



TRM Project

TRM-LED

Ether**CAT**® 

Hardware Architecture

User Manual

Technical Documentation

DOCUMENT DATA

Data	06/10/2026
Revision	3
FileName	TRM-LED rev.3.pdf
Document type	User manual
Authors	UT – TPA Srl
Department	DIP
Notes	Customer document
	This documentation is the property of TPA s.r.l. Any unauthorized duplication is forbidden. The Company reserves the right to modify the content of the document at any time."

REVISIONS

Revision Number	Data	List of changes
Rev.0	12/12/2025	First delivering
Rev.1	30/07/2026	According to code v1.0
Rev.2	21/09/2026	Change fotos
Rev.3	06/10/2026	Info about cable lengths and class A

INDEX

DOCUMENT DATA.....	2
REVISIONS.....	2
CONTENTS	4
1 FUNCTIONAL SPECIFICATION.....	5
2 MECHANICAL DATA	7
3 ETHERCAT COMMUNICATION.....	8
4 CONNECTIONS.....	9
4.1 Direct connection to a LED strip.....	9
4.2 Connection by the LED Extender	9
5 OBJECT DICTIONARY AND PROTOCOL.....	10
5.1 Standard Object Dictionary	10
5.3 PDO's list.....	13
5.4 LED Protocol and data Format.....	14
5.5 Data Protocol over ECAT	15
6 FRAME FORMAT	17
6.1 Command frame format	17
6.1.1 NUM and A[3:0]	17
6.1.2 CFC - Command Function Code and B[3:0].....	17
6.1.3 CHECKSUM	18
6.2 Answer frame format.....	18
6.2.1 NUM and A[3:0]	18
6.1.2 AFC – Answer Function Code and B[3:0].....	18
6.2.3 CHECKSUM	19
7 HANDSHAKE	20
7.1 NOT-ACKNOWLEDGE (AFC = 0x0).....	20
8 COMMANDS.....	21
8.1 START-STOP (CFC=AFC=0x1).....	21
8.2 WRITE OUTPUT - 1 COLOR FOR 3 SEGMENTS (CFC=AFC=0x2)	23
8.3 SET STRIP – SET THE STRIP WITH A COLOR (CFC=AFC=0x3).....	25
8.4 READ/WRITE STATUS COMMAND (CFC=AFC=0x5)	27
8.5 SHIFT FORWARD/BACKWARD (CFC=AFC=0xA).....	29
8.6 WRITE OUTPUT - 2 COLORS FOR 2 SEGMENTS (CFC=AFC=0xB).....	31
8.7 WRITE OUTPUT - 1 COLOR FOR 2 EXTENDED SEGMENTS (CFC=AFC=0xC).....	33
8.8 COLOR MAP WRITING (CFC=AFC=0xD).....	35
8.9 WRITE OUTPUT – 3 SEGMENTS WITH 3 COLORS BY MAP (CFC=AFC=0xE)...	37
8.10 ALIM – check +5V Power Supply	38
9 TRM-LED VIEW.....	39



CONTENTS

This document is the User Manual for TRM-LED device. TRM-LED belongs to the TRM series of devices by TPA s.r.l. which are bus field devices to remote the CNC action on a tool machinery or, generally, in a controlled system; the devices are connected to a standard EtherCAT bus. TRM-LED is a module which provides 2 independent serial digital output to connect 2 WS2812B-protocol multi-LED stripes. This document shows how to use TRM-LED, to connect it and all the technical specifications regarding the product.

This document is the property of TPA Srl.

Not authorized duplication is forbidden.

This description is only intended for the use of trained specialists in control and automation engineering who are invited to always use the currently valid documentation.

It is recommended that the documentation (notes, explanations, diagrams, ecc.) are followed when installing and commissioning these components.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The Company reserves the right to revise and modify the current documentation at any time.

The Company doesn't respond of not-correct or improper use of the device.

Safeguard clause

If any information about this device will result not clear, please do not hesitate to take contacts with TPA.srl Service.



Albatros - Logo



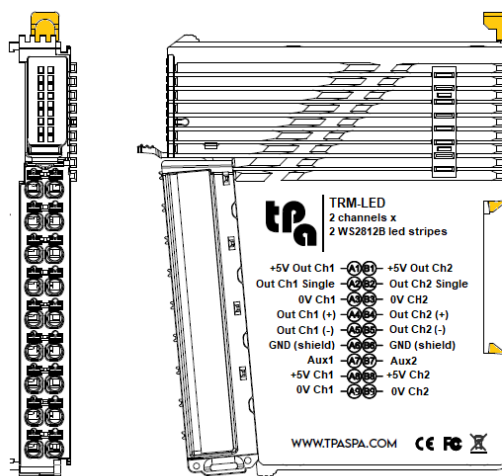
TPA S.r.l - Logo

1 FUNCTIONAL SPECIFICATION

TRM-LED is a module which provides 2 digital channels for driving multicolor led streaps. Digital channels follow the WS2812B standard protocol with 2 lines with digital streams about 800KHz. In alternative, the same protocol is provided by 2 serial RS485 lines (800KHz) for remoting the led stripes; next to the stripes, a LED Extender device (by TPA) will reconvert signal to a single ended format.

TRM-LED needs to be connected to the bus coupler TRM-ECAT (directly or in chain by other devices of the TRM serie). By this, EtherCAT link and logical power supply are established, conforming to IEC61158. No other application is allowed. TRM-LED is a Class-A device for industrial environments.

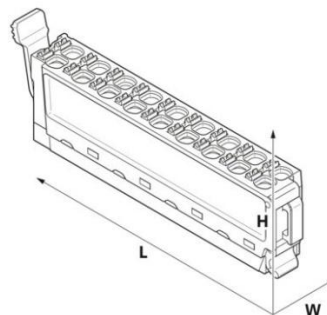
Power Supply (+5V) must be provided by the external connector; it remains discoupled from the logical power supply, provided by the internal bus connection from TRM_ECAT.



On the right side, a gray silkscreen is printed and it reports:

- the device Name and its main function
- the batch production number
- the consortium EtherCAT logo to specify the environment of use and compatibility
- the TPA reference - TPA's logo, QR code and site address
- CE certification logo
- a simplified schematic for input connections

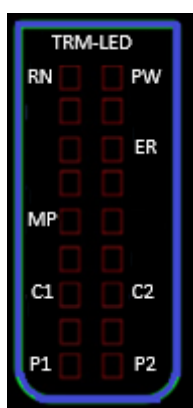
All the connections are reported to a 18-positions connector provided with the device. The connector used is.



HSCP-SP 1,5-18-IO-7035 color gray

producer code 1569965 - Phoenix

	PIN TOP LEFT	PIN TOP RIGHT	
+5V Ch1 out	A1	B1	+5V Ch2 out
Out Ch1 Single Ended	A2	B2	Out Ch2 Single Ended
0V Ch1	A3	B3	0V Ch2
Out Ch1 (+)	A4	B4	Out Ch2 (+)
Out Ch1 (-)	A5	B5	Out Ch2 (-)
GND (shield)	A6	B6	GND (shield)
Aux1	A7	B7	Aux2
+5V Ch1	A8	B8	+5V Ch2
0V Ch1	A9	B9	0V Ch2
	PIN BOTTOM LEFT	PIN BOTTOM RIGHT	



LED		LED	
RN	ECAT link status	PW	Power Supply ECAT
		E	Error status
MP	Microprocessor active		
C1	Channel 1 Activity	C2	Channel 2 Activity
P1	Power Supply +5V Ch1	P2	Power Supply +5V Ch2

On the top of the frontal image of TRM-LED a list of LED indicators is set.

LED behaviour	TRM-LED State condition
Extinguished	INIT State
Blinking (regular)	PreOP State
Single flash	SafeOP State
Bright	Operational

Regarding the RN TRM-LED status led

Other LEDs' behaviour:

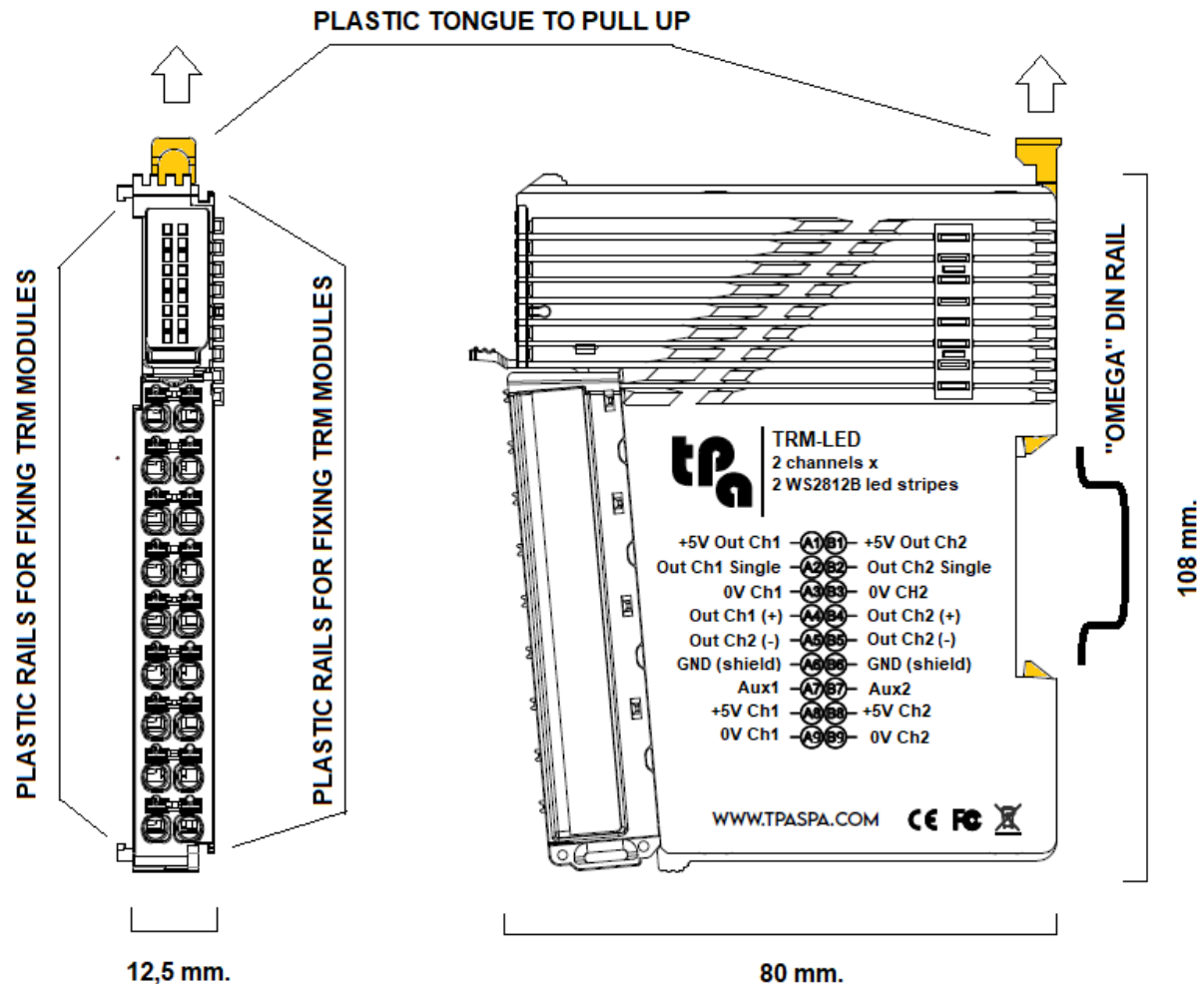
- *PW led brights on when EBUS power supply is OK.*
- *P1 led brights on when external +5V power supply for channel 1 is OK.*
- *P2 led brights on when external +5V power supply for channel 2 is OK.*
- *CH1 led blinks on (impulse) when channel 1 is ACTIVE (data transmission).*
- *CH2 led blinks on (impulse) when channel 2 is ACTIVE (data transmission).*

Electrical parameters	Values
Signals	2 x Single Ended (CMOS/TTL) or RS485
Nominal output frequency	800KHz (WS2812B)
Max. output current	20mA (short circuit proof) per channel
Operating Temperature	-20°C – 60°C
Storage Temperature	-40°C – 60°C
Relative humidity	5 – 95% (not condensating)
Dimensions	108x12,5x80mm.
Net weight	60g.
EMC	CE/FCC
Protection level	IP20

2 MECHANICAL DATA

TRM-LED module can be installed onto a metallic rail DIN 60715, with "OMEGA" profile, model 7,5mm. or 15mm.

To add a TRM-LED device to a TRM-ECAT or to another TRM module in a blockchain, let's introduce it along the right side of the TRM-LED module. There are two plastic rails on a TRM device that help to get the correct position side by side with the last TRM device. Then, throw up the orange tongue to fix it to the DIN rail.



Dimension	Value	Notes
Height	108 mm.	Let's consider the extension of the tongue and the possibility to have a min. extended space to handle it
Dept	80 mm.	Let's consider the extension of the connectors and the curves of the cables
Thickness	12,5 mm.	

3 ETHERCAT COMMUNICATION

The TRM-LED module is compliant with standard EtherCAT normed by ETG in IEC61158.

TRM-LED is addressed as a station in the EtherCAT network; it can run only in an ECAT network.

Other TRM's serie devices can be mechanically and electrically joined to TRM-LED, to constitute a blockchain, as shown.



Data communication into the blockchain is supported by E-BUS, a local bus derived from ECAT. The communication is full-duplex, real time clocked and based on 100Base-Tx exchange on internal E-BUS. This bus is coupled toward ECAT by a TRM-ECAT modul. TRM-LED cannot be used not joined to TRM-ECAT.

The max exchange rate (the real Time Clock of the network) can be accorded in this range: 250µs, 500µs, 1ms, 2ms, 4ms or more. It is a Master station dependent parameter.

Output signals are updated in accordance with the communication update.

Methods to access and to run it are provided by a description ESI file (TRM-LED.xml is provided for setting up the device into the control system).

TRM-LED presents a COE protocol with a Dictionary of objects.

The ECAT control unit sees it as a node, a Hardware Station point, with configurable objects and PDOs. Depending by the number of LEDs to be driven, PDO's configuration can be extended till max 1000 led for channel. Each LED needs a 3-byte RGB information.

Please, look at Object Dictionary chapter.

4 CONNECTIONS

The device can be used with 2 kinds of circuits:

1. directly connecting LED strip lights to TRM-LED
2. connecting LED strips by the external LED Extender device (by TPA)

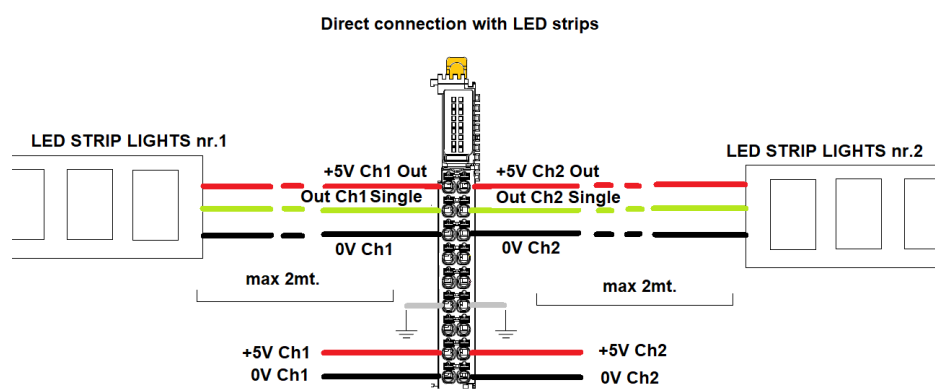
Also Per i cavi di alimentazione è raccomandato utilizzare almeno cavi AWG12 che portino 15A.

For power supply cables, please use cables with a conduction section for more than 15A.

For digital signals, don't exceed over 2mt. for direct connection to LED strip lights.

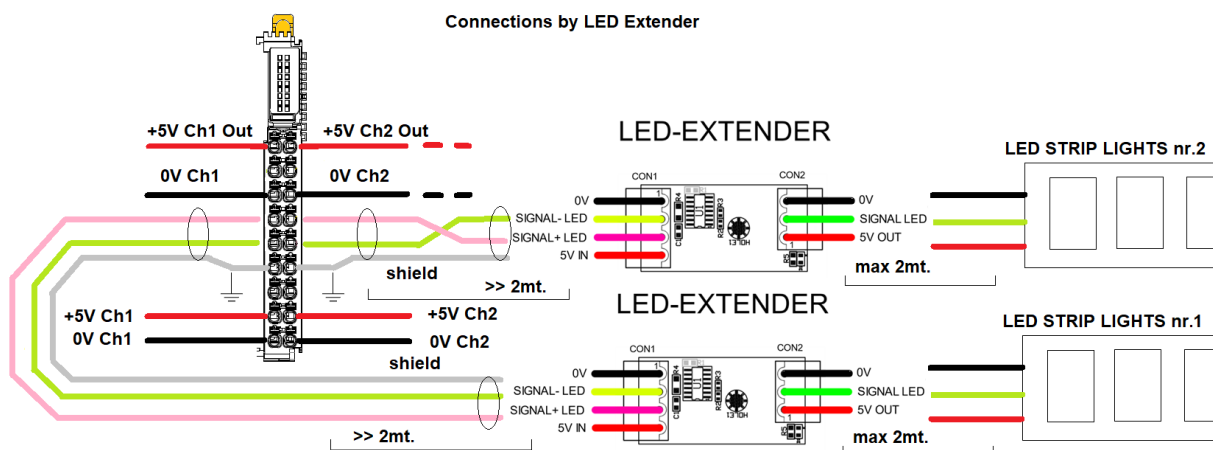
Over this length, it's better to use a configuration with LED Extender. For LED Extender configuration, please use twisted pair differential connection with shield.

4.1 Direct connection to a LED strip



Laboratory proves have accerted the sensibility of this cabling method regards to EMC. To preserve the signal integrity, please do not exceed the 2m. max length limit. Maybe useful to protect the signal by a shield connected to ground.

4.2 Connection by the LED Extender



For this kind of connection, it's mandatory to use twisted pair shielded cable. To preserve the signal integrity, please do not exceed 30m. max length limit. Maybe useful to introduce a termination resistor between LED(+) and LED(-) terminals at LED-EXTENDER (120Ohm 1W). Protect the signal by a shield, connected to ground. Do not exceed 2m. length back LED-EXTENDER.

For signal integrity, please keep in mind the signal main frequency is 800KHz, with a second armonic frequency at 2,4MHz.

The application has shown to be sensible to EM noise in the range 10-15MHz.

5 OBJECT DICTIONARY AND PROTOCOL

The device TRM-IO shows a Object Dictionary, made by standard COE Objects and specific Objects. All the available objects are described in TRM-IO ESI file (TRM-IO.xml).

Communication can be implemented both as PDOs or SDOs.

- SDO communication is made for settings and for seldom commands.
- PDO communication is made for a continuous command's rate. It implements also a dummy command (NULL command) for no-activity times.

For SDO communication, Mailbox service is enabled.

5.1 Standard Object Dictionary

0x1000 Device Type

Index(hex)	Name	Data	Data type	Marker	Default
1000.0	Device Type	EtherCAT type of the device	UINT32	RO	TBD

0x1008 Device Name

Index(hex)	Name	Data	Data type	Marker	Default
1008.0	Device Name	EtherCAT name of the device	STRING	RO	TRM-LED

0x1009 Hardware version

Index(hex)	Name	Data	Data type	Marker	Default
1009.0	Device HW version	EtherCAT data about HW version	STRING	RO	1.00

0x100A Software version

Index(hex)	Name	Data	Data type	Marker	Default
100A.0	Device SW version	EtherCAT data about SW version	STRING	RO	1.00

0x1018 Id of the node in ECAT net

Index(hex)	Name	Data	Data type	Marker	Default
1018.0	Identity	To identify the device from station information	UINT8	RO	0x04
1018.01	Vendor ID	EtherCAT from the station supplier ID	UINT32	RO	0x000002EA
1018.02	Product Code	EtherCAT from the station product ID	UINT32	RO	TBD
1018.03	Revision	EtherCAT from the station revision NrD	UINT32	RO	0x00000100
1018.04	Serial Number	EtherCAT from the station Serial Nr	UINT32	RO	0x00000000

0x1C00 Sync Manager type

Index(hex)	Name	Data	Data type	Marker	Default
1C00.00	Sync Manager nr.	Using the Sync Manager	UINT8	RO	0x04
1C00.01	Sync Mng.1	EtherCAT from the station supplier ID	UINT8	RO	0x01
1C00.02	Sync Mng.2	EtherCAT from the station product ID	UINT8	RO	0x02
1C00.03	Sync Mng.3	EtherCAT from the station revision NrD	UINT8	RO	0x03
1C00.04	Sync Mng.4r	EtherCAT from the station Serial Nr	UINT8	RO	0x04

0x1C12 Assignment of cyclic RxPDO

Index(hex)	Name	Data	Data type	Marker	Default
1C12.00	Subindex 0 Range [1]	How many Channels are available	UINT8	RW	0x01
1C12.01	RxPDO1	RxPDO from ECAT master	UINT16	RW	0x1600 fixed

0x1C13 Assignment of cyclic TxPDO

Index(hex)	Name	Data	Data type	Marker	Default
1C13.00	Subindex 0 Range [1]	How many Channels are available	UINT8	RW	0x01
1C13.01	TxPDO1	TxPDO 1 to ECAT master	UINT16	RW	0x1A00 fixed

0x1C32 SM Output Parameters

Index(hex)	Name	Data	Data type	Marker	Default
1C32.00	Subindex 0	Synchronization parameters for output	UINT8	RW	0x20
1C32.01	Sync mode	Current synchronization mode: 0: Free Run 1: Synchron with SM 2 Event 2: DC-Mode - Synchron with SYNC0 Event 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0200
1C32.02	Cycle time	Cycle time (in ns): • Free Run: cycle time of the local timer • DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x00000000
1C32.03	Shift Time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000
1C32.04	Sync modes supported	Supported synchronization modes: • Bit0=1: Free Run is supported • Bit2=1: DC-Mode is supported (SYNC0)	UINT16	RO	0x0500
1C32.05	Min cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00000000
1C32.06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC Mode only)	UINT32	RO	0x00000000
1C32.08	Command	0: Measurement of the local cycle time is stopped 1: Measurement of the local cycle time is started The entries <u>0x1C32:03</u> , <u>0x1C32:05</u> , <u>0x1C32:06</u> , <u>0x1C32:09</u> , <u>0x1C33:03</u> , <u>0x1C33:06</u> , <u>0x1C33:09</u> are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000
1C32.09	Max delay	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000
1C32.0A	Sync0 time	Support only Freerun	UINT32	RO	0x00000000
1C32.0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC Mode only)	UINT32	RO	0x00000000
1C32.0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT32	RO	0x00000000
1C32.0D	Shift too short counter	Number of intervals between SYNC0 and SYNC1 events that are too short (DC Mode only)	UINT32	RO	0x00000000
1C32.20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC Mode only)	BOOLEAN	RO	0x00

0x1C33 SM Input Parameters

Index(hex)	Name	Data	Data type	Marker	Default
1C33.00	Subindex 0	Synchronization parameters for output	UINT8	RW	0x20
1C33.01	Sync mode	Current synchronization mode: 0: Free Run 1: Synchron with SM 2 Event 2: DC-Mode - Synchron with SYNC0 Event 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0200
1C33.02	Cycle time	Cycle time (in ns): • Free Run: cycle time of the local timer • DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F4240
1C33.03	Shift Time	Time between SYNC0 event and reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000
1C33.04	Sync modes supported	Supported synchronization modes: • Bit0=1: Free Run is supported • Bit2-3 = 01: DC-Mode is supported	UINT16	RO	0x0500
1C33.05	Min cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00000000
1C33.06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000
1C33.08	Command	0: Measurement of the local cycle time is stopped 1: Measurement of the local cycle time is started The entries <u>0x1C32:03</u> , <u>0x1C32:05</u> , <u>0x1C32:06</u> , <u>0x1C32:09</u> , <u>0x1C33:03</u> , <u>0x1C33:06</u> , <u>0x1C33:09</u> are updated with the	UINT16	RW	0x0000

		maximum measured values. For a subsequent measurement the measured values are reset			
1C33.09	Max delay	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000
1C33.0A	Sync0 time	Support only DC Sync0 & Freerun	UINT32	RO	0x00000000
1C33.0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC Mode only)	UINT32	RO	0x00000000
1C33.0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT32	RO	0x00000000
1C33.0D	Shift too short counter	Number of intervals between SYNC0 and SYNC1 events that are too short (DC Mode only)	UINT32	RO	0x00000000
1C33.20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC Mode only)	BOOLEAN	RO	0x00

5.3 PDO's list

TRM-LED exchanges PDOs with the net ECAT master. Each PDO is a datagram, a sort of box to communicate commands and data. For TRM-LED, RxPDO brings commands and data to the device; on the other hand, TxPDO brings answer and read data to the master.

The couple of RxPDO-TxPDO is called "Communication Channel".

TRM-LED provides one Communication Channel (the couple of PDOs, Objects 0x1600 – 0x1A00).

Object	Description	Assignment	Mapping
0x1600	Communication Channel 0 – RxPDO 0	Default, assigned in 0x1C12 suindex 1	No PDO mapping: From Obj: 0x7000.01 To Obj: 0x7000.16

Object	Description	Assignment	Mapping
0x1A00	Communication Channel 0 – TxPDO 0	Default, assigned in 0x1C13 suindex 1	No PDO mapping: From Obj: 0x6000.01 To Obj: 0x6000.16

Pay attention to the order of assignment! The increasing PDO's order must be observed (from 0x1600 to 0x1603, from 0x1A00 to 0x1A03); if the condition is not respected, a configuration error will be returned by TRM-LED.

Each RxPDO/TxPDO is so made:

0x1600 fixed RxPDO mapping of output data objects (0x7000 type)

Index(hex)	Name	Object (hex.dec)	Data type	Marker	Default
16n0.01	Entry 1	0x70n0.01	UINT8	RW	0x00
16n0.02	Entry 2	0x70n0.02	UINT8	RW	0x00
16n0.03	Entry 3	0x70n0.03	UINT8	RW	0x00
16n0.04	Entry 4	0x70n0.04	UINT8	RW	0x00
16n0.05	Entry 5	0x70n0.05	UINT8	RW	0x00
16n0.06	Entry 6	0x70n0.06	UINT8	RW	0x00
16n0.07	Entry 7	0x70n0.07	UINT8	RW	0x00
16n0.08	Entry 8	0x70n0.08	UINT8	RW	0x00
16n0.09	Entry 9	0x70n0.09	UINT8	RW	0x00
16n0.0A	Entry 10	0x70n0.10	UINT8	RW	0x00
16n0.0B	Entry 11	0x70n0.11	UINT8	RW	0x00
16n0.0C	Entry 12	0x70n0.12	UINT8	RW	0x00
16n0.0D	Entry 13	0x70n0.13	UINT8	RW	0x00
16n0.0E	Entry 14	0x70n0.14	UINT8	RW	0x00
16n0.0F	Entry 15	0x70n0.15	UINT8	RW	0x00
16n0.10	Entry 16	0x70n0.16	UINT8	RW	0x00

0x1A00 fixed TxPDO mapping of input data objects (0x6000 type)

Index(hex)	Name	Object (hex.dec)	Data type	Marker	Default
1An0.01	Entry 1	0x60n0.01	UINT8	RO	0x00
1An0.02	Entry 2	0x60n0.02	UINT8	RO	0x00
1An0.03	Entry 3	0x60n0.03	UINT8	RO	0x00
1An0.04	Entry 4	0x60n0.04	UINT8	RO	0x00
1An0.05	Entry 5	0x60n0.05	UINT8	RO	0x00
1An0.06	Entry 6	0x60n0.06	UINT8	RO	0x00
1An0.07	Entry 7	0x60n0.07	UINT8	RO	0x00
1An0.08	Entry 8	0x60n0.08	UINT8	RO	0x00
1An0.09	Entry 9	0x60n0.09	UINT8	RO	0x00
1An0.0A	Entry 10	0x60n0.10	UINT8	RO	0x00
1An0.0B	Entry 11	0x60n0.11	UINT8	RO	0x00
1An0.0C	Entry 12	0x60n0.12	UINT8	RO	0x00
1An0.0D	Entry 13	0x60n0.13	UINT8	RO	0x00
1An0.0E	Entry 14	0x60n0.14	UINT8	RO	0x00
1An0.0F	Entry 15	0x60n0.15	UINT8	RO	0x00
1An0.10	Entry 16	0x60n0.16	UINT8	RO	0x00

5.4 LED Protocol and data Format

The data format is how data are organized into PDOs and their specific meaning. TRM-LED decodes received data (RxPDOs) and executes the commands, driving the LED strips.

In the same way, TxPDOs are sent back to the ECAT master to answer to the forwarded commands or to return required data from the device.

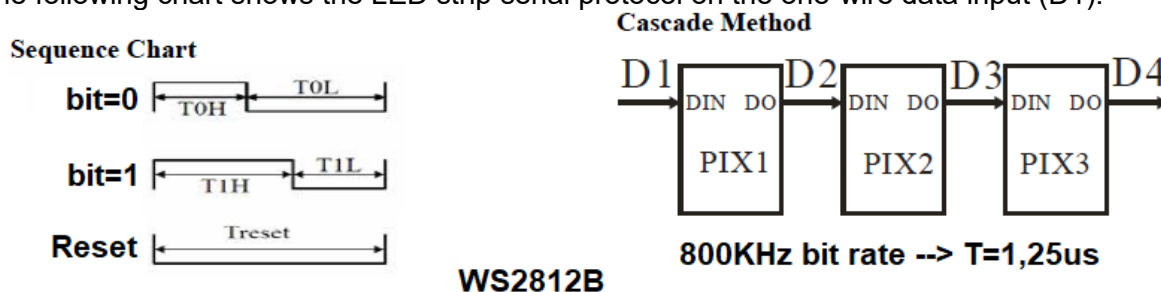
LEDs into strips are driven by internal ICs which implement a serial one-wire protocol, following the standard WS2812B. Data exchange happens only in writing direction:

TRM-LED → LED Strips

Three wires are used by strips:

- LED power supply +5V (can be sourced from TRM-LED +5Vout max 10A, or directly from the main power supply by-passing the TRM-LED)
- 0Volt power supply return-back
- One-wire serial signal terminal (WS2812B). It can be tied directly from TRM-LED (Out Ch1-2 Single) or passing by LED-Extender (Signal Led). See chapter 4.

The following chart shows the LED strip serial protocol on the one-wire data input (D1):



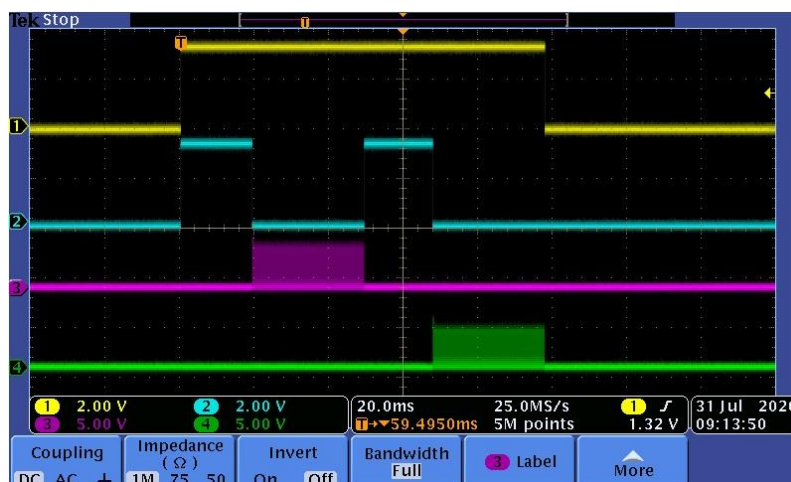
For updating a 1000 LED Strip, let's estimate the execution time.

The execution time is composed by three terms:

- The first time is the time-to-command T_c and depends by the complexity of the pattern to light on. Generally, it's about few milliseconds, since 1 command = 2 x RTC ECAT.
- The second time is the elaboration time T_e elapsed by TRM-LED to compose the pattern; Ex. for 1000 LEDs $T_e = 20\mu s \times 1000 = 20ms$.
- The third time is the transmission-time T_t ; Ex. for 1000 LEDs $T_t = 1,25\mu s \times 8 \times 1000 \times 3 = 30ms$.

For a single command sequence and an RTC=1ms it can estimate:

$$T_{tot} = T_c + (T_{e1} + T_{t1}) + (T_{e2} + T_{t2}) = 2ms + (20ms + 30ms) + (20ms + 30ms) = 102ms.$$



Yellow is the command duration. Pink and green rectangles are transmission data to the strips

Command data sent to the LED strip has been received by the TRM-LED device into the RxPDOs.

Each PDO carries 16 bytes. The datagram field has a format, here explained, where the meaning of the DATA1-12 bytes depends by the command coding:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
A3	A2	A1	A0	NUM				Byte 0
B3	B2	B1	B0	CFC/AFC				Byte 1
DATA 1								Byte 2
DATA 2								Byte 3
DATA 3								Byte 4
DATA 4								Byte 5
DATA 5								Byte 6
DATA 6								Byte 7
DATA 7								Byte 8
DATA 8								Byte 9
DATA 9								Byte 10
DATA 10								Byte 11
DATA 11								Byte 12
DATA 12								Byte 13
<u>Optional data</u>								Byte 14
CHECKSUM (optional)								Byte 15

For a Write Command (RxPDOs):

NUM = Current Command Number (incremental value from 1 to 15, roll-over 15→1; 0=reset)

CFC = Command Function Code (1-15, 0=no command)

A[3:0] = 0x5 (mandatory if command CHECKSUM required, else dummy)

B[3:0] = 0xA (mandatory if answer CHECKSUM required, else dummy)

DATA byte[2-13] = datagram, data field for the command

OPTIONAL DATA byte[14] = 0x00 in Tx, currently non used

CHECKSUM = sum of byte 1-15

For a Read Command (TxPDOs):

NUM = Current Command Number (last received value from 1 to 15, roll-over 15→1; 0=fault). This field returns the last Current Command Number when the command has been executed.

AFC = Answer Function Code (1-15, 0=no command). It repeats the last current command when the command work is in progress.

A[3:0] = 0xA (mandatory if NO ERRORS, else 0x00)

B[3:0] = 0x5 (mandatory if NO ERRORS, else 0x00)

DATA byte[2-13] = datagram, data field for the answer

OPTIONAL DATA byte[14] = 0x00 no Power Supply; 0x03=Power Supply CH2(bit1)+CH1(bit0)

CHECKSUM = sum of byte 1-15

5.5 Data Protocol over ECAT

The data protocol is based on the Master-Slave model: the slave doesn't autonomously send any data to the master, but always as answer to a command sent by the master.

Thus, a Command is a message in the direction Master-to-Slave and depends by the Master; an Answer is a message in the direction Slave-to-Master and it is the final act of a communication.

For each command there is always one answer.

Every ECAT time-cycle, a new pattern is sent to the slave. The pattern can be the last one same pattern. So, the command is continuously updated.

In correspondence, an answer pattern is generated and sent to the master by the TRM-LED. In every ECAT cycle, the answer is updated in accord to the command.

There is no synchronization or time-out between the master and a slave.

The master is the CNC executor on the ECAT master and it can communicate with more slaves (more TRM-LED modules).

When the master decides to communicate with a slave,

1. It updates data to transmit into a RxPDO format. Data can be already broadcast even if they not completely updated.
2. It updates the command CFC field.
3. Then, as last operation, it updates the NUM field, increasing the current NUM value (NUM++).
4. This last byte updating is an "atomic" operation which is detected by the slave as the trigger event for decoding the new command; only after changing the NUM field the slave can read the other fields and starts to execute the command.
5. Detecting the change of NUM, the slave starts to execute the new command, and it immediately changes the filed AFC in the answer. In this way the master can understand that the command is in progress.
6. When the command is completely executed, the slave returns data, firstly updating data in the answer frame and, as last action, it updates the filed NUM with the current NUM number (the same of the command).
7. NUM received = NUM transmitted means that the command has been successfully executed and closed.

The situation to accept a command is given by 3 conditions:

1. Tx Checksum is correct
2. A new valid command is broadcast
3. A new command number is provided

The situation to grant a command is working in progress is given by 2 conditions:

1. Rx Checksum is correct
2. The required command is broadcast back
3. The last executed command number is returned

The situation to grant a command has been executed is given by 3 conditions:

1. Rx Checksum is correct
2. The required command is broadcast back
3. The new command number is returned

Some other rules complete the protocol:

- The command is a message always made by 16 bytes.
- The last byte is a CheckSum = sum of data 1-15. If CheckSum is not correct, the command is dummy.
- The answer is a message always made by 16 bytes.
- The last byte is a CheckSum = sum of data 1-15. If CheckSum is not correct, the command is dummy.
- The CheckSum value is calculated as a simple sum without carry of the proceeding data 1-15.

6 FRAME FORMAT

6.1 Command frame format

Commands are made by a 16 byte frame.
The frame is so made.

6.1.1 NUM and A[3:0]

The field NUM=byte0[3:0] is called Command Number.
It is generated increasing the preceeding command number.

$$\text{New NUM} = \text{last NUM} + 1$$

This condition is the trigger event to begin the execution of the new command.
For this reason, all data and command field must be updated to the new command values before updating this field. After sending the command, the last NUM must be updated to this last one.
The command will be closed (executed) when in the answer NUM filed will appear the same value.

NUM range is 1-15, updating is NUM=NUM+1, roll-over 15-1
NUM=0 is reserved.

A[3:0] = byte0[7:4] is an optional field. When A[3:0]=0x5, the transmission CHECKSUM becomes mandatory and it must be updated.

6.1.2 CFC - Command Function Code and B[3:0]

The field CFC=byte1[3:0] is the Command Function Code. Every code can be a command, following the table below.

CFC's range is from 0x1 a 0xF, 0x0 is reserved as reset code.

To each command code, the same code is reserved as answer. So, for an instance, to command code CFC=1 the matching answer will be AFC=1 ad so on.

AFC=0 code is reserved for error cases, when the required command code is not supported by the device or when the CHECKSUM value is not correct.

CFC codes

0x0	RESERVED - RESET
0x1	START/STOP
0x2	WRITE OUTPUT - 1 COLOR FOR 3 SEGMENTS
0x3	SET STRIP (NEW!!!)
0x5	READ/WRITE STATUS
0xA	SHIFT
0xB	WRITE OUTPUT - 2 COLORS FOR 2 SEGMENTS
0xC	WRITE OUTPUT - 1 COLOR FOR 2 EXTENDED SEGMENTS
0xD	WRITE THE MAP OF COLORS
0xE	WRITE OUTPUT – 3 COLORS OF THE MAP FOR 3 SEGMENTS

6.1.3 CHECKSUM

Its value is the sum without carry of the numeric value of Byte0 to Byte15 of the frame. The receiver verifies the value of the CHECKSUM, calculating the sum with the just received bytes. If the values match, then the message is valid and the command will be executed. If the values don't match, an AFC=0 (not-acknowledge) answer code is returned.

Command format

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	1	NUM				Byte 0
1	0	1	<u>0</u>	CFC				Byte 1
DATA 1								Byte 2
DATA 2								Byte 3
DATA 3								Byte 4
DATA 4								Byte 5
DATA 5								Byte 6
DATA 6								Byte 7
DATA 7								Byte 8
DATA 8								Byte 9
DATA 9								Byte 10
DATA 10								Byte 11
DATA 11								Byte 12
DATA 12								Byte 13
0x00								Byte 14
CHECKSUM								Byte 15

6.2 Answer frame format

Answers are made by a 16 byte frame.
The frame is so made.

6.2.1 NUM and A[3:0]

The field NUM=byte0[3:0] is called Command Number.

It is updated at the end of the command execution, to close the communication session. This happens by returning the same NUM value received with the new command; this action must be done only after having updated the DATA and AFC fields with the correct answer. Before this moment, NUM will return the last NUM number of command (executed).

NUM = last NUM current
NUM = new NUM to close
(then, last NUM = new NUM)

The NUM value range is 1 to 15, NUM=0 is reserved. After 15, it rolls over to 1, according to the received new NUM. The increment of NUM value is featured by the master.

A[3:0] = byte0[7:4] is a no-meaning field and it is tied to 0xA for compatibility.

6.1.2 AFC – Answer Function Code and B[3:0]

The field AFC=byte1[3:0] is the Answer Function Code. It matches the required Command Function Code. When the trigger event of a new command is happened (new NUM), the CFC value is mirrored into the AFC and soon the command is set in progress.

To each command code CFC, the same code is reserved as answer. So, for an instance, to command code CFC=1 the matching answer will be AFC=1 and so on.

AFC=0 code is reserved for error cases, when the required command code is not supported by the device or when the CHECKSUM value is not correct.

AFC range matches with the CFC range.

CFC codes

0x0	RESERVED - RESET
0x1	START/STOP
0x2	WRITE OUTPUT - 1 COLOR FOR 3 SEGMENTS
0x3	SE STRIP (NEW!!!)
0x5	READ/WRITE STATUS
0xA	SHIFT
0xB	WRITE OUTPUT - 2 COLORS FOR 2 SEGMENTS
0xC	WRITE OUTPUT - 1 COLOR FOR 2 EXTENDED SEGMENTS
0xD	WRITE THE MAP OF COLORS
0xE	WRITE OUTPUT – 3 COLORS OF THE MAP FOR 3 SEGMENTS

6.2.3 CHECKSUM

Its value is the sum without carry of the numeric value of Byte0 to Byte15 of the frame. If the received field B[3:0]=0xA, the receiver calculates the value of the CHECKSUM of Byte0 to Byte15 of its data and update Byte16. For any other value of B[3:0], Byte16 is left 0x00.

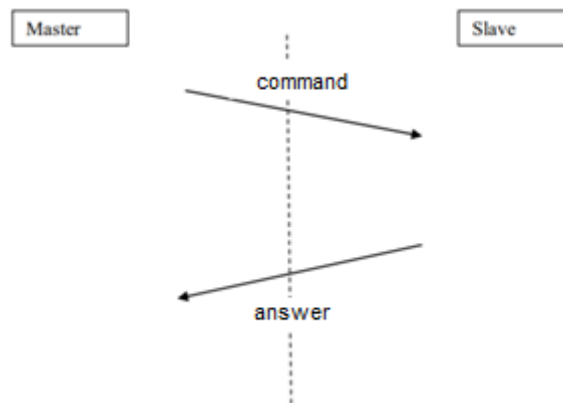
Answer format

lower format

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	0	NUM				Byte 0
0	1	0	1	AFC				Byte 1
DATA 1								Byte 2
DATA 2								Byte 3
DATA 3								Byte 4
DATA 4								Byte 5
DATA 5								Byte 6
DATA 6								Byte 7
DATA 7								Byte 8
DATA 8								Byte 9
DATA 9								Byte 10
DATA 10								Byte 11
DATA 11								Byte 12
DATA 12								Byte 13
0x00								Byte 14
CHECKSUM								Byte 15

7 HANDSHAKE

A communication cycle always starts with a message (command) sent by the master to the slave receiver end it always ends with a message (answer) from the slave receiver to the master.



7.1 NOT-ACKNOWLEDGE (AFC = 0x0)

This is the answer NOT-ACKNOWLEDGE, for every received complete command, but:

- Not supported by the TRM-LED
- With a wrong checksum (communication error)
- Answer to 0xA0 command is NACK

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	0	NUM				Byte 0
0	1	0	1	AFC=0				Byte 1
X	X	X	X	X	X	X	X	Byte 2
X	X	X	X	X	X	X	X	Byte 3
X	X	X	X	X	X	X	X	Byte 4
X	X	X	X	X	X	X	X	Byte 5
X	X	X	X	X	X	X	X	Byte 6
X	X	X	X	X	X	X	X	Byte 7
X	X	X	X	X	X	X	X	Byte 8
X	X	X	X	X	X	X	X	Byte 9
X	X	X	X	X	X	X	X	Byte 10
X	X	X	X	X	X	X	X	Byte 11
X	X	X	X	X	X	X	X	Byte 12
X	X	X	X	X	X	X	X	Byte 13
0x00								Byte 14
CHECKSUM								Byte 15

AFC = 0x0

NUM = new NUM

8 COMMANDS

General definitions

Channel 1 – Channel 2: these terms are concerned with the two physical output to drive LED strips on board of TRM-LED. Channel 1 can map 1000 LEDS (0-999), Channel 2 can map 1000 LEDS (1000-1999)

Segment: this term is used to mean a set of contiguous LEDS along the same Channel. It is identified by the progressive number of the first LED (LED numeration starts from 0). A segment must be included in range 0-999 (Channel 1) or 1000-1999 (Channel 2).

Color – RGB: each LED accepts three bytes to define its chromatic behaviour, following the RGB coding. The first byte is for RED, the second is for GREEN, the third is for BLUE. Every color can be modulated into a range 0-255. Intensity of the color is fixed, max 20%, limited by the device.

8.1 START-STOP (CFC=AFC=0x1)

This is the command to switch on/off the communication protocol of the device.

To start any type of communication of the master you need to send a START/STOP command with the sub-command = 0x01 (START function) to the device.

If on the ECAT net there are more TRM-LEDs, they must receive their initialization command START/STOP with the START function each one.

In the same way, the command START/STOP with function STOP inhibits the communication with the device, and the device doesn't answer to any command (default condition of the device). The positive answer (no Not Acknowledge), AFC=CFC and NUM=new NUM, means the device is ready to communicate and execute commands (START function).

Command:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	1	NUM				Byte 0
1	0	1	0	0	0	0	1	Byte 1
0	0	0	0	0	0	0	0	Byte 2
0	0	0	0	0	0	0	0	Byte 3
0	0	0	0	0	0	0	0	Byte 4
0	0	0	0	0	0	0	0	Byte 5
0	0	0	0	0	0	0	0	Byte 6
X	X	X	X	X	X	X	X	Byte 7
X	X	X	X	X	X	X	X	Byte 8
X	X	X	X	X	X	X	X	Byte 9
X	X	X	X	X	X	X	X	Byte 10
X	X	X	X	X	X	X	X	Byte 11
X	X	X	X	X	X	X	X	Byte 12
X	X	X	X	X	X	X	X	Byte 13
0x00								Byte 14
CHECKSUM								Byte 15

Answer:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	0	NUM				Byte 0
0	1	0	1	0	0	0	1	Byte 1
X	X	X	X	X	X	X	X	Byte 2
X	X	X	X	X	X	X	X	Byte 3
X	X	X	X	X	X	X	X	Byte 4
X	X	X	X	X	X	X	X	Byte 5
X	X	X	X	X	X	X	X	Byte 6
X	X	X	X	X	X	X	X	Byte 7
X	X	X	X	X	X	X	X	Byte 8
X	X	X	X	X	X	X	X	Byte 9
X	X	X	X	X	X	X	X	Byte 10
REV				VER				Byte 11
X	X	X	X	X	X	X	X	Byte 12
X	X	X	X	X	X	X	X	Byte 13
ALIM								Byte 14
CHECKSUM								Byte 15

Ver = version number**Rev** = revision number

X = may be the same command data, mirrored back

ALIM = Power Supply CH1+CH2

8.2 WRITE OUTPUT - 1 COLOR FOR 3 SEGMENTS (CFC=AFC=0x2)

This command features to modify the state of 3 segments of LED with max length 255 LED. The state of color of the three segments is the same for all.

The device can manage LED of both the output channels on which is possible to drive 1000 LED max each one. To address LEDS of channel A and B, use addresses 0-999 for the first channel and addresses 1000-1999 for the second one.

Command:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	1	NUM				Byte 0
1	0	1	0	0	0	1	0	Byte 1
Address leds segment 1 LSB								Byte 2
Address leds segment 1 MSB								Byte 3
Number of leds segment 1								Byte 4
Address leds segment 2 LSB								Byte 5
Address leds segment 2 MSB								Byte 6
Number of leds segmeno 2								Byte 7
Address leds segment 3 LSB								Byte 8
Address leds segment 3 MSB								Byte 9
Number of leds segment 3								Byte 10
Color R								Byte 11
Color G								Byte 12
Color B								Byte 13
0x00								Byte 14
CHECKSUM								Byte 15

Address leds segment N = Position of the first led of the segmento N (LEDs numeration starts from 0).

Number of leds segment N = Lenght of the segment in number of LEDs (if 0 no action on the LEDs).

Color X = Color of the segment N.

Ex. Tx command:

0x50 0xA2 0x00 0x00 0x0A 0x14 0x00 0x0A 0x28 0x00 0x0A 0x00 0xFF 0x00 0x00 0x4B
will light on in green leds 0-9 (segment1), 20-29 (segment2), 40-49 (segment3) of channel 1.

Answer:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	0	NUM				Byte 0
0	1	0	1	0	0	1	0	Byte 1
X	X	X	X	X	X	X	X	Byte 2
X	X	X	X	X	X	X	X	Byte 3
X	X	X	X	X	X	X	X	Byte 4
X	X	X	X	X	X	X	X	Byte 5
X	X	X	X	X	X	X	X	Byte 6
X	X	X	X	X	X	X	X	Byte 7
X	X	X	X	X	X	X	X	Byte 8
X	X	X	X	X	X	X	X	Byte 9
X	X	X	X	X	X	X	X	Byte 10
X	X	X	X	X	X	X	X	Byte 11
X	X	X	X	X	X	X	X	Byte 12
X	X	X	X	X	X	X	X	Byte 13
ALIM								Byte 14
CHECKSUM								Byte 15

X = may be the same command data, mirrored back

ALIM = Power Supply CH1+CH2

8.3 SET STRIP – SET THE STRIP WITH A COLOR (CFC=AFC=0x3)

This command features to modify the state of all LEDs of the strip: two channels, two strips. The state of color of the the single strip is the same for all LEDs.

The device can manage two different colors for the two strips / output channels.

Then, it sets the number of LEDs for each strip. The command is useful to set-up the two strips and to clear them (set R-G-B = 0).

Command:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	1	NUM				Byte 0
1	0	1	0	0	0	1	1	Byte 1
Length strip 1 LSB								Byte 2
Length strip 1 MSB								Byte 3
Length strip 2 LSB								Byte 4
Length strip 2 MSB								Byte 5
Color R strip 1								Byte 6
Color G strip 1								Byte 7
Color B strip 1								Byte 8
Color R strip 2								Byte 9
Color G strip 2								Byte 10
Color B strip 2								Byte 11
								Byte 12
								Byte 13
0x00								Byte 14
CHECKSUM								Byte 15

Ex. Tx command:

0x51 0xA3 0xE8 0x03 0xE8 0x03 0xFF 0x00 0x00 0x00 0x00 0xFF 0x00 0x00 0x00 0xC8
will light on in RED leds 0-999 (strip1), in BLUE 1000-1999 (strip2).

Answer:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	0	NUM				Byte 0
0	1	0	1	0	0	1	0	Byte 1
X	X	X	X	X	X	X	X	Byte 2
X	X	X	X	X	X	X	X	Byte 3
X	X	X	X	X	X	X	X	Byte 4
X	X	X	X	X	X	X	X	Byte 5
X	X	X	X	X	X	X	X	Byte 6
X	X	X	X	X	X	X	X	Byte 7
X	X	X	X	X	X	X	X	Byte 8
X	X	X	X	X	X	X	X	Byte 9
X	X	X	X	X	X	X	X	Byte 10
X	X	X	X	X	X	X	X	Byte 11
X	X	X	X	X	X	X	X	Byte 12
X	X	X	X	X	X	X	X	Byte 13
ALIM								Byte 14
CHECKSUM								Byte 15

X = may be the same command data, mirrored back

ALIM = Power Supply CH1+CH2

8.4 READ/WRITE STATUS COMMAND (CFC=AFC=0x5)

By this command it is possible to config the running condition of the device, how many LEDs it must drive.

This command features:

- select to use differential or single ended serial lines towards the LED stripes. Single ended means the stripe can be connected directly with the device for a short track; differential lines means TRM-LED is connected to the device “LED-EXTENDER” for extending the track towards the LED stripe.
- set the number of LED for every channel; in the WS2821B protocol this means to set how many periods (bits) the outputs must generate to drive the LED strip.

Command:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	1	ADDR				Byte 0
1	0	1	0	0	1	0	1	Byte 1
X	X	X	X	X	X	X	X	Byte 2
0	0	0	0	0	0	0	0	Byte 3
0	0	0	0	0	0	MODE_2	MODE_1	Byte 4
LED NUMBER FOR CHANNEL 1 LSB								Byte 5
LED NUMBER FOR CHANNEL 1 MSB								Byte 6
LED NUMBER FOR CHANNEL 2 LSB								Byte 7
LED NUMBER FOR CHANNEL 2 MSB								Byte 8
X	X	X	X	X	X	X	X	Byte 9
X	X	X	X	X	X	X	X	Byte 10
X	X	X	X	X	X	X	X	Byte 11
X	X	X	X	X	X	X	X	Byte 12
X	X	X	X	X	X	X	X	Byte 13
0x00								Byte 14
CHECKSUM								Byte 15

MODE_1 = Channel 1 single-ended=0 / differential=1.

MODE_2 = Channel 2 single-ended=0 / differential=1.

LED NUMBER FOR CHANNEL 1 = Number of LEDs connected to Channel 1.

LED NUMBER FOR CHANNEL 2 = Number of LEDs connected to Channel 2.

Answer:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	0	ADDR				Byte 0
0	1	0	1	0	1	0	1	Byte 1
X	X	X	X	X	X	X	X	Byte 2
X	X	X	X	X	X	X	X	Byte 3
X	X	X	X	X	X	MODE_2	MODE_1	Byte 4
LED NUMBER FOR CHANNEL 1 LSB								Byte 5
LED NUMBER FOR CHANNEL 1 MSB								Byte 6
LED NUMBER FOR CHANNEL 2 LSB								Byte 7
LED NUMBER FOR CHANNEL 2 MSB								Byte 8
X	X	X	X	X	X	X	X	Byte 9
X	X	X	X	X	X	X	X	Byte 10
X	X	X	X	X	X	X	X	Byte 11
X	X	X	X	X	X	X	X	Byte 12
X	X	X	X	X	X	X	X	Byte 13
ALIM								Byte 14
CHECKSUM								Byte 15

MODE_1 = Channel 1 single-ended=0 / differential=1.

MODE_2 = Channel 2 single-ended=0 / differential=1.

LED NUMBER FOR CHANNEL 1 = Number of LEDs connected to Channel 1.

LED NUMBER FOR CHANNEL 2 = Number of LEDs connected to Channel 2.

ALIM = Power Supply CH1+CH2

8.5 SHIFT FORWARD/BACKWARD (CFC=AFC=0xA)

This command will modify the state of max 2 segment of LEDs, scrolling forward or backward the lighted current pattern (limited by a start edge and a final edge) of a LED-steps value. The scroll direction is driven by a direction bit.

When the scrolling operation reaches the edge of the strip, the behaviour can be set by the carry bit RN.

If carry is not enabled, the first LED of the strip maintains the current state; if carry is enabled, the state condition of the last LED of the strip is moved to the first one. It has the same behaviour when the shift direction is negative.

The device can drive 2 Channels of max 1000 LED each one. Addresses between 0 and 999 regard the 1000 LEDs on Channel 1. Addresses between 1000 and 1999 regard the 1000 LEDs on Channel 2.

Command:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	1	NUM				Byte 0
1	0	1	0	1	0	1	0	Byte 1
0	0	0	0	0	0	RN1	DN1	Byte 2
Segment 1 initial edge LSB								Byte 3
Segment 1 initial edge MSB								Byte 4
Segment 1 final edge LSB								Byte 5
Segment 1 final edge MSB								Byte 6
Segment 1 positions to shift								Byte 7
0	0	0	0	0	0	RN2	DN2	Byte 8
Segment 2 initial edge LSB								Byte 9
Segment 2 initial edge MSB								Byte 10
Segment 2 final edge LSB								Byte 11
Segment 2 final edge MSB								Byte 12
Segment 2 positions to shift								Byte 13
0x00								Byte 14
CHECKSUM								Byte 15

Segment initial edge 1,2= first led belonged to segment 1,2 (starting from 0).

Segment final edge 1,2 = last led belonged to segment 1,2 (starting from 0).

Segment 1,2 positions to shift = how many LED-steps to scroll in forward o backward direction.

DN1,2 = Direction towards segment 1/2 will be shifted.

0 per shiftare avanti.

1 per shiftare indietro.

RN1,2 = Enable of the shift carry.

0 carry not enabled

1 carry enabled (after the last led is sets, the first one lights on and viceversa)

Answer:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	0	NUM				Byte 0
0	1	0	1	1	0	1	0	Byte 1
X	X	X	X	X	X	X	X	Byte 2
X	X	X	X	X	X	X	X	Byte 3
X	X	X	X	X	X	X	X	Byte 4
X	X	X	X	X	X	X	X	Byte 5
X	X	X	X	X	X	X	X	Byte 6
X	X	X	X	X	X	X	X	Byte 7
X	X	X	X	X	X	X	X	Byte 8
X	X	X	X	X	X	X	X	Byte 9
X	X	X	X	X	X	X	X	Byte 10
X	X	X	X	X	X	X	X	Byte 11
X	X	X	X	X	X	X	X	Byte 12
X	X	X	X	X	X	X	X	Byte 13
ALIM								Byte 14
CHECKSUM								Byte 15

X = may be the same command data, mirrored back

ALIM = Power Supply CH1+CH2

8.6 WRITE OUTPUT - 2 COLORS FOR 2 SEGMENTS (CFC=AFC=0xB)

This command features to modify in the same time the status of 2 segments of max 255 of different colors.

To address the first channel of max 1000 LEDS let's use address 0-999.

To address the second channel of max 1000 LEDS let's use address 1000-1999.

Command:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	1	NUM				Byte 0
1	0	1	0	1	0	1	1	Byte 1
Initial LED segment 1 LSB								Byte 2
Initial LED segment 1 MSB								Byte 3
Number of LEDs segment 1								Byte 4
Color R segment 1								Byte 5
Color G segment 1								Byte 6
Color B segment 1								Byte 7
Initial LED segment 2 LSB								Byte 8
Initial LED segment 2 MSB								Byte 9
Number of LEDs segment 2								Byte 10
Color R segment 2								Byte 11
Color G segment 2								Byte 12
Color B segment 2								Byte 13
0x00								Byte 14
CHECKSUM								Byte 15

Initial LED segmento 1,2 = Initial edge of the first LED of segment 1,2 (starting from 0).

Number of LEDs segmento 1,2 = length of the segment 1,2 (if 0, no modification to the segment).

Colore X segmento 1,2 = R,G,B colors of the segment 1,2.

Answer:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	0	NUM				Byte 0
0	1	0	1	1	0	1	1	Byte 1
X	X	X	X	X	X	X	X	Byte 2
X	X	X	X	X	X	X	X	Byte 3
X	X	X	X	X	X	X	X	Byte 4
X	X	X	X	X	X	X	X	Byte 5
X	X	X	X	X	X	X	X	Byte 6
X	X	X	X	X	X	X	X	Byte 7
X	X	X	X	X	X	X	X	Byte 8
X	X	X	X	X	X	X	X	Byte 9
X	X	X	X	X	X	X	X	Byte 10
X	X	X	X	X	X	X	X	Byte 11
X	X	X	X	X	X	X	X	Byte 12
X	X	X	X	X	X	X	X	Byte 13
ALIM								Byte 14
CHECKSUM								Byte 15

X = may be the same command data, mirrored back

ALIM = Power Supply CH1+CH2

8.7 WRITE OUTPUT - 1 COLOR FOR 2 EXTENDED SEGMENTS (CFC=AFC=0xC)

This command features to modify in the same time the status of 2 segments of max 1000 LEDs of the same color (same R-G-B code).

To address the first channel of max 1000 LEDS let's use address 0-999.

To address the second channel of max 1000 LEDS let's use address 1000-1999.

Command:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	1	NUM				Byte 0
1	0	1	0	1	1	0	0	Byte 1
Initial LED segment 1 LSB								Byte 2
Initial LED segment 1 MSB								Byte 3
Number of LEDs for segment 1 LSB								Byte 4
Number of LEDs for segment 1 MSB								Byte 5
Initial LED segment 2 LSB								Byte 6
Initial LED segment 2 MSB								Byte 7
Number of LEDs for segment 2 LSB								Byte 8
Number of LEDs for segment 2 MSB								Byte 9
Color R								Byte 10
Color G								Byte 11
Color B								Byte 12
0x00								Byte 13
0x00								Byte 14
CHECKSUM								Byte 15

Initial LED segment 1,2 = first LED edge position for segment 1,2 (starting from 0).

Number of LEDs for segment 1,2 = length of the segment (if 0, no modification).

Color X = Color of the segment.

Answer:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	0	NUM				Byte 0
0	1	0	1	1	1	0	0	Byte 1
X	X	X	X	X	X	X	X	Byte 2
X	X	X	X	X	X	X	X	Byte 3
X	X	X	X	X	X	X	X	Byte 4
X	X	X	X	X	X	X	X	Byte 5
X	X	X	X	X	X	X	X	Byte 6
X	X	X	X	X	X	X	X	Byte 7
X	X	X	X	X	X	X	X	Byte 8
X	X	X	X	X	X	X	X	Byte 9
X	X	X	X	X	X	X	X	Byte 10
X	X	X	X	X	X	X	X	Byte 11
X	X	X	X	X	X	X	X	Byte 12
X	X	X	X	X	X	X	X	Byte 13
ALIM								Byte 14
CHECKSUM								Byte 15

X = may be the same command data, mirrored back

ALIM = Power Supply CH1+CH2

8.8 COLOR MAP WRITING (CFC=AFC=0xD)

This command is made to write into the “map of colors”, a sort of palette to store same colors as RGB codes. The “map of colors” contains 50 elements, with an index from #1 to #50. This command can save 3 elements into the map. Set Map Address = 0 if you don't need to save all 3 the colors.

Command:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	1	NUM				Byte 0
1	0	1	0	1	1	0	1	Byte 1
Map Address 1								Byte 2
Color R 1								Byte 3
Color G 1								Byte 4
Color B 1								Byte 5
Map Address 2								Byte 6
Color R 2								Byte 7
Color G 2								Byte 8
Color B 2								Byte 9
Map Address 3								Byte 10
Color R 3								Byte 11
Color G 3								Byte 12
Color B 3								Byte 13
0x00								Byte 14
CHECKSUM								Byte 15

Map Address = address of the new color(s) – range (1-50).

Colore X = R-G-B code to identify the new color.

Answer:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	0	NUM				Byte 0
0	1	0	1	1	1	0	1	Byte 1
X	X	X	X	X	X	X	X	Byte 2
X	X	X	X	X	X	X	X	Byte 3
X	X	X	X	X	X	X	X	Byte 4
X	X	X	X	X	X	X	X	Byte 5
X	X	X	X	X	X	X	X	Byte 6
X	X	X	X	X	X	X	X	Byte 7
X	X	X	X	X	X	X	X	Byte 8
X	X	X	X	X	X	X	X	Byte 9
X	X	X	X	X	X	X	X	Byte 10
X	X	X	X	X	X	X	X	Byte 11
X	X	X	X	X	X	X	X	Byte 12
X	X	X	X	X	X	X	X	Byte 13
ALIM								Byte 14
CHECKSUM								Byte 15

X = may be the same command data, mirrored back

ALIM = Power Supply CH1+CH2

8.9 WRITE OUTPUT – 3 SEGMENTS WITH 3 COLORS BY MAP (CFC=AFC=0xE)

This command features the possibility to modify 3 segments with max length 255 LEDs, assigning for each segment a color stored in the map of colors, calling it by the index (1-50).

To address the first channel of max 1000 LEDS let's use address 0-999.

To address the second channel of max 1000 LEDS let's use address 1000-1999.

Command:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	1	NUM				Byte 0
1	0	1	0	1	1	1	0	Byte 1
Initial LED segment 1 LSB								Byte 2
Initial LED segment 1 MSB								Byte 3
Number of LEDs for segment 1								Byte 4
Color by Map Address (index) for segment 1								Byte 5
Initial LED segment 2 LSB								Byte 6
Initial LED segment 2 MSB								Byte 7
Number of LEDs for segment 2								Byte 8
Color by Map Address (index) for segment 2								Byte 9
Initial LED segment 3 LSB								Byte 10
Initial LED segment 3 MSB								Byte 11
Number of LEDs for segment 3								Byte 12
Color by Map Address (index) for segment 3								Byte 13
0x00								Byte 14
CHECKSUM								Byte 15

Initial LED segment 1,2,3 = first LED edge of the segment 1,2,3 (starting from 0).

Number of LEDs of segment 1,2,3 = length of the segment 1,2,3 (if 0, no modification).

Color Map Address for segment 1,2,3 = Map Address (index 1-50) for the color chosen for segment 1,2,3. If index=0, the segment is switched off (no light).

Answer:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	1	0	NUM				Byte 0
0	1	0	1	1	1	1	0	Byte 1
X	X	X	X	X	X	X	X	Byte 2
X	X	X	X	X	X	X	X	Byte 3
X	X	X	X	X	X	X	X	Byte 4
X	X	X	X	X	X	X	X	Byte 5
X	X	X	X	X	X	X	X	Byte 6
X	X	X	X	X	X	X	X	Byte 7
X	X	X	X	X	X	X	X	Byte 8
X	X	X	X	X	X	X	X	Byte 9
X	X	X	X	X	X	X	X	Byte 10
X	X	X	X	X	X	X	X	Byte 11
X	X	X	X	X	X	X	X	Byte 12
X	X	X	X	X	X	X	X	Byte 13
ALIM								Byte 14
CHECKSUM								Byte 15

X = may be the same command data, mirrored back

ALIM = Power Supply CH1+CH2

8.10 ALIM – check +5V Power Supply

For every command, the matching answer provides the information about the presence of Power Supply for the LEDs' output.

The +5V CH1 and +5V CH2 power supplies must be provided by external sources. These power supplies give power to the serial drivers for the WS2812B lanes. It is also possible to provide the current for the LED stripes, with the limit of 5A which is the limit for the contacts of the connector.

This means that +5V power supply:

- It can be connected to TRM-LED and derived from it towards the LED stripe (max 5A)
- It can be connected directly to the LED stripe and in parallel also tied to TRM-LED

There are two separate Power Supplies for the 2 channels: +5V CH1, +5V CH2 (0V is shared).

ALIM

7	6	5	4	3	2	1	0	ALIM
0	0	0	0	0	0	CH2	CH1	Byte 13

CH1 = +5V CH1 (1 on/ 0 off)

CH2 = +5V CH2 (1 on/ 0 off)

9 TRM-LED VIEW



TRM-LED
Front view



TRM-LED
Right side view

T.P.A. Srl Tecnologie e Prodotti per l'Automazione - Via Carducci, 221 - 20099 Sesto S.
Giovanni

Tel. +390236527550 - www.tpaspa.it - P.I.: IT02016240968



Tecnologie e Prodotti per l'Automazione
via Carducci 221
20099 Sesto san Giovanni (MI)
Tel. +39 02 36527550

www.tpaspa.it