

LUMIPLUS

MODBUS MANUAL V2

© Fluidra, S.A. 2013. All Rights Reserved.

All the trademarks are registered by Fluidra SA and/or its affiliates, or its respective owners. Fluidra SA and its licensors will own all right, title and interest to the manual, technology and information including all portions, copies or modifications thereof.

Every effort has been made to ensure that the information given is correct. However, due to continuous product improvement, Fluidra reserves the right to make changes to products and technical data without prior notice.

1.	INTRODUCTION TO MODBUS AND PRODUCT	4
1.1.	Principle of operation	4
1.2.	Basic Characteristics	4
2.	ELECTRICAL CONNECTIONS	5
3.	CABLE CHARACTERISTICS	5
4.	BUS ISOLATION AND TERMINATION RESISTORS	6
5.	PANEL INDICATOR	7
6.	MODBUS FUNCTIONS	7
6.1.	Functions supported	7
6.2.	Exception responses	8
7.	DEVICE DESCRIPTION AND CONFIGURATION	8
7.1.	General Description	8
7.2.	State Machine Diagram	9
7.3.	Address and baud rate selection	10
7.3.1.	Address setting	10
7.3.2.	Baud rate selection	10
7.4.	Broadcasting	11
7.5.	Watchdog	11
7.5.1.	Watchdog Triggering	11
7.5.2.	Watchdog SETUP	12
7.5.3.	Watchdog predefined Color/Sequence	13
8.	OPERATION MODES	14
8.1.	Basic mode	14
8.1.1.	Start Predefined Color/Sequence	14
8.1.2.	Checking current state	15
8.1.3.	Actuator Change Request	16
8.1.4.	Request ON	16
8.1.5.	Request OFF	17
8.1.6.	Changing Color	17
8.1.7.	Changing Sequence	19
8.1.8.	Changing Speed	20
8.1.9.	Sleep Time	21
8.1.10.	Checking LATCHED Alarms Register	21
8.1.11.	Checking instantaneous error Register	22
9.	BASIC MODBUS-RTU REGISTER MAP	23
10.	TROUBLESHOOTING GUIDE	28
11.	PRODUCT REVISION	28

1. INTRODUCTION TO MODBUS AND PRODUCT

Thank you very much for purchasing our Lumiplus with MODBUS-RTU features. This manual is intended for professional installer, if you are not, please consult to your official distributor.

MODBUS is an open field bus successfully used through the world to connect field devices to a main controller. This is the reason why MODBUS has been our choice to offer to our customers and partners an automated solution easy to integrate not only with our brand products but also with a vast collection of third party components and controllers

MODBUS, MODBUS-RTU and other related names are registered trademarks of MODBUS Organization. Further information and documentation can be found at <http://www.Modbus.org/>

1.1. PRINCIPLE OF OPERATION

The Lumiplus implements MODBUS-RTU as a control-communications feature that allows its operation and supervision tasks from a MODBUS automation environment. Preventive maintenance and fault analysis is also possible thanks to the implementation of internal registers in the Lumiplus with the more relevant operational and error events.

Whenever the Lumiplus is installed, you are not forced to connect it to a MODBUS system, as far as you do not aim to control or supervise it externally. The Lumiplus can run in local mode, as traditionally done, without using the MODBUS layer.

However, we expect that the implementation of MODBUS-RTU in the Lumiplus will open to our advanced customers and partners a wide range of new opportunities and implementation scenarios thanks to the simplicity and flexibility of the MODBUS-RTU layer.

Using a MODBUS-RTU message, the Lumiplus can change to a specific color, sequence or velocity, report errors, historical data and so on, giving to the user/installer a wide range of new features based in the automation of an already existing and proved Lumiplus.

1.2. BASIC CHARACTERISTICS

The MODBUS communication system provides a Master/Slave implementation among devices sharing a physical connection. For the Lumiplus, the physical connection is a RS485 half duplex serial layer, which has been chosen among other options due to its wide implementation and roughness.

For the Lumiplus, a RS-485 half duplex wired connection has been implemented and the projector is designed to run in a single-master system. In this implementation, Master and Slave figures has a clear role that is crucial to clear understand for a proper system implementation.

Master Device: Device that controls the data exchange in the bus and, if necessary, implements co-ordination tasks among different slaves (i.e. PLC Programmable Logic Controller, SCADA, etc).

Slave Device: Devices connected to the bus that attends to the requests from the master, either reporting information or executing tasks as per Master request.

2. ELECTRICAL CONNECTIONS

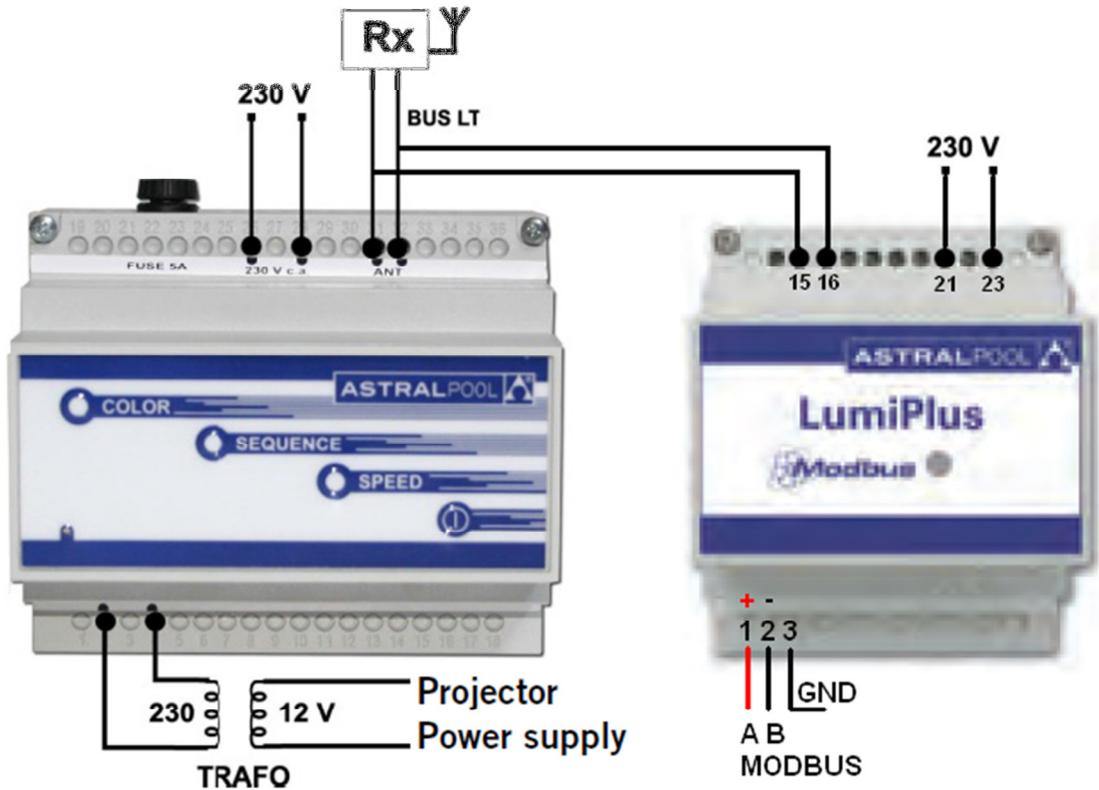


Image 1 Electrical Connection

Note: some manufacturers assign for the RS-485 port the “A” connection as a “+”, and “B” as a “-”, while others reverse this nomenclature. The Lumiplus uses the “A” as “+”, and the “B” as “-”. Mind this aspect when connecting to the bus devices coming from different manufacturers.

3. CABLE CHARACTERISTICS

The recommended wiring for a MODBUS-RTU Communication is based in a linear structure, active bus with termination at both ends. Is possible coupling and uncoupling of devices during operation without affecting other devices. The wire shall be twisted and shielded according to EN 50 170.

The values of transmission rate supported for the device, allow maximum cable length of 1,200 m without repeaters, or up to 10 km using repeaters, when installation is according to the standard.

For the balanced pairs used in an RS485-system, a Characteristic Impedance with a value higher than 100 Ohms may be preferred, especially for 19200 and higher baud rates.

4. BUS ISOLATION AND TERMINATION RESISTORS

If the communication bus is accessible for the user, it shall be double insulated. As far as in general the accessibility of the bus to users will depend on each single installation, safety isolation has been implemented in the Lumiplus physical bus layer. Moreover, for safety purposes, it is recommended to ensure that other devices sharing this bus also implements this insulation.

Additionally, the use of bus insulated devices not only enhances the security level, furthermore increases the equipment reliability, larger immunity to electromagnetic interference, longer life, higher reliability, more stability over the range of temperatures.

Whenever single or multiple devices are connected sharing a bus physical connection, it is recommended to use terminating resistors at the ends of the bus, even more when use large cable length or high speed data rates. The terminating resistor is used to prevent an RF signal from being reflected back from the end, causing interference. The terminating resistor must be in both ends of the bus, connected in parallel (as shown in the image below). A typical value of this resistance is 120Ω , $0.5W$. The value of the resistor must be the same in both ends. The terminating resistors are the resistors R_T of the image below

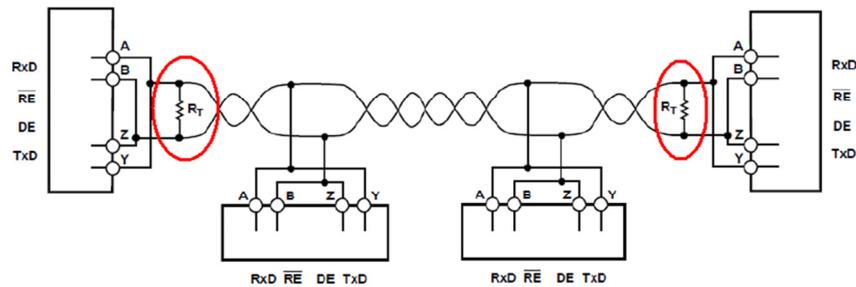


Image 2 Terminating Resistors

5. PANEL INDICATOR

The Lumiplus module has a LED that is ON when the module is powered. When data is transmitted through Modbus, this LED blinks.



Image 3 Lumiplus Panel Indicator

6. MODBUS FUNCTIONS

6.1. FUNCTIONS SUPPORTED

Please, be careful at the possible actuations, and make sure that the function used is the correct.

BIT ACCESS MODE

Functions in bit access mode are implemented according to the MODBUS-RTU standard described in http://www.Modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf

0x01 READ COILS

0x0F WRITE MULTIPLE COILS

0x05 WRITE SINGLE COIL. *This function is not implemented in the Lumiplus. Since they can be work rounded with the functions "Write Multiple Coils", to the special case where the number of coils to be written is one.*

0x02 READ DISCRETE INPUTS.

REGISTER ACCESS MODE

Functions in register access mode are implemented according to the MODBUS-RTU standard described in http://www.Modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf. In general registers are unsigned 16 bit coded.

0x03 READ HOLDING REGISTERS

0x04 READ INPUT REGISTERS

0x10 WRITE MULTIPLE REGISTERS

6.2. EXCEPTION RESPONSES

Exception responses are implemented according to the MODBUS-RTU standard described in the chapter MODBUS exception responses:

http://www.Modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf

The exceptions implemented are from 1 to 3.

7. DEVICE DESCRIPTION AND CONFIGURATION

7.1. GENERAL DESCRIPTION

If a request is made through Modbus and later another one with the remote control, the Modbus will indicate the color set through Modbus, it is because the unidirectionality between Modbus module and the Lumiplus modulator controlled through remote control.

When a simultaneous request is made through Modbus and remote control, and there is no conflict, the last request will take effect.

In general, there is not check on the constancy of the values sent to specific registers. Therefore is the operator responsibility to check that consistency.

In this manual, the numbers in hexadecimal have been represented with the format **0xZZ**, where ZZ is the number.

The register map that governs the 7.2 State Machine Diagram and all that is explained below is in the chapter 9 Basic Modbus-RTU register map.

7.2. STATE MACHINE DIAGRAM

When the system Powers ON, the state is always Start. The Start mode does not indicate a Light ON, it only indicates the alimentation of the Lumiplus, and NO request has been made after alimentation.

It is possible to change the Start Predefined Color, Sequence and Speed, also is possible to configure the Start mode with all the Lights OFF. From Start is possible to go to Request Color State through requesting an Update Predefined Color, is possible to go to Request Sequence & Speed State through requesting an Update Sequence & Speed or go to the Watchdog State if it triggers.

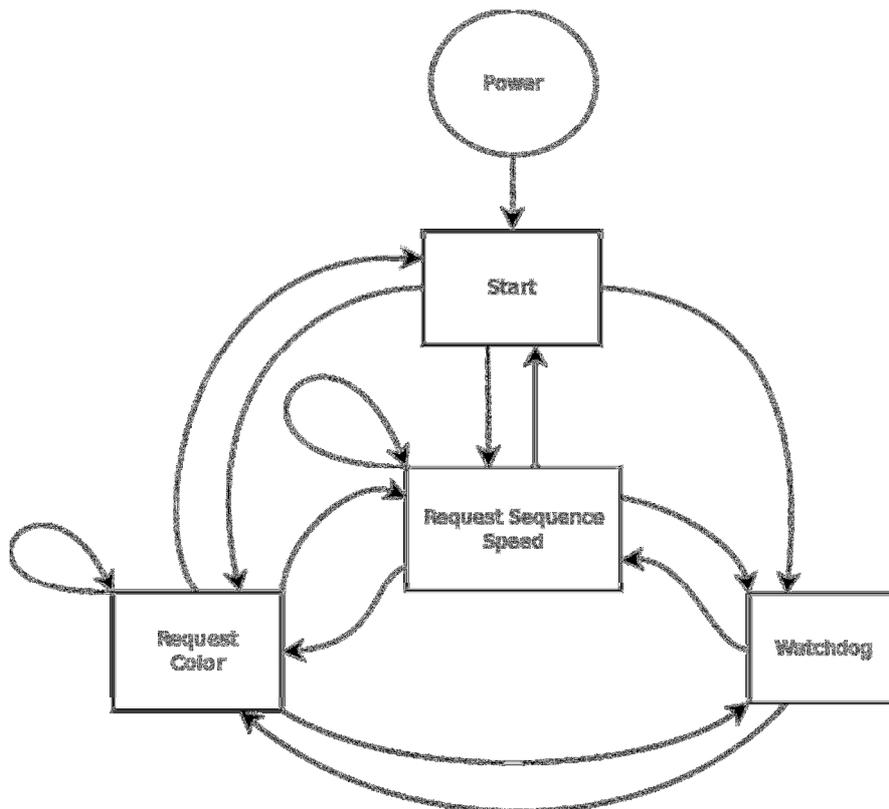


Diagram 1 State Machine Diagram

7.3. ADDRESS AND BAUD RATE SELECTION

7.3.1. ADDRESS SETTING

The address of the Lumiplus in the bus is set through the 0x00 Holding Register.

ID_Address: Address of the Lumiplus in the bus.

Factory setting: 0x30.

Suggested range: 0x30 - 0x3F.

The factory default for the Lumiplus is (0x30). However you can change this value by sending writing this holding registers and as far as you check to do not introduce collisions or conflicts with other slave's addresses.

Example: changing the address ID from 0x30(default) to 0x31.

Transmit Message: 30 10 00 00 00 01 02 00 31 3E 15

Where:

- 30 is the slave address. (The actual ID address).
- 10 is the function used. Write Multiple Registers.
- 00 00 is the address of the Holding Register to be written.
- 00 01 is the number of Holding Registers to be written. 1 in this case.
- 02 is the number of bytes of data to be sent.
- 00 31 is the new ID address.
- 3E 15 is the CRC.

7.3.2. BAUD RATE SELECTION

The Baud Rate selection of the serial communications with the Lumiplus set through the 0x01 Holding Register. By default, 9600 bps and 8E1 (8 data bits, Even Parity, 1 stop bit) is implemented. However, 19200 bps and 2 stop bits are also supported (when NO parity is implemented).

The reason for supporting N2 frames is to keep the MODBUS standard requirement of sending eleven bits per byte (1 start + 8 data + 1 parity + 1 stop). Whenever a No parity configuration is chosen, then 2 stop bits are introduced to keep the eleven bits per byte required by the standard.

For compatibility reasons, N1 frames are also supported. However, mind that using this selection you are not fulfilling the MODBUS standard requirements as far as only ten bits per byte are used.

According to this, the baud rate and frame selection is completed defining the baud rate (in bauds), number of data bits, parity and number of stop bits.

COM_Setup:	Communication setup	
Factory setting:	0	9600, 8E1
Supported values:	0	9600, 8E1
	1	19200, 8E1
	2	9600, 8N2
	3	19200, 8N2
	4	9600, 8N1
	5	19200 8N1

7.4. BROADCASTING

Broadcasting is not supported by the Lumiplus.

7.5. WATCHDOG

The watchdog is a feature implemented in the Lumiplus aimed to check if the communications in the bus keep alive. This is mainly a safety feature intended to detect a fail in the communications, and allows changing to a predefined state.

The Watchdog is triggered when the device does not receive two messages in less than watchdog time specified.

According to these descriptions, either the time considered for triggering the watchdog as well as how to proceed then, needs to be defined.

If the watchdog has been activated, any request through Modbus will not be made. In this State, the Lumiplus only can be read, and will not accept changes. The alarm of the watchdog in holding register 0x20 will be at 1; only setting this alarm to 0 through Modbus or switching OFF the power supply, Lumiplus will allow changes or requests.

To allow correct errors if a bad configuration of the watchdog has been done, the first 30 seconds of the Power ON of Lumiplus, the watchdog will be stopped. This allows the user to change the watchdog time if a short time has been configured, or allows to deactivate the watchdog.

7.5.1. WATCHDOG TRIGGERING

The watchdog triggering time is defined in the Holding Register 0x10. This time is set in seconds. 0s means watchdog is disabled. This is the default value.

To enable the watchdog feature, set the watchdog time to a value different from 0. However, mind the implications of this value as long times may deal to useful setups.

Example: Set watchdog triggering time to 30s:

Transmit Message: 30 10 00 10 00 01 02 00 1E 7D 59

Where:

30 is the slave address.

10 is the function used. Write Multiple Registers.

00 10 is the address of the first Holding Register to be written.

00 01 is the number of Holding Registers to be written. 1 in this case.

02 is the number of bytes of data to be sent.

00 1E is the value to be sent. 30 in decimal.

7D 59 is the CRC.

Now, the watchdog time is set to 30s. Therefore, whenever two properly constructed messages are read in less than 30s, even not addressed to the Lumiplus, the watchdog is not triggered. Otherwise, it is triggered.

To know the Watchdog time, is necessary read the holding registers.

The response of the Watchdog_time, in this case, will be: 30 03 02 00 1E 45 88

30 is the slave address.

03 is the function used. Read holding registers.

02 is the number of bytes of data to be read.

00 1E is the time configured. 30 in decimal.

45 88 is the CRC.

7.5.2. WATCHDOG SETUP

The Watchdog Setup, register 0x11, must be setup in conjunction with the Watchdog triggering register, and the watchdog predefined color/sequence.

The High Byte of the register defines how to proceed when the Watchdog is triggered. If it is set to 0, then the Watchdog predefined color/sequence will be considered. If it is set to 1, then the communication bridge is reset.

The Low Byte is not considered in this function. Set it to 0.

Example: Configuring the watchdog to do the defined in the watchdog predefined color/sequence:

Transmit Message: 30 10 00 11 00 01 02 00 00 FC 80

Where:

30 is the slave address.

10 is the function used. Write multiple registers.

00 11 is the address of the first register to be written.

00 01 is the number of registers to be written. 1 in this case

02 is the number of bytes to be sent.

00 High Byte. The watchdog will do the defined in the Watchdog predefined color/sequence.

00 Low Byte. Is not considered, 00 is a correct value.
FC 80 is the CRC.

7.5.2 Watchdog SETUP and 7.5.3 Watchdog predefined Color/Sequence should be configured in conjunction. See the example 7.5.3 Watchdog predefined Color/Sequence, defining the Color / Sequence to be executed when watchdog triggers.

7.5.3. WATCHDOG PREDEFINED COLOR/SEQUENCE

The watchdog predefined color sequence, Holding Register 0x14, determines the color or the sequence predefined when the watchdog triggers.

The **High byte** set the **speed** of the sequence, it can take values from 0 to 8. If it is set to 0, the speed will be 0, which means that a color is fixed.

The **Low Byte** sets the **color** if the High Byte is 0, or the **sequence** if a different value of 0 is set in the High Byte. The Low byte, can take values from 0 to 12, these are 12 possible colors or 8 possible sequences, to be chosen. If the Low Byte is set to 0 no color has been selected, which means that the Lumiplus has no lights turned ON.

If the register is configured like (0xFFFF), the luminaries remain with the color / sequence previous of the alarm state.

Example:

Set watchdog predefined color/sequence in case that watchdog is triggered, in this example the Lumiplus will has all the lights turned OFF, and the Status when triggers will be Watchdog:

Transmit Message: 30 10 00 14 00 01 02 00 00 FC D5

Where:

30 is the slave address.

10 is the function used. Write multiple registers.

00 14 is the address of the first register to be written.

00 01 is the number of registers to be written.. 1 in this case.

02 is the number of bytes to be sent.

00 High Byte. is the speed, 0 is a fixed color.

00 Low Byte. Color 0 chosen. (Color 0 only exist in the watchdog predefined color/sequence, and in the Start predefined color/sequence) there is no color, which means turn OFF the lights.

F5 D5 is the CRC.

NOTE: Mind that watchdog function is implemented for safety reasons. However, the reasons for activating it, or not, depends on the installer/integrator criteria. Always, mind of implications on what is being implemented. Use this feature under your responsibility.

8. OPERATION MODES

8.1. BASIC MODE

In this section it is assumed that a successful connection has been established with the Lumiplus and therefore, address, baud settings and watchdog behavior has been already set.

8.1.1. START PREDEFINED COLOR/SEQUENCE

The Start Predefined Color/Sequence, is the color or sequence in which will work the Lumiplus when it Powers ON.

The **High byte** set the **speed** of the sequence, it can take values from 0 to 8. if it is set to 0, the speed will be 0 which means that a color is fixed.

The **Low byte** sets the **color** if the High byte is 0, or the **sequence** if a different value of 0 is set in the High Byte. The Low byte can take values from 0 to 12, these are 12 configured colors, or 8 sequences. If it is set to 0, any light will be ON when the Lumiplus powers ON.

If the register is configured like (0xFFFF), the luminaries remain with the color / sequence previous of the last OFF.

The way to setting this parameter is the same that the Watchdog predefined color sequence, the only difference is the number of the holding register, in this case the holding register is 0x15.

In the Start Predefined Color/Sequence and in the Watchdog predefined Color/Sequence, is not necessary to make a request update to change the color, when the system Starts or, the watchdog actuates.

Specific color requests and sequences are defined according to the following codification:

COLOUR N°	COLOUR
1	Red
2	Green
3	Blue
4	Yellow
5	Cyan
6	Magenta

COLOUR N°	COLOUR
7	Pale violet
8	Sky-blue
9	Orange
10	Pink
11	Emerald green
12	White

SEQUENCE	ORDER OF COLOURS												
1	Red	Green	Blue										
2	Cyan	Magenta	Yellow										
3	Green	Emerald green	Cyan	Sky-blue									
4	Red	Orange	Green	Orange									
5	Red	Pink	Blue	Pink									
6	Red	Orange	Green	Cyan	Blue	Pink							
7	Pale violet	Magenta	Cyan	Yellow	White	Yellow	Cyan	Magenta					
8	Red	Green	Blue	Yellow	Cyan	Magenta	Pale violet	Sky-blue	Orange	Pink	Emerald green	White	

8.1.2. CHECKING CURRENT STATE

The state in which the Lumiplus is currently set is available through *Status* Input Register 0x00. This register has a different meaning for the Low byte and for the High byte.

The less significant bit is used to show if an error has occurred and is set to 1 whenever an error exist. Detailed information of the error/errors detected can be requested to the Alarms Input register at 0x01. Only the watchdog alarm has been implemented, which means that if an error exist, surely is a watchdog error (watchdog has been triggered). Further to the raising of an error the Lumiplus report an unrecoverable error that needs to power off-power on to reset, or set the alarm to 0.

Until the alarms have been reset, the Lumiplus will not accept any change, and only will response information (read only).

The High Byte is used to show the current state of the Lumiplus.

The codes for the different states implemented in the high byte are shown in the following table:

- 0x00 Start.
- 0x01 Watchdog.
- 0x02 Request Color.
- 0x03 Request Sequence.

Mind the unidirectionality described in chapter 7.1 General Description.

Coding Examples (High byte first):

- 0x0002 2 Start State. ON, No error.
- 0x0101 257 Watchdog State, Alarm. Light OFF. Reset needed. (This casuistic is possible when the watchdog predefined color/sequence is configured to turn OFF the lights).
- 0x0103 259 Watchdog State, Alarm. Light ON. Reset needed. (This casuistic is possible when the watchdog predefined color/sequence is a color or sequence).

Example:

To know in which State is the Lumiplus, the transmit message must be:

30 04 00 00 00 01 35 EB

Where:

- 30 is the slave address.
- 04 is the function used. Read input registers.
- 00 00 is the address of the first input register to be read.
- 00 01 is the number of input registers to be read, 1 in this case.
- 35 EB is the CRC

The response is 30 04 02 00 02 45 35, that indicates Start Mode, without alarm, and light ON.

Where:

30 is the slave address.
04 is the function used. Read input registers.
02 is the number of bytes to be read.
00 is the high byte, it indicates that is in Start Mode
02 is the low byte, it indicates some light is ON.
45 35 is the CRC.

In the example below, the time watchdog has been activated and the time without communications has exceeded the watchdog time:

The response is 30 04 02 01 01 04 A4, that indicates Watchdog State with watchdog alarm.

Where:

30 is the slave address.
04 is the function used. Read input registers.
02 is the number of bytes to be read.
01 is the high byte, it indicates that is in Watchdog state.
01 is the low byte, it indicates that there is some alarm.
04 A4 is the CRC.

8.1.3. ACTUATOR CHANGE REQUEST

Actuator change request can be sent to the Lumiplus through the Holding Register 0x21. However, these requests can also be sent using a bit address mode starting on coil bit 0x210. Mind that you can choose either register address mode or bit address mode. However, for clarification purposes and to avoid misunderstandings, you should not mix both approaches as far as you do not have a clear idea on what are you doing.

8.1.4. REQUEST ON

To turn ON the lights of Lumiplus through MODBUS, the simplest way is setting to 1 the coil 0x210.

Example, bit address mode:

Transmitting message: 30 0F 02 10 00 01 01 01 ED AE

Where:

30 is the slave address.
0F is the function used. Write multiple coils.
02 10 is the address of the first coil to be set.
00 01 is the number of coils to be set, 1 in this case.
01 is the number of bytes of data to be sent.
01 is to set the request, and then turn ON the lights.
ED AE is the CRC.

Example, byte mode, Holding Register:

Transmitting message: 30 10 00 21 00 01 02 00 01 38 B0

Where:

30 is the slave address.

10 is the function used. Write multiple registers.

00 21 is the address of the first Holding Register to be set.

00 01 is the number of Holding Register to be set, 1 in this case.

02 is the number of bytes of data sent

00 01 is the data send.

38 B0 is the CRC

8.1.5. REQUEST OFF

To turn OFF the lights of Lumiplus through MODBUS, the simplest way is setting to 1 the coil 0x211.

Example:

Transmitting message 30 0F 02 11 00 01 01 01 D0 6E

Where:

30 is the slave address.

0F is the function used. Write multiple coils.

02 11 is the address of the first coil to set.

00 01 is the number of coils to be set, 1 in this case.

01 is the number of bytes of data sent.

01 is to set the request, and then turn OFF the lights.

D0 6E is the CRC.

Also, is possible to make a Request OFF by setting the Holding Register 0x21 to 0x02

8.1.6. CHANGING COLOR

To change the color of the Lumiplus through Modbus, is necessary to complete two steps, these steps, where not necessary in the Watchdog and Start Predefined Color/Sequence.

- 1. It is necessary to choose the predefined color.**
- 2. It is necessary to make a Request Update Predefined Color.**

Choosing the predefined color:

The way to choose the color is using the Requests Holding Register (0x25).

There is 12 predefined colors to choose. From 0x1 to 0xC (from 1 to 12) according to the table above.

COLOUR N°	COLOUR
1	Red
2	Green
3	Blue
4	Yellow
5	Cyan
6	Magenta

COLOUR N°	COLOUR
7	Pale violet
8	Sky-blue
9	Orange
10	Pink
11	Emerald green
12	White

Example:

Transmitting message 30 10 00 25 00 01 02 00 01 39 34

Where:

30 is the slave address.

10 is the function used. Write multiple registers.

00 25 is the address of the first Holding Register to be set.

00 01 is the number of Holding Register to be set, 1 in this case.

02 is the number of bytes of data sent

00 01 is the color send (Red)

39 34 is the CRC

Request Update Predefined Color:

A way to request an Update Predefined Color is using the bit-address mode on the Requests Holding Register 0x21. The simplest way is setting to 1 the coil 0x213.

Transmitting message: 30 0F 02 13 00 01 01 01 A9 AE

Where:

30 is the slave address.

0F is the function used. Write multiple coils.

02 13 is the address of the first Coil to be set.

00 01 is the number of Coils to be set, 1 in this case.

01 is the number of bytes of data sent

01 is the request Update

A9 AE is the CRC

Also, is possible to make a Request Update predefined Color by setting the Holding Register 0x21 to 0x08.

8.1.7. CHANGING SEQUENCE

To change the Sequence of the Lumiplus through Modbus, is necessary to complete two steps.

1. **It is necessary to choose the predefined sequence.**
2. **It is necessary to make a Request Update Sequence & Speed.**

Choosing the predefined sequence:

The way to choose the sequence is using the Requests Holding Register 0x26.

There is 8 predefined sequences to be chosen. From 0x1 to 0x8 (from 1 to 8).

Example: changing the sequence to 8.

Transmitting message 30 10 00 26 00 01 02 00 08 F9 01

Where:

30 is the slave address.

10 is the function used. Write multiple registers.

00 26 is the address of the first Holding Register to be set.

00 01 is the number of Holding Registers to be set, 1 in this case.

02 is the number of bytes of data to be sent.

00 08 is the sequence to be send (the sequence 8).

F9 01 is the CRC.

Request Update Predefined Sequence & Speed:

A way to request an Update Predefined sequence & speed is using the bit-address mode on the Requests Holding Register 0x21, setting to 1 the coil 0x214.

Transmitting message: 30 0F 02 14 00 01 01 01 1C 6E

Where:

30 is the slave address.

0F is the function used. Write multiple coils.

02 14 is the address of the first Coil to be set.

00 01 is the number of Coils to be set, 1 in this case.

01 is the number of bytes of data to be sent.

01 is the request Update.

1C 6E is the CRC.

Also, is possible to make a Request Update predefined Sequence & Speed by setting the Holding Register 0x21 to 0x10.

8.1.8. CHANGING SPEED

To change the Speed of the Lumiplus through Modbus, is necessary to complete two steps.

1. **It is necessary to choose the predefined speed.**
2. **It is necessary to make a Request Update Sequence & Speed.**

Choosing the predefined speed:

The way to choose the sequence is using the Requests Holding Register (0x27).

There is 8 predefined speed to choose. From 0x1 to 0x8 (from 1 to 8).

Example: changing the speed to 8.

Transmitting message 30 10 00 27 00 01 02 00 08 F8 D0

Where:

30 is the slave address.

10 is the function used. Write multiple registers.

00 27 is the address of the first Holding Register to be set.

00 01 is the number of Holding Registers to be set, 1 in this case.

02 is the number of bytes of data sent.

00 08 is the speed send (speed 8).

F8 D0 is the CRC.

Request Update Predefined Sequence & Speed:

A way to request an Update Predefined sequence & speed is using the bit-address mode on the Requests Holding Register 0x21. The simplest way is setting to 1 the coil 0x214.

Transmitting message: 30 0F 02 14 00 01 01 01 1C 6E

Where:

30 is the slave address.

0F is the function used. Write multiple coils.

02 14 is the address of the first Coil to be set.

00 01 is the number of Coils to be set, 1 in this case.

01 is the number of bytes of data to be sent.

01 is the request Update.

1C 6E is the CRC.

Also, is possible to make a Request Update predefined Sequence & Speed by setting the Holding Register 0x21 to 0x10.

8.1.9. SLEEP TIME

Lumiplus has a Sleep mode that can be configured. After the time configured, the Lumiplus will turn all the lights OFF. The user can choose one of the 8 sleep times configured, and there is no possible to choose a different time of these. If it is set to 0, no Sleep mode has been activated.

There is 8 predefined times to choose. From 0x1 to 0x8 (from 1 to 8):

0	No sleep mode.
1	5 minutes.
2	15 minutes.
3	30 minutes.
4	60 minutes.
5	90 minutes.
6	120 minutes.
7	240 minutes.
8	480 minutes.

Example, changing the time of sleep to 1 (5 minutes):

Transmitting message 30 10 00 28 00 01 02 00 01 38 29

Where:

30 is the slave address.

10 is the function used. Write multiple registers.

00 28 is the address of the first Holding Register to be set.

00 01 is the number of Holding Registers to be set, 1 in this case.

02 is the number of bytes of data sent.

00 01 is the time send (time 1, 5 minutes).

38 29 is the CRC.

8.1.10. CHECKING LATCHED ALARMS REGISTER

It is possible to check which alarm has been activated since last alimentation disconnection or since last Alarm reset. Only the Watchdog alarm has been implemented. To check it, Holding Register 0x20 must be read, the bits 0 from 14 are not used, but can be read, and the bit15 is the Watchdog alarm.. The values in 1, indicates that the alarm has been activated.

If the alimentation fails, all the alarms will be at 0.

Example: reading latched alarms with holding registers when watchdog alarm has been activated

Received message: 30 03 02 80 00 A4 40

Where:

30 is the slave address.

03 is the function used. Read Holding Registers.

02 is the number of bytes of data to be read.
80 is the high byte of data read, it indicates Watchdog Error
00 is the low byte of data read, it indicates that does not have alarms in the low byte.
A4 40 is the CRC.

Also, the Latched alarms can be checked with bit mode with Coils. The bit-interpretation of this error is shown below, the coils 0x200 to 0x20E are not used, but it can be read without limitation:

Coil 0x20F MODBUS Watchdog

Example: reading latched alarms with coils when watchdog alarm has been activated:

The received message is: 30 01 01 01 9E B4

Where:

30 is the slave address.
01 is the function used. Read Coils.
01 is the number of bytes of data to be read.
01 indicates that the alarm has been activated.
9E B4 is the CRC.

The way to reset the alarms is setting by 0 the holding register 0x20.

Example, reset all the alarms:

Transmitting message: 30 10 00 20 00 01 02 00 00 F8 A1

Where:

30 is the slave address.
10 is the function used. Write multiple registers.
00 20 is the address of the first Holding Register to be set.
00 01 is the number of Holding Register to be set, 1 in this case.
02 is the number of bytes of data sent.
00 00 is the data send. (0 to set to 0 the alarms).
F8 A1 is the CRC.

8.1.11. CHECKING INSTANTANEOUS ERROR REGISTER

Whenever errors exist, the less significant bit in the Status Input Register 0x00 is set to 1.

Furthermore, for having a more detailed information on the error occurred, these can be requested to the Input Register 0x01, or reading from Digital Input 0x010 to 0x01F, in this case, as only the Watchdog alarm is implemented, the Digital Input 0x01F is the only alarm that can change to 1. (It corresponds to the watchdog alarm).

The bit-interpretation of the Input Register 0x01 error register is shown below, the bits 0 to 14 are not used, but it can be read without limitation, while .bit15 shows the Watchdog MODBUS error.

9. BASIC MODBUS-RTU REGISTER MAP

The table shown in this chapter is our exclusive and original register map with the name of the function and their address.

Furthermore to the register map itself, it is important to understand the relationship between Error words (Holding Register 0x20) and Instantaneous alarms (Input Register 0x01).

There is also a direct relationship between Holding Register 0x20 and Input Register 0x01. While Holding Register 0x20 is the total alarms latched, the Input Register 0x01 is the current alarm. This relationship also applies to the bit-to-bit relation between registers. Only the Watchdog alarm has been implemented, which means that the Latched alarms and instantaneous alarms, always match.

To reset the alarm errors, it is necessary to reset it from the Holding Register 0x20 and not from Input Register 0x01, due to, the Input Register 0x01 will reset when the current error alarm disappears. To reset all the alarms, it is necessary to set to 0 from Coil 0x200 to 0x20F. Note: a disconnection of the power supply will reset all the latched alarms.

Name	Holding Registers	Input Registers	Coils	Inputs	Description
ID_Address	0x00				This parameter is the address of the Lumiplus, by default it is set to 0x30. If the installation has more than one device connected in the bus, is necessary to change the ID_Address of the device before connecting another one.
COM_Setup	0x01				This parameter determinates the Baud rate selection. There are 6 possible configurations to choose: Factory setting: 0. 9600, 8E1 Supported values: 1. 19200, 8E1 2. 9600, 8N2 3. 19200, 8N2 4. 9600, 8N1 5. 19200, 8N1
ID_Manufacturer_hi	0x02				This parameter indicates the high byte that represents the Manufacturer code.
ID_Manufacturer_lo	0x03				This parameter indicates the low byte that represents the Manufacturer code.
ID_Product_code_hi	0x04				This parameter indicates the high byte that represents the Product code.
ID_Product_code_lo	0x05				This parameter indicates the low byte that represents the Product code.
Reserved	0x06				Reserved.
HW_Version	0x07				This parameter indicates the Hardware Version of the Lumiplus.
SW_Version	0x08				This parameter indicates the Software Version of the Lumiplus.
MODEL_Serie_hi	0x09				This parameter indicates the high byte of the Serial Number.
MODEL_Serie_lo	0x0A				This parameter indicates the low byte of the Serial Number.
MODEL_Production_hi	0x0B				This parameter indicates the high byte of the production batch.
MODEL_Production_low	0x0C				This parameter indicates the low byte of the production batch.

Name	Holding Registers	Input Registers	Coils	Inputs	Description
Watchdog_time	0x10				This parameter defines the time in seconds that can elapse between the last communication and the actuation of the watchdog. If it is set to 0, the Watchdog is disabled. Range: 0 – 65535.
Watchdog_config State when overflow WD Config of working mode	0x11 low byte high byte				This parameter defines how will actuate the watchdog if the communications fail. The Low Byte is not considered. It should be at 0. The High Byte is the configuration of the working mode. 0 = Defined in Watchdog predefined color/sequence. 1 = module reset.
WDT predefined color/sequence Color/Sequence Speed	0x14 low byte high byte				This parameter defines the color or sequence that will be activated when the Watchdog triggers. 0xFFFF remains with the color / sequence previous to the alarm state. The Low Byte is the color if Speed=0, or sequence if Speed is different of 0. The High Byte: Speed: 0 = Color Fixed. 1 to 8 = Sequence.
Start predefined color/sequence Color/Sequence Speed	0x15 low byte high byte				This parameter defines the color or sequence of the Lumiplus when Starts. 0xFFFF remains with the color / sequence previous to the last OFF. The Low Byte is the color if Speed = 0; or sequence if Speed is different of 0. The High Byte: speed: 0 = Color Fixed. 1 to 8 = Sequence.

Name	Holding Registers	Input Registers	Coils	Inputs	Description
Status		0x00			This parameter indicates in which State is the Lumiplus, and if there is any alarm.
Alarm ON		low byte .bit 0 .bit 1 .bit2..7		0x00 0x01	The Low byte indicates the alarm and if some light is ON. Some alarm has been activated (if it is set to 1). Some light is ON (if it is set to 1). Not Used.
0: Start state 1: Watchdog state 2: Request Update Predefined Color 3: Request Update Sequence & Speed		high byte			The High Byte indicates the actual State.
Latched Alarm Watchdog	0x20 .bit1..14 .bit 15		0x20F		This parameter indicates which alarm has been activated. Not Used.
Instantaneous Alarm Watchdog		0x01 .bit1..14 .bit 15		0x01F	This parameter indicates the instantaneous alarms. Not Used.
Actuators 0x00 Request ON Request OFF Request Update Predefined Color Request Update Sequence&Speed	0x21 .bit 0 .bit 1 .bit 2 .bit 3 .bit 4 .bit5..15		0x210 0x211 0x212 0x213 0x214		These parameters are the possible requests. Reserved. Not Used.

Name	Holding Registers	Input Registers	Coils	Inputs	Description
Predefined color	0x25				This parameter determinates the color of the light. Once chosen, a request update color is necessary to see the color change. The colors are from 1 to 12.
Predefined Sequence	0x26				This parameter determinates the sequence of the light. Once chosen, a request update sequence & speed is necessary to see the sequence change. The sequence are from 1 to 8.
Sequence speed	0x27				This parameter determinates the speed of the variation of the light. Once chosen, a request update sequence & speed is necessary to see the speed change. The speeds are from 1 to 8.
Sleep	0x28				This parameter determinates the time in which the light will turn OFF. Is not necessary any request update. The sleep time are from 1 to 8.

10. TROUBLESHOOTING GUIDE

Problem	Solution
<p>The Master does not receive any response to its requests.</p>	<ul style="list-style-type: none"> a) Verify cable polarity A,B. b) Verify ground connection between Master and Slave. c) With a long bus cable, check the termination resistors according with the chapter 4 of this manual. d) Make sure than the baudrate and the bit settings are the same on Master and Slaves. e) Make sure that the Master is including the CRC in the sending, according with Modbus standard. Check the calculation of CRC. f) Verify that the function requested is supported by the Slave. g) Check the Slave address is the one that you are using in the request. If you are not sure, isolate the Slave (connect it directly to the Master) and execute a "discovery" process in the Master. In case that the master has not this kind of process automatically, check manually the slaves addresses form 1 to 255 with the different baudrates and bit configurations supported by the Slave.
<p>The Slave does not execute any request</p>	<ul style="list-style-type: none"> a) Verify cable polarity A,B. b) Verify ground connection between Master and Slave. c) With a long bus cable, check the termination resistors according with the chapter 4 of this manual. d) Make sure than the baudrate and the bit settings are the same on Master and Slaves. e) Make sure that the Master is including the CRC in the sending, according with Modbus standard. Check the calculation of CRC. f) Verify that the function requested is supported by the Slave. g) Check the Slave address is the one that you are using in the request. If you are not sure, isolate the Slave (connect it directly to the Master) and execute a "discovery" process in the Master. In case that the master has not this kind of process automatically, check manually the slaves addresses form 1 to 255 with the different baudrates and bit configurations supported by the Slave.
<p>The Slave does not execute some requests or the "messages error rate" is too high.</p>	<ul style="list-style-type: none"> a) Make sure you are using twisted shielded cable according to EN50170 (especially with long bus cables or in environments with high electromagnetic pollution, for example near inverters or switching power supplies). b) Verify the cable shielding is connected to ground as in Master as in Slaves. c) Check the termination resistors according with the chapter 4 of

	<p>this manual.</p> <p>d) Configure a lower baudrate in Master and Slaves.</p>
Unexpected WatchDog errors	<p>a) Verify if the "messages error rate" of the bus is too high.</p> <p>b) Verify if the bus is saturated with messages, in this case the time between two messages to the same Slave could be bigger that the WatchDog time configured in the Slave. Increase the WatchDog time in the Slave or increase the baudrate of the bus (Master and Slaves)</p>

11. PRODUCT REVISION

Manual v.2.0 : All the information of this manual, describes the behavior of the Hardware Version 1 and Software Version 1. Includes a troubleshooting guide