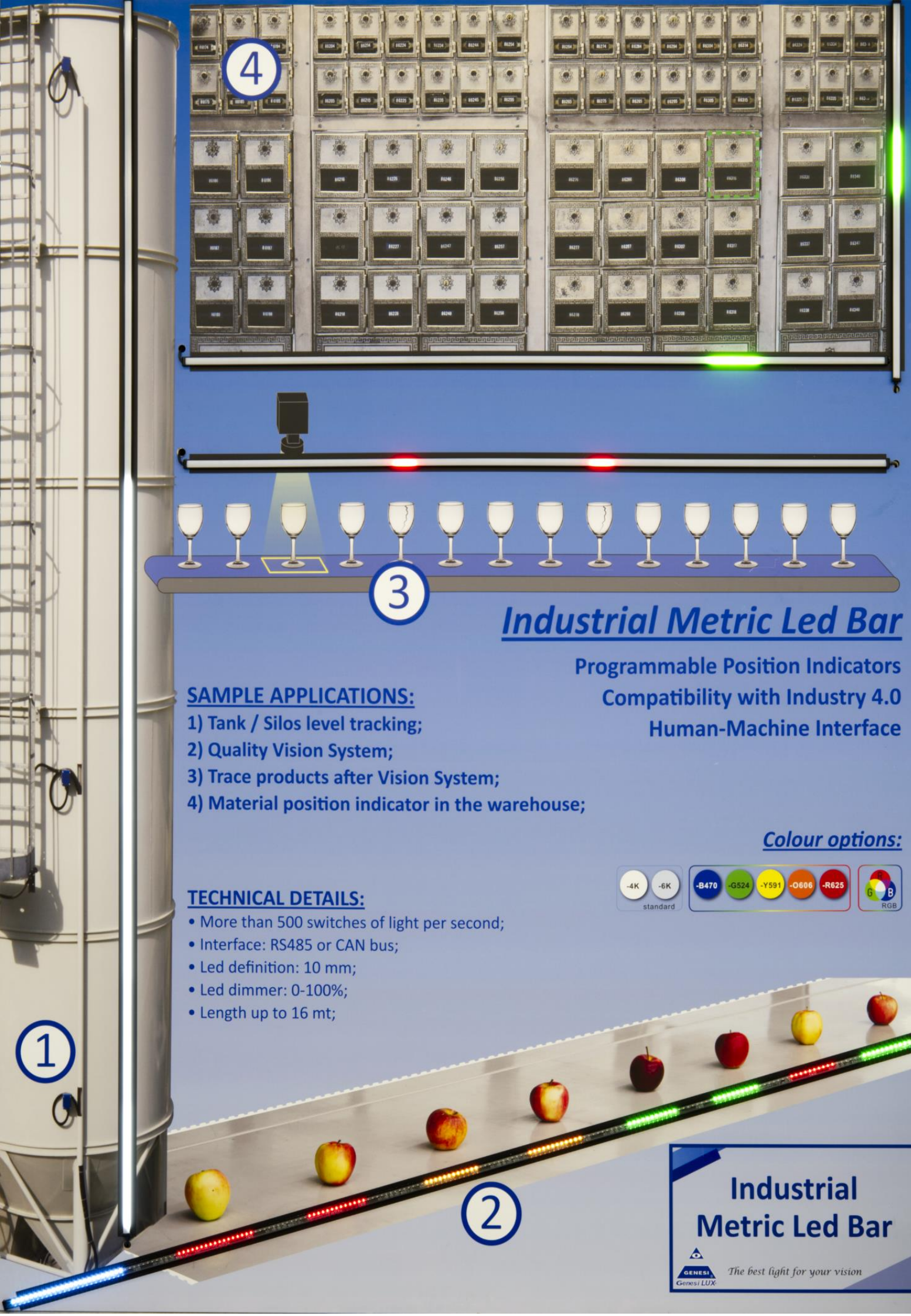


Industrial Metric Led Bar

Programmable Position Indicators
Compatibility with Industry 4.0
Human-Machine Interface

SAMPLE APPLICATIONS:

1) Tank / Silos level tracking;
2) Quality Vision System;
3) Trace products after Vision System;
4) Material position indicator in the warehouse;

Colour options:

-4K -6K standard | -B470 -G524 -Y591 -O606 -R625 | RGB

TECHNICAL DETAILS:

- More than 500 switches of light per second;
- Interface: RS485 or CAN bus;
- Led definition: 10 mm;
- Led dimmer: 0-100%;
- Length up to 16 mt;

Industrial Metric Led Bar

GENESI Genesi LUX

The best light for your vision

GE448 industrial metric LED bar RGB

Industrial metric LED bar RGB

DATASHEET

Features

- Microchip SAMC Arm CORTEX® core
- Low power consumption
- Modular assemblies
- Very small form factor
- Up to 64 addressable modules
- CE regulatory approvals

Applications

- Machine loading indication
- LED lamps signaling
- Position identification
- On path real time driving

Contents

1. Description

GE448 is a module with 25 - 3528 PLCC4 shape - LEDs individually addressable. Its tiny form factor, just 250mm x 11,8mm makes it ideal for thin, long lamp strips, in a huge amount of application. Each module can be assembled so that long lanes of 10mm slot of LEDs can be achieved. Up to 64 modules can be addressed by the same master controller.
Both internal and external wire cabling are available, 4 and 5 pin M12 connectors along with 4 pin M8.
RS485 (or CAN as option) buses are available for communication.

2. Hardware specification

General conditions are referred to V_{SUPPLY}=24Vcc and T_a=25°C, when not otherwise specified.

2.1 Absolute maximum ratings

Table 1. Absolute maximum ratings

Symbol	Parameter	Min.	Typ.	Max.	Unit
T_{STG}	Storage temperature range	-40	-	85	°C
V_{PS}	Power supply voltage	20	-	28	V
I_{PS}	Power supply current *	-	-	110	mA

-

2.2 Recommended operating conditions

Table 2. Recommended operating conditions

Symbol	Parameter	Min.	Typ.	Max.	Unit
T_A	Operating ambient Temperature range	0	-	60	°C
V_{PS}	Power supply voltage	22	24	26	V

2.3 Current consumption

Table 3. Current consumption

Symbol	Parameter	Test conditions.	Typ.	Unit
I_{DD}	Supply current for low power electronics	Control	17,5	mA
		Each active LED*	3,2	mA
		V_{LED}	4,85	V

(*) 100% WHITE

2.4 Pin assignment

In order to work properly, each module needs to be driven as follow:

SUPPLY: +24V, GND

Communication signals: RS485-A, RS485-B

Sync signal: common signal, reserved for future use, not wired in standard versions

They can be wired directly on the board pads (see next paragraph) or by means of M8 / M12 connector.

2.5 Pin placement

Each module has same electrical signal in both ends, so that cascade assembly can be done easily.

2.5.1 Input signals

Fig. 1 – Input electrical signals

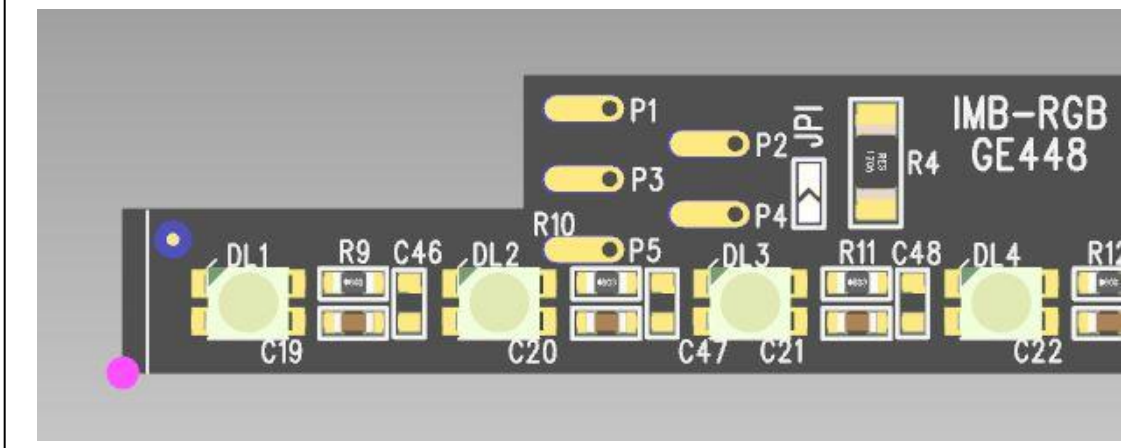


Table 4. Input pad assignment

PAD	Signal
P1	Gnd
P2	RS485 +
P3	RS485 -
P4	+24V supply
P5	RFU

2.5.2 Output signals

Fig. 2 – Output electrical signals

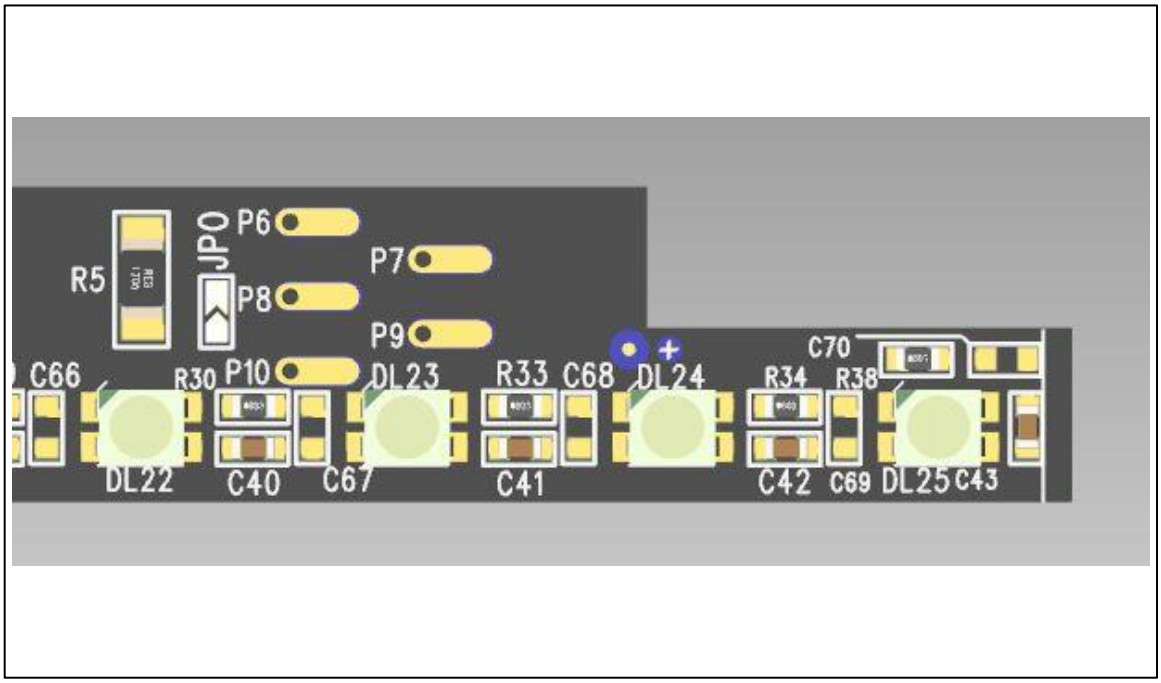


Table 5. Output pin assignment

PAD	Signal
P6	Gnd
P7	RS485 +
P8	RS485 -
P9	+24V supply
P10	RFU

2.6 Hardware block diagram

3. Hardware design

3.1 Features

3.1.1 Supply

In order to properly supply the device, a 24Vdc +/- 10% would be recommended. The total current consumption is related to :

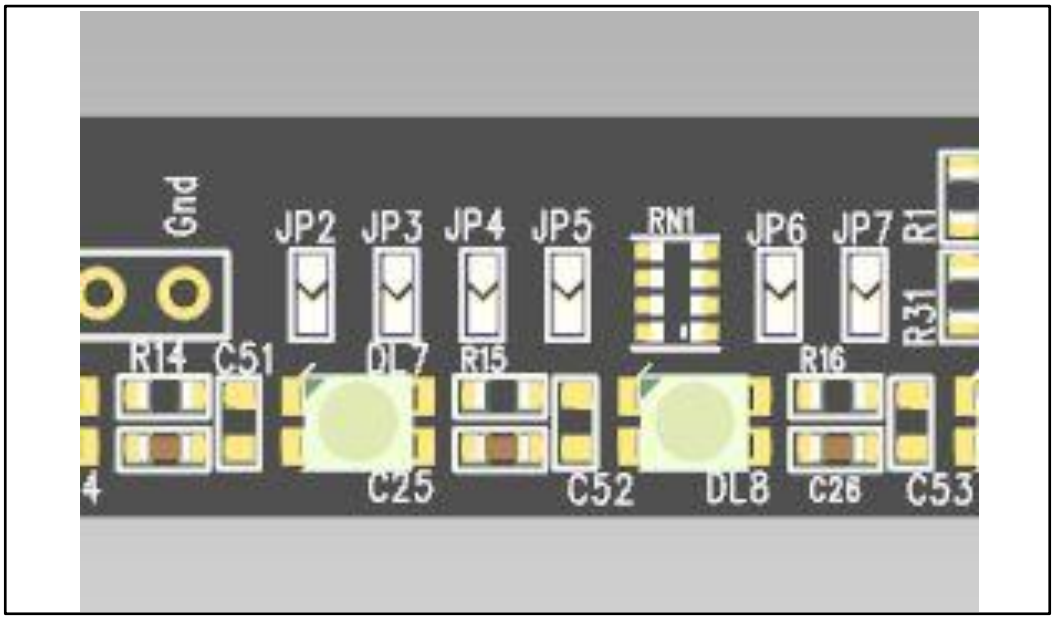
1) number of connected modules
2) number of effective working LEDs

It is strongly recommended to keep total amount of sink current below 3A peak.

3.1.2 Address setting

Each module can be individually programmed with a its own address by means of proper solderable PADs (read jumpers) on board.

Fig. 3 – Address pads/jumpers



If modules came assembled by factory, this work has been already done. You may see jumpers on pads JP2, JP3, JP4, JP5 along with JP6, JP7.
As general role, JP6 and JP7 select quarter modules within linear meter, and others the meters, as shown in table below:

Table 6. Address jumper configuration: meter (m) selection

Jumper / Meter (m)	JP2	JP3	JP4	JP5	Actual address (m)
1					0x10 + qm
2				✓	0x14 + qm
3			✓		0x18 + qm
4			✓	✓	0x1C + qm
5		✓			0x20 + qm
6		✓		✓	0x24 + qm
7		✓	✓		0x28 + qm
8		✓	✓	✓	0x2C + qm
9	✓				0x30 + qm
10	✓			✓	0x34 + qm
11	✓		✓		0x38 + qm
12	✓		✓	✓	0x3C + qm
13	✓	✓			0x40 + qm
14	✓	✓		✓	0x44 + qm
15	✓	✓	✓		0x48 + qm
16	✓	✓	✓	✓	0x4C + qm

Table 7. Quarter meter (qm) selection

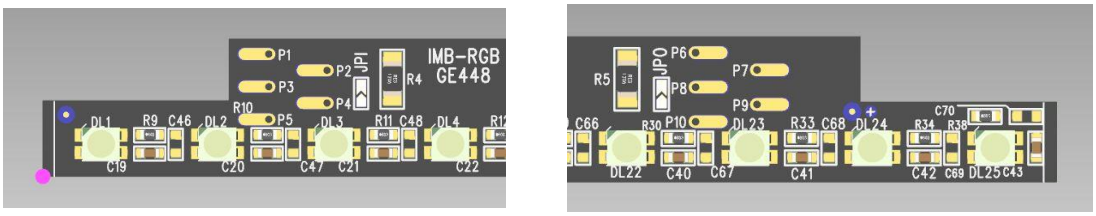
Jumper / Quarter Meter (qm)	JP6	JP7	Actual address (qm)
1			0x00
2		✓	0x01
3	✓		0x02
4	✓	✓	0x03

✓ = apply a soldering point

3.1.3 DATA bus

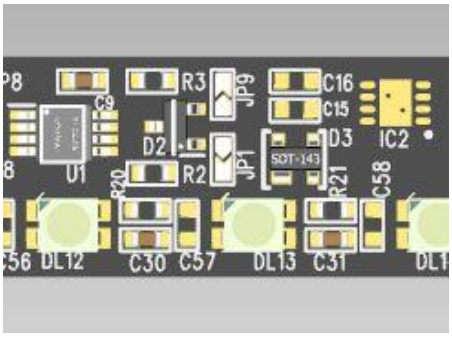
Data bus adopted has been optimized to be used as balanced 120 Ohm. So be careful to connect master as closed as possible to the first module, and if needed, set properly head and tail resistors (when MASTER comes with no output resistor). All modules come with 120 Ohm resistors both in head and in tail location, just close **JPI** and/or **JPO** jumpers.

Fig. 4 – JPI and JPO to setup termination resistors



Fail safe resistor are also assembled. Just close jumpers JP1 and JP9 if needed.

Fig. 5 – Fail safe resistors setup



In high added value boards, a +24V bus is also available. Can be used as master to devices manner and viceversa, or simply as a synchronization channel.

3.1.4 Communication configuration

RS485 half-duplex bus communication has been adopted. Parameters are as follow:

Table 8. Communication parameters

Parameter	Value
BaudRate	115200
DataBits	8
StopBits	1
Parity	None
Handshake hardware	None
Handshake software	None

3.2 Pin usage

3.2.1 MASTER to Module connection

When outcome from factory, first module is wired with M12 male – 5 poles 22cm cable with following connection:

Table 9. Wire to Module connection

Cable color	PAD	Signal
BLUE	P1	Gnd
WHITE	P2	RS485 +
BLACK	P3	RS485 -
BROWN	P4	+24V supply
GRAY (*)	P5	RFU

(*) optional

3.2.2 Module to Module – same profile

In order to assemble modules in cascade mode, same pair of signals are available in tail/head position. Split pairs allow electrical connections of contiguous boards using 4 x 50mm long wires

Fig. 6 – Output <> Input electrical signals

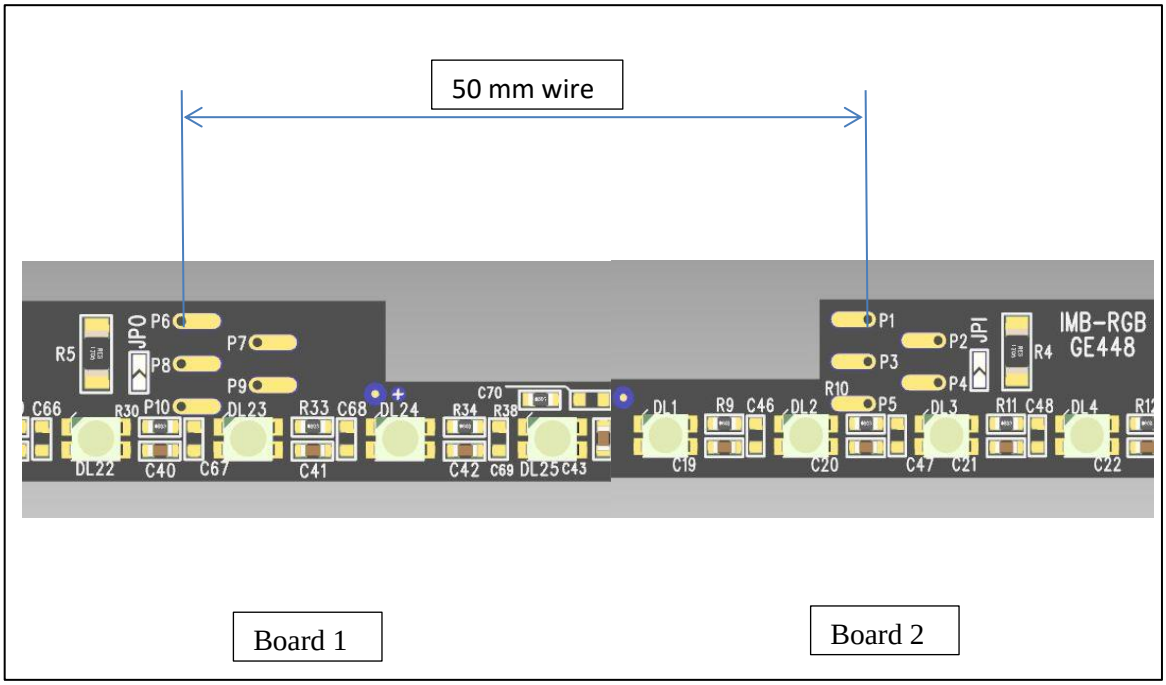


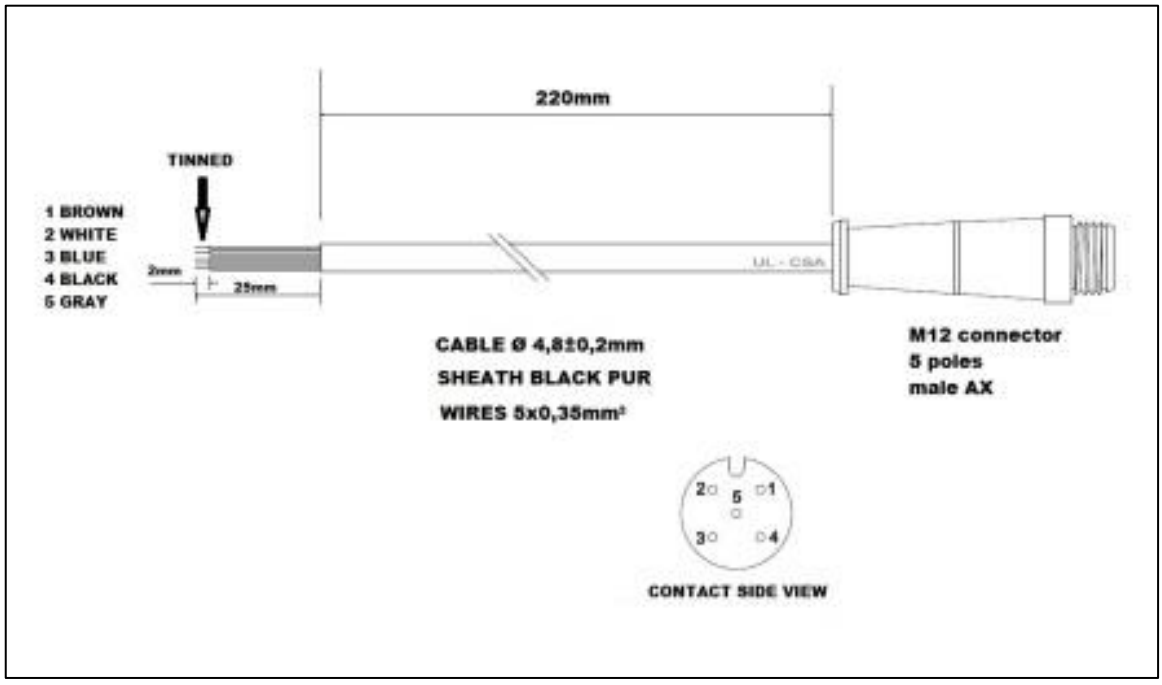
Table 10. Output <> Input pin assignment

From PAD	To PAD	Signal
P6	P1	Gnd
P7	P2	RS485 +
P8	P3	RS485 -
P9	P4	+24V supply
P10	P5	RFU

3.2.3 Module to Module – different profiles

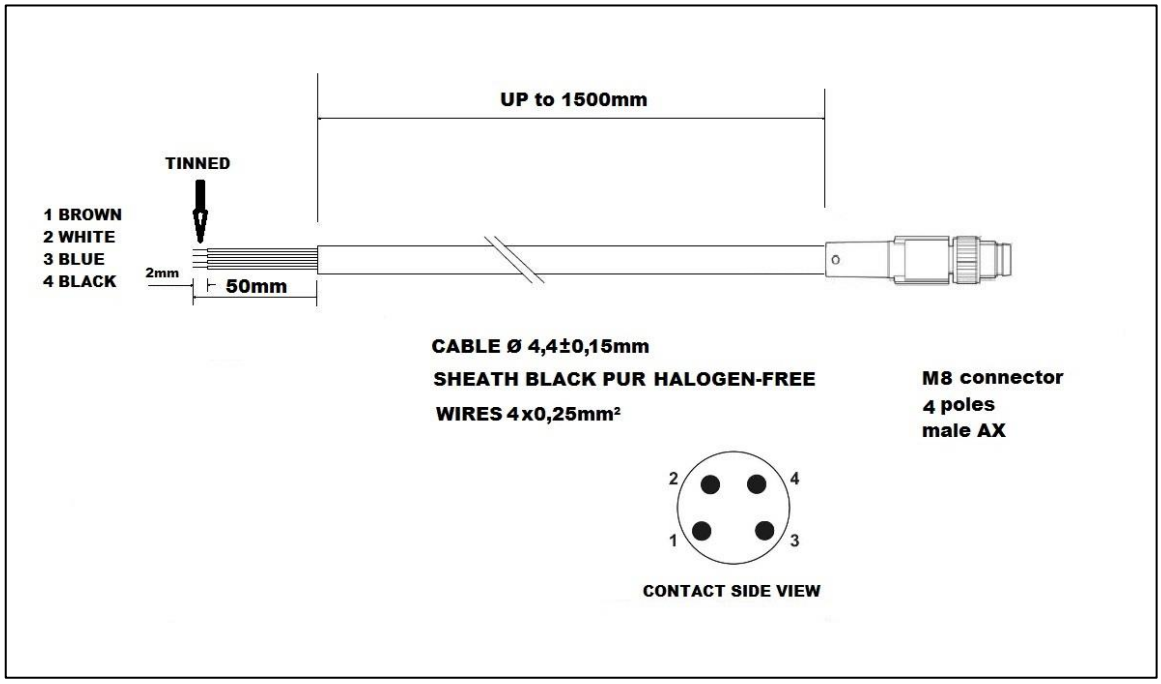
Different bars can be assembled to obtain a single-continuous bar by means of head-tail wires and M12 female(tail) male(head). As follows, color cables and M12 connector pins:

Fig. 7 – Wire colors and M12 5 pole connector



In saving space application, M8 connector with high flexible halogen free PUR is available:

Fig. 8 – Wire colors and M8 4 pole connector



3.3 Designing considerations

In order to keep design simple and transmission efficient, a 8 byte protocol to and from device has been adopted. With broadcasting exception, a receiver has always to confirm the reception of its messages. Communication is half duplex and only between a sender (MASTER) and a device (SLAVE). MASTER sends a message to a SLAVE and waits for a confirmation, which is blocking with a proper timeout. No direct communication between SLAVES is allowed. Broadcasting is allowed. In this case, MASTER does not need any confirmation back.

3.3.1 TO device protocol

Head	Mitt	Dest	Cmd	Byte1	Byte2	Byte3	CRC

Head: one byte fixed field, **0x80**;
Mitt: one byte sender address, in range **0x00…0x0F**;
Dest: one byte receiver address, in range **0x10…0xEF**;
Cmd: 7 bit command (BIT7…BIT1) + BIT0=DATA0 (1 bit);
Byte1…Byte3: DATA1…DATA3 to be transferred to device
CRC: simple XOR of all bytes but Head

3.3.2 FROM device protocol

Head	Mitt	Dest	Cmd	Byte1	Byte2	Byte3	CRC

Head: one byte fixed field, **0x90**;
Mitt: one byte sender address, in range **0x10…0xEF**;
Dest: one byte receiver address, in range **0x00…0x0F**;
Cmd: 7 bit command (BIT7…BIT1) + BIT0= DATA0 (1 bit);
Byte1…Byte3: DATA1…DATA3 read from device;
CRC: simple XOR of all bytes but Head

3.3.3 Application example

4. Command list

Several commands have been hardcoded to interact with modules and obtain different results. Available are as follows. Let's assume:

1) MASTER has address 0x00
2) SLAVE has address 0x10
3) Bn denotes the BYTE number 'n'
4) bm denotes the BIT number 'm'
5) A BYTE is represented as (HIGH) b7 b6 b5 b4 b3 b2 b1 b0 (LOW)
6) b0 can be linked to LED1, B1b7 can be linked to LED2, B1b6 can be linked to LED3… B3b0 can be linked to LED25

Each message has to be sent to each SLAVE with its proper address.
Each slave answers back to each request.
In event of broadcast messages, no response is asserted.
To execute a broadcast command, just replace DEST field with 0xFF.
Not all commands are allowed to broadcast communication.
If so, a ✓ is placed, an ✘ otherwise.

4.1 SET LED COLOR_0/COLOR_1 0x00

Switches LED in selected position on COLOR_0 (if 0) or COLOR_1 (if 1).
Response is LED status.

Request

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x80	0x10	0x00	0x0b0	B1	B2	B3	CRC

b0=1:LED1 set to COLOR_1
b0=0:LED1 set to COLOR_0
Bnbm=1: LED(F(n,m) as assumption 6)) COLOR_1
Bnbm=0: LED(F(n,m) as assumption 6)) COLOR_0
CRC: 0x10 xor 0x0b0 xor B1 xor B2 xor B3

Response

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x90	0x00	0x10	0x0b0	B1	B2	B3	CRC

NOTE: if not specified, by default COLOR_0=BLACK, COLOR_1=WHITE

Example:
REQ: 0x80 0x10 0x00 0x01 0x55 0x55 0x55 0x44
Turns COLOR_1 odd position LEDs and COLOR_0 even position LEDs
RESP: 0x90 0x00 0x10 0x01 0x55 0x55 0x55 0x44
Response is LED status

BROADCAST: ✓ .

4.2 PWM adjust 0x02 (NOT IMPLEMENTED)

LED intensity of each module can be adjusted by means of PWM regulator. 100% PWM along with 50 lux points of FINE STEP are standard programmed (maximum intensity). FINE STEP could be useful to reduce max light intensity to a desired value, then dimmed by PWM.

Request

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x80	0x10	0x00	0x02	PWM	FINE	B3	CRC

PWM: percentage of LED luminosity (0…100 % available)
FINE: number of lux points per PWM point (1…50 available)
B3: does not care
CRC: 0x10 xor 0x02 xor PWM xor FINE xor B3

Response

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x90	0x00	0x10	0x02	PWM	FINE	B3	CRC

Example:

REQ: 0x80 0x10 0x00 0x02 0x32 0x0A 0x55 0x7F
Sets up 50% duty cycle LED luminosity over 10 x 100 lux points
RESP: 0x90 0x00 0x10 0x02 0x32 0x0A 0x55 0x7F
Response is 50% LED duty cycle over 10 x 100 lux points

BROADCAST: ✓ .

4.3 FIRMWARE version 0x04

Each module executes its own firmware. User can check their version.

Request

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x80	0x10	0x00	0x04	B1	B2	B3	CRC

B1,B2, B3: do not care
CRC: 0x10 xor 0x04 xor B1 xor B2 xor B3

Response

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x90	0x00	0x10	0x04	MAJOR	MINOR	B3	CRC

Example:

REQ: 0x80 0x10 0x00 0x04 0x55 0x55 0x55 0x41
Request for firmware version (MAJOR.MINOR)
RESP: 0x90 0x00 0x10 0x04 0x01 0x01 0x55 0x41
Response is firmware version 1.1

BROADCAST: ✘ .

4.4 REPLY delay 0x06

A reply delay to host is configurable, if requested. A 3 bytes value, in us, is possible.

Request

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x80	0x10	0x00	0x06	B1	B2	B3	CRC

B1, B2, B3: 3 bytes wide delay in us (0 value = no delay)
CRC: 0x10 xor 0x06 xor B1 xor B2 xor B3

Response

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x90	0x00	0x10	0x06	B1	B2	B3	CRC

Example:

REQ: 0x80 0x10 0x00 0x06 0x0F 0x42 0x40 0x1B
Request for reply delay of 1s (1000000us)
RESP: 0x90 0x00 0x10 0x06 0x0F 0x42 0x40 0x1B
Responses (and furthers) are delayed of 1s

BROADCAST: ✓ .

4.5 RESTORE default COLOR_X 0x0E

COLOR_X can be reloaded with default values at any time, accordingly with list below:

COLOR_0: BLACK (RED=0, GREEN=0; BLUE=0)
COLOR_1: WHITE (RED=255, GREEN=255; BLUE=255)
COLOR_2: RED (RED=255, GREEN=0; BLUE=0)
COLOR_3: ORANGE (RED=255, GREEN=50; BLUE=0)
COLOR_4: YELLOW (RED=255, GREEN=150; BLUE=0)
COLOR_5: GREEN (RED=0, GREEN=255; BLUE=0)
COLOR_6: BLUE (RED=0, GREEN=0; BLUE=255)
COLOR_7: PURPLE (RED=255, GREEN=0; BLUE=255)

Request

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x80	0x10	0x00	0x0E	B1	B2	B3	CRC

B1, B2, B3: do not care
CRC: 0x10 xor 0x0E xor B1 xor B2 xor B3

Response

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x90	0x00	0x10	0x0E	B1	B2	B3	CRC

Example:

REQ: 0x80 0x10 0x00 0x0E 0x00 0x00 0x00 0x1E
Request for set default COLOR_X
RESP: 0x90 0x00 0x10 0x0E 0x00 0x00 0x00 0x1E
Response for set default COLOR_X

BROADCAST: ✓ .

4.6 SET COLOR_X 0x10+X*2

It is possible to preset an array of 8 different colors:

COLOR_0 (CMD=0x10)
COLOR_1 (CMD=0x12)
COLOR_2 (CMD=0x14)
…
COLOR_7 (CMD=0x1E)

With RGB – 24bit notation

Request

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x80	0x10	0x00	CMD	RED	GREEN	BLUE	CRC

RED, GREEN, BLUE: in range 0…255
CRC: 0x10 xor CMD xor RED xor GREEN xor BLUE

Response

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x90	0x00	0x10	CMD	RED	GREEN	BLUE	CRC

Example:

REQ: 0x80 0x10 0x00 0x10 0xFF 0x00 0x00 0xFF
Sets COLOR_0 to true red
RESP: 0x90 0x00 0x10 0x10 0xFF 0x00 0x00 0xFF
Confirms COLOR_0 set to true red

BROADCAST: ✓ .

4.7 SET LED_X COLOR 0x42+(X-1)*2

Each RGB LED can be turned with RGB color individually. A command is available for each led in range 1...25. Command list is as below:

LED1: CMD=0x42 (only buffer update), CMD=0x43 (buffer update and show)
LED2: CMD=0x44 (only buffer update), CMD=0x45 (buffer update and show)
…
LED25: CMD=0x72 (only buffer update), CMD=0x73 (buffer update and show)

Request

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x80	0x10	0x00	CMD	RED	GREEN	BLUE	CRC

RED, GREEN, BLUE: in range 0…255
CRC: 0x10 xor CMD xor RED xor GREEN xor BLUE

Response

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x90	0x00	0x10	CMD	RED	GREEN	BLUE	CRC

NOTE: CMD commands explained above just fill a memory image of the bar. The idea is first to create the whole memory image of the bar, then show it on bar led. This is accomplished sending the last CMD+0x01, forcing first the last filling, then the proper show.

Example:

REQ: 0x80 0x10 0x00 0x43 0xFF 0x00 0x00 0xAC
Sets LED1 to true red then show
RESP: 0x90 0x00 0x10 0x43 0xFF 0x00 0x00 0xAC
Confirms LED1 set to true red then show

BROADCAST: ✓ .

4.8 SET LED_X to COLOR_Y 0x80…0x8B

Each LED_X (where X is in range 1...25) can be turned ON with color stored in COLOR_Y array (with Y in range 0…7). So that, to complete the 25 led bar, 3 messages are needed:

1) 25 bit list of LSB of Y index of each LED (CMD=0x80 + twobit_spur_1)
2) 25 bit list of MID of Y index of each LED (CMD=0x84 + twobit_spur_2)
3) 25 bit list of MSB of Y index of each LED (CMD=0x88 + twobit_spur_3)

The terms twobit_spur_z, with z in range 1…3, have values:

0 if 25th bit is 0 and no led update is requested
1 if 25th bit is 1 and no led update is requested
2 if 25th bit is 0 and led update is requested
3 if 25th bit is 1 and led update is requested

Led updated should be requested only when the last of the 3 messages is sent

Request

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x80	0x10	0x00	CMD	BYTE1	BYTE2	BYTE3	CRC

BYTE1…3: in range 0…255
CRC: 0x10 xor CMD xor BYTE1 xor BYTE2 xor BYTE3

Response

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x90	0x00	0x10	CMD	BYTE1	BYTE2	BYTE3	CRC

Example: we want to apply COLOR_Y to LED_X in the following way.

COLOR_0 to LED_1, LED_9, LED_17, LED_25
COLOR_1 to LED_2, LED_10, LED_18
COLOR_2 to LED_3, LED_11, LED_19
…
COLOR_7 to LED_8, LED_16, LED_24

LED_	25	…	12	11	10	9	8	7	6	5	4	3	2	1
LSB	0	…	1	0	1	0	1	0	1	0	1	0	1	0
MID	0	…	1	1	0	0	1	1	0	0	1	1	0	0
MSB	0	…	0	0	0	0	1	1	1	1	0	0	0	0
COLOR_	**0**	**…**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

twobit_spur_1= twobit_spur_2=0 => CMD1=0x80, CMD2=0x84
twobit_spur_3=2 => CMD3=0x88+0x02=0x8A

REQ: 0x80 0x10 0x00 0x80 0xAA 0xAA 0xAA 0x3A
Sets LSB of COLOR_Y bit list (but visualizes)
RESP: 0x90 0x00 0x10 0x80 0xAA 0xAA 0xAA 0x3A
Confirm
REQ: 0x80 0x10 0x00 0x84 0xCC 0xCC 0xCC 0x58
Sets MID of COLOR_Y bit list (but visualizes)
RESP: 0x90 0x00 0x10 0x84 0xCC 0xCC 0xCC 0x58
Confirm
REQ: 0x80 0x10 0x00 0x8A 0xF0 0xF0 0xF0 0x6A
Sets MSB of COLOR_Y bit list (then visualizes)
RESP: 0x90 0x00 0x10 0x8A 0xF0 0xF0 0xF0 0x6A
Confirm

The final result is (with default COLOR_Y):

LED_	25	…	12	11	10	9	8	7	6	5	4	3	2	1
COLOR		…												

BROADCAST: ✓ .

4.9 LED UPDATE 0x8E

Led updated should be requested only when the last of the 3 messages is sent. But it is possible to send more than one triples of LED commands on several modules, then finalize their setup by one only update command. This to avoid refresh slope delays.

Request

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x80	0x10	0x00	CMD	BYTE1	BYTE2	BYTE3	CRC

BYTE1...3: do not care
CRC: 0x10 xor CMD xor BYTE1 xor BYTE2 xor BYTE3

Response

HEAD	Dest	Mitt	Cmd	Byte1	Byte2	Byte3	CRC
0x90	0x00	0x10	CMD	BYTE1	BYTE2	BYTE3	CRC

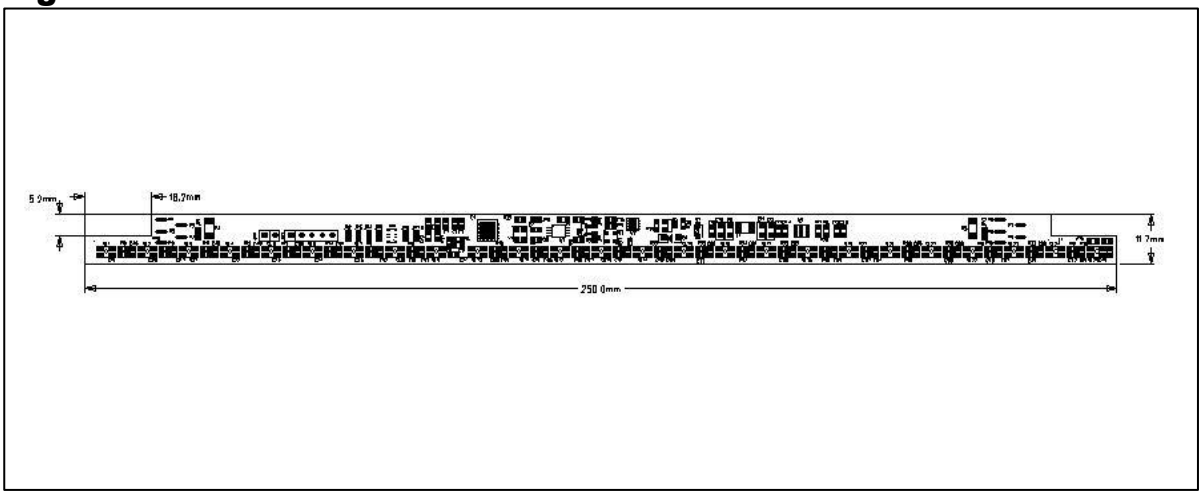
Example:

REQ: 0x80 0xFF 0x00 0x8E 0xAA 0xAA 0xAA 0xDB

BROADCAST: ✓ .

5.Mechanical data

Fig. 9 – PCB dimensions



6.Regulatory compliance

All Genesi LUX® products are RoHS compliant.

7. Ordering information

Several commercial models are available with different features, upon customer's specific request. Please contact sales.

8. Revision history

Table 12. Revision history

Rev	Date	Author	Description	Approved by
1.0	15/06/2020	Giuliano Calzolari	Preliminary release	Mauro Munari
1.1	26/02/2021	Giuliano Calzolari	Added LED UPDATE command	Mauro Munari

9. Disclaimers

Please Read Carefully

Information in this document is provided solely in connection with Genesi LUX® products. Genesi LUX® and its subsidiaries reserve the right to make changes, corrections, modifications or improvements to this document, the products and services described herein at any time, without notices.

All Genesi LUX® products are sold pursuant to Genesi LUX® terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the Genesi LUX® products and services described herein, and Genesi LUX® assumes no liability whatsoever relating to the choice, selection or use of the Genesi LUX® products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by Genesi LUX® for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN GENESI LUX®'S TERMS AND CONDITIONS OF SALE GENESI LUX® DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF GENESI LUX® PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY TWO AUTHORIZED GENESI LUX® REPRESENTATIVES, GENESI LUX® PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. GENESI LUX® PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of Genesi LUX® products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by Genesi LUX® for the Genesi LUX® product or service described herein and shall not create or extend in any manner whatsoever, any liability of Genesi LUX®.

Genesi LUX® and Genesi LUX® logo are trademarks or registered trademarks of Genesi Elettronica Srl in various countries.Information in this document supersedes and replaces all information previously supplied.

The Genesi LUX® logo is a registered trademark of Genesi Elettronica Srl. All other names are the property of their respective owners.

GENESI ELETTRONICA Srl
Via Bachelet, 3
41057 – Spilamberto – MODENA (ITALY)
Phone +39 059 785566
Cap.Soc. 25000 Euro i.v.- C.F. P.IVA e Cod.It : 02100940366 – C.C.I.A.A. 02100940366 – Rea n. 265938