



OPTO ENGINEERING



INSTRUCTIONS MANUAL

LTIC1CH-D1-4

Digital lighting controller 1 CH – Firmware version 1.05



ACCESSORIES

INDEX

- 1. Disclaimer 5**
- 2. Safety notes 5**
- 3. Product end-of-life handling 5**
- 4. General description 6**
 - 4.1. Benefits of current control 6
- 5. Getting started 6**
- 6. Mechanical fixing 6**
- 7. Heat dissipation 7**
- 8. Connections 7**
 - 8.1. Layout of connectors 7
 - 8.2. Power supply, enable and dimming inputs, fault output 9
 - 8.2.1. Power supply 9
 - 8.2.2. Enable input 10
 - 8.2.3. Dimming input 10
 - 8.2.4. Fault output 10
 - 8.3. Light output, thermal sensor input and fan output 11
 - 8.3.1. Light output 11
 - 8.3.2. Auxiliary thermal sensor input 11
 - 8.3.3. Auxiliary cooling fan output 11
 - 8.3.3.1. Proportional algorithm 12
 - 8.3.3.2. Hysteretic algorithm 13
 - 8.3.3.3. Manual algorithm 13
 - 8.4. Serial RS485 interface 13
 - 8.5. Cable size and length 13
- 9. Communication interfaces 14**
 - 9.1. Serial RS485 interface 14
 - 9.2. Ethernet interface 14
- 10. Visual indicators 15**
- 11. INIT button functions 16**
- 12. Wiring diagrams 16**
 - 12.1. Wiring example #1: basic connection 16
 - 12.2. Wiring example #2: advanced connection 17
- 13. Operation 18**
 - 13.1. Operation in continuous mode 19
 - 13.2. Operation in pulsed mode 20
 - 13.3. Flickering optimization 21

13.4. Operation with Modbus	21
13.4.1. Comparison of Modbus/RTU, Modbus/TCP and Modbus/UDP	22
13.4.2. Supported function codes	22
13.4.3. Read Holding Registers (0x03).....	22
13.4.4. Write Single Register (0x06).....	22
13.4.5. Write Multiple Registers (0x10).....	23
13.5. Register file	23
13.5.1. Register DEVICE_TYPE.....	27
13.5.2. Register BOOT_VERSION	27
13.5.3. Register MCU_VERSION	27
13.5.4. Register ENABLE_CONTROL.....	27
13.5.5. Register CURRENT_CONTROL	27
13.5.6. Register CURRENT_RANGE	28
13.5.7. Register CURRENT_PRESET	28
13.5.8. Register FAULT_CONTROL.....	29
13.5.9. Register FAULT_BOARD_TMP_LOW	29
13.5.10. Register FAULT_BOARD_TMP_HIGH	29
13.5.11. Register FAULT_LIGHT_TMP_LOW.....	30
13.5.12. Register FAULT_LIGHT_TMP_HIGH.....	30
13.5.13. Register FAN_CONTROL.....	31
13.5.14. Register FAN_TMP_LOW.....	31
13.5.15. Register FAN_TMP_HIGH.....	31
13.5.16. Register FAN_DUTY_INF	31
13.5.17. Register FAN_DUTY_SUP	32
13.5.18. Register FAN_DUTY_MIN	32
13.5.19. Register FAN_DUTY_MAX.....	32
13.5.20. Register FAN_DUTY_MANUAL.....	32
13.5.21. Register FAN_PWM_FREQUENCY	33
13.5.22. Register OSC_CONTROL	33
13.5.23. Register OSC_FREQUENCY	33
13.5.24. Register RS485_MODBUS_ADDR.....	33
13.5.25. Register RS485_LINE_SPEED	34
13.5.26. Register RS485_LINE_PARITY	34
13.5.27. Registers ETH_MAC_ADDR[0:2]	34
13.5.28. Registers ETH_HOSTNAME[0-7].....	35
13.5.29. Register ETH_DHCP_ENABLE	35
13.5.30. Register ETH_IP_ADDR_HI	35
13.5.31. Register ETH_IP_ADDR_LO	35
13.5.32. Register ETH_SUBNET_MASK_HI	35

13.5.33. Register ETH_SUBNET_MASK_LO	35
13.5.34. Register ETH_DEF_GATEWAY_HI	36
13.5.35. Register ETH_DEF_GATEWAY_LO	36
13.5.36. Register ETH_PRI_DNS_HI	36
13.5.37. Register ETH_PRI_DNS_LO	36
13.5.38. Register ETH_SEC_DNS_HI	36
13.5.39. Register ETH_SEC_DNS_LO	36
13.5.40. Register ETH_MODBUS_ADDR	36
13.5.41. Register ETH_MODBUS_TCP_PORT	37
13.5.42. Register ETH_MODBUS_UDP_PORT	37
13.5.43. Registers WEB_PASSWORD[0-3]	37
13.5.44. Register BOARD_TEMPERATURE	37
13.5.45. Register LIGHT_TEMPERATURE	37
13.5.46. Register SUPPLY_VOLTAGE	37
13.5.47. Register DRIVER_CURRENT	38
13.5.48. Register FAN_VOLTAGE	38
13.5.49. Register INTERNAL_VOLTAGE	38
13.5.50. Register LIGHT_VOLTAGE	38
13.5.51. Register LIGHT_CURRENT	38
13.5.52. Register DIMMING_VOLTAGE	39
13.5.53. Register DIPSWITCH_SETTING	39
13.5.54. Register ENABLE_INPUT	39
13.5.55. Register FAULT_FLAG	39
13.5.56. Register FAULT_OUTPUT	39
13.5.57. Register FAN_DUTY	40
13.5.58. Register ERROR_WORD	40
13.5.59. Register BOARD_COMMAND	40
13.6. Operation with a web browser	41
13.6.1. Main page and navigation menu	41
13.6.2. Controller setup	43
13.6.2.1. Controller setup – Main settings	45
13.6.2.2. Controller setup – Fault detection	46
13.6.2.3. Controller setup – Fan settings	47
13.6.2.4. Controller setup – Oscillator settings	48
13.6.3. General setup	48
13.6.4. Advanced setup	50
14. Operation with FabImage	52
14.1. Filter selection	52
14.2. Designing a simple program for reading a register	54

14.3. Designing a simple program for writing a register	55
15. Electromagnetic compatibility	57
16. Firmware update procedure	57

1. Disclaimer

Always deploy and store Opto Engineering products in the prescribed conditions in order to ensure proper functioning. Failing to comply with the following conditions may shorten the product lifetime and/or result in malfunctioning, performance degradation or failure.

Ensure that incorrect functioning of this equipment cannot cause any dangerous situation or significant financial loss to occur. It is essential that the user ensures that the operation of the controller is suitable for their application. All trademarks mentioned herein belong to their respective owners.

Except as prohibited by law:

- All hardware, software and documentation are provided on an “as is” basis
- Opto Engineering accepts no liability for consequential loss, of any kind

Upon receiving your Opto Engineering product, visually examine the product for any damage during shipping. If the product is damaged upon receipt, please notify Opto Engineering immediately.

2. Safety notes

Please read the following notes before using this controller. Contact your distributor or dealer for any doubts or further advice.

This device must not be used in an application where its failure could cause a hazard to human health or damage to other equipment. Keep in mind that if the device is used in a manner not foreseen by the manufacturer, the protection provided by its circuits and by its enclosure may be impaired.

This is a low voltage device. As such, the potential difference between any combination of applied signals must not exceed, at all times, the supply voltage. Higher voltages may cause a fault and can be dangerous to human health.

This device has limited protection against transients caused by inductive loads. If necessary, use external protection devices like fast diodes or, better, specific transient protectors.

The user must be careful to connect the inputs and outputs correctly and to protect the output wiring and load from unintentional short-circuits. When the device is switched off, there is still energy stored in the internal capacitors for at least five minutes.

When operating the controller at the maximum ratings it can get very hot. The controller should be positioned where personnel cannot accidentally touch it and away from flammable materials. Never exceed the power ratings stated in the manual.

3. Product end-of-life handling

Observe the following guidelines when recycling this equipment or its components.

Production of this equipment required the extraction and use of natural resources. The equipment may contain substances that could be harmful to the environment or human health if improperly handled at the product's end of life. In order to avoid release of such substances into the environment and to reduce the use of natural resources, we encourage you to recycle this product in an appropriate system that will ensure that most of the materials are reused or recycled appropriately.



This symbol indicates that this product complies with the applicable European Union requirements according to the **WEEE (Waste Electrical and Electronic Equipment) Directive 2012/19/EU**

4. General description

Any machine vision application employs some kind of light controller. Light controllers are widely used to both optimize illumination intensity and obtain repeatable trigger sequencing between lights and vision cameras.

This controller is a compact unit intended for continuous, low flickering lighting applications in computer vision. It includes power supply conditioning and current intensity control.

Use the 4-way dip switch accessible from the front panel to select the range of the output current according to the light specification. The output current can then be adjusted using either the front panel knob or an external analogue 0-10 V dimming signal. A logic 0-24 V enable input can be used to turn the connected light on and off. A logic 0-24 V fault output is provided to signal overheating or malfunction.

Alternatively to the use of the 4-way dip switch, the front panel knob and the three discrete signals just described, the controller can be fully set up by a PC using the integrated serial RS485 or Ethernet interfaces. Configurations are saved in non-volatile memory so that the controller will resume operation after a power cycle.

An auxiliary thermal sensor input can be used to sense the light temperature (if a thermal sensor is available in the light). An auxiliary fan output can be used to supply an external cooling fan (if a fan is available in the light).

For older firmware versions, please contact us on www.opto-e.com to receive the corresponding manual.

4.1. Benefits of current control

Most LED manufacturers suggest their products to be driven using a constant current source, not a constant voltage source. This is because, using a constant voltage driving, small variations in temperature or voltage at the LEDs can cause a noticeable change in their brightness.

Brightness control with voltage is also very difficult because of the non-linearity of brightness with voltage. On the contrary, the brightness is approximately linear with current, so by driving the LEDs with a known current, intensity control is linear.

This digital controller has a single, programmable, current-controlled continuous output with current ranging from zero up to 4 A.

5. Getting started

Carefully read the sections on [Safety Notes](#) and [Heat Dissipation](#) and check the product fits your needs. Mount the controller using a DIN rail as described in the section on [Mechanical fixing](#).

Connect the controller as in the section on [Connections](#). When the controller powers up it should show the POWER LED lit with a flashing green colour.

Read the section on [Operation](#). The controller can be configured either by using the front panel 4-way dip switch and knob or, alternatively, by using a PC connected to the controller integrated serial RS485 or Ethernet interfaces (see sections [Operation with Modbus](#) and [Operation with a web browser](#)).

6. Mechanical fixing

The controller must be mounted on a DIN rail. Allow free flow of air around the unit. The controller has an IP rating of 20 and should be installed so that moisture and dirt cannot enter it.

An enclosure may also be required for other parts of the system such as power supplies. That enclosure would provide both mechanical and environmental protection in industrial applications.

7. Heat dissipation

The controller integrates several analogue circuits to produce the constant current output. This means that it generates heat which needs to be dissipated. The operating temperature range is 0 °C to 40 °C.

The controller can approximately dissipate an average power of about 10 W.

The maximum permissible controller temperature is 90 °C. If the temperature rises above 90 °C, the controller switches off the light output. The light output is then reactivated once the temperature falls below 70 °C.

There are several ways to reduce the heat generated by the controller. The simplest way would be to turn the light off when not needed. If the light is on only when necessary, the generated heat can be drastically diminished. Another opportunity would be to reduce the output current, if permitted by the application.

The controller must be powered with a fixed supply voltage of either 24 V or 48 V DC. Intermediate voltages are strictly not allowed. As a general guideline, use either 24 V or 48 V if the light power is less than 50 W and use 48 V if the light power is 50 W or more.

8. Connections

See the next sections for information about connections. All connections are made via screw terminals on both the top and bottom panels of the controller (front view). Check all connections carefully before switching on the equipment.

As anticipated, the controller has a single 24 V or 48 V DC power supply. Intermediate voltages are not supported and must be avoided. Use either 24 V or 48 V if the light power is less than 50 W and use 48 V if the light power is 50 W or more.

The controller has an external analogue 0-10 V dimming input that can be used to adjust the value of the output current and consequently the intensity of the light emitted.

The controller has a logic 0-24 V enable input that can be used to switch the light output on and off. The enable input can be used to reduce power dissipation or to protect the end user from photobiological and other hazards that can occur during fault conditions. The enable input is active high.

The controller has a logic fault output, used to signal overheating or malfunction. This output is internally pulled up to the supply voltage (either 24 V or 48 V). The fault output is active low.

An auxiliary thermal sensor input and an auxiliary cooling fan output are also provided as signals on the terminal blocks.

The controller has two communication interfaces available on the bottom panel: a serial RS485 port and an Ethernet RJ45 jack. These interfaces can be used to remotely configure the controller operation in general and specifically adjust the output current.

8.1. Layout of connectors

The drawing in *Figure 1: connectors on the controller* depicts all the controller connections, which are easily accessible on both the top and bottom sides of the controller (front view). As indicated in the drawing, connectors are identified by their unique designators (P1, P2, P3, P4 and P5).

The connectors are briefly described below. A detailed description follows in the next sections.

- Connector P1 is used for:
 - The power supply
 - The enable input signal
 - The external dimming input signal

- The fault output signal
- Connector P2 is used for:
 - The light
 - An auxiliary thermal sensor (generally mounted inside the light)
 - An auxiliary fan for the active cooling of the light
- Connector P3 is a USB port (B type), not active at the moment
- Connector P4 is a serial RS485 port
- Connector P5 is an Ethernet RJ45 jack

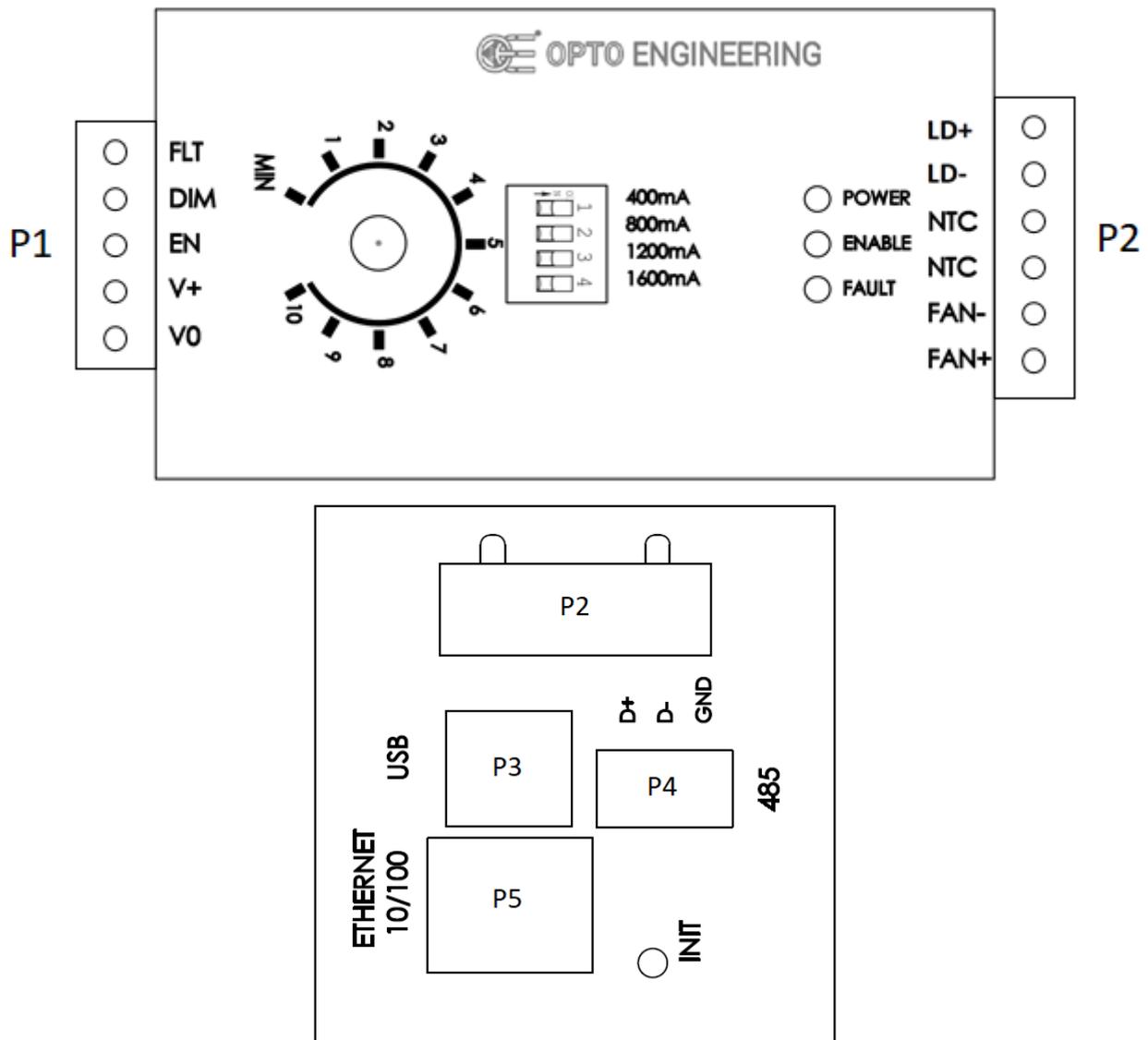


Figure 1: connectors on the controller

For connectors P1, P2 and P4 a mating plug is provided in the controller package. For convenience the relevant manufacturer part numbers are listed in *Table 1: mating plugs for the controller connectors*. Even if equivalent mating plugs may be available, these are the recommended components.

Connector designator	Manufacturer	Mating plug part number
P1	Phoenix Contact	1792979
P2	Phoenix Contact	3705390
P4	Phoenix Contact	1803581

Table 1: mating plugs for the controller connectors

8.2. Power supply, enable and dimming inputs, fault output

Connector P1 (see *Figure 1: connectors on the controller*) is used for the power supply, the enable input, the dimming input and for the fault output.

Connector pinout, ordered from left to right, is listed in *Table 2: pinout of connector P1*.

Number	Name	Description	Note
1	FLT	Fault output	Internally pulled up to supply voltage (either 24 V or 48 V)
2	DIM	Dimming input	Used to adjust the output current
3	EN	Enable input	Connect to 12 V or greater to turn on the controller output and supply the light
5	V+	Power supply. Positive terminal	
6	V0	Power supply. Negative terminal	

Table 2: pinout of connector P1

See the following section for more information about connector P1.

8.2.1. Power supply

The power supply voltage must be 24 V or 48 V. Intermediate voltages are not allowed and must be avoided. Use either 24 V or 48 V if the light power is less than 50 W and use 48 V if the light power is 50 W or more. A dedicated and well-regulated switching power supply is required. The external power supply must be capable of supplying the current needed for the light at the selected supply voltage.

The controller is not equipped with an internal fuse, so choose a power supply unit that limits its output current by design or use a suitable protecting fuse. The maximum allowed current must be 6 A. The fuse should be appropriately de-rated if mounted in an enclosure, as the inside temperature can be higher than the ambient temperature.

Ensure that the wire gauge used for the power connections is appropriate for the current to be drawn. The power supply low voltage and mains wirings should be separately routed.

Power supply is delivered to the controller using pin 5 (V+) and pin 6 (V0) of the screw terminals of connector P1. Connector pinout, ordered from left to right, is listed in *Table 2: pinout of connector P1*.

Ensure that the polarity of V+ and V0 is correct. When the controller is powered the POWER green LED on the top panel flashes periodically.

8.2.2. Enable input

The controller has an enable input that can be used to turn the light output on and off. To activate the output and switch the light on, connect the enable input to a voltage source of 12 V or greater. The input impedance of this signal is 20 k Ω .

The enable input signal is delivered to the controller using pin 3 (EN) and pin 6 (V0) of the screw terminals of connector P1. Connector pinout, ordered from left to right, is listed in *Table 2: pinout of connector P1*.

When the controller is enabled, the ENABLE yellow LED on the top panel is continuously lit.

Alternatively to using the discrete enable input signal, the controller may be enabled and disabled using the serial RS485 and Ethernet interfaces (see section [Operation](#)).

8.2.3. Dimming input

The controller has three different ways to adjust the value of the output current and consequently the intensity of the light. The basic way is to use the knob mounted on the controller front panel (see section [Operation](#)). The alternative way is to use the external analogue dimming input. The third way is to use the serial RS485 and Ethernet interfaces (see sections [Operation with Modbus](#) and [Operation with a web browser](#)).

If the external dimming input is used, the output current can be adjusted by applying an analogue signal (from 0 V up to 10 V) between pin 2 (DIM) and pin 6 (V0) of the screw terminals of connector P1. Connector pinout, ordered from left to right, is listed in *Table 2: pinout of connector P1*.

If the dimming signal is driven by the minimum voltage (0 V) the controller will deliver the lowest output current, corresponding to the lowest light intensity.

If the dimming signal is driven by the maximum voltage (10 V) the controller will deliver the highest output current, as selected by 4-way dip switch, corresponding to the highest light intensity.

For the dimming voltages between 0 V and 10 V the controller will deliver an intermediate output current (from the lowest output current up to the maximum current value selected by 4-way dip switch), corresponding to an intermediate light intensity.

The input impedance of this signal is 4.7 k Ω .

8.2.4. Fault output

The controller has a fault output that is used to signal overheating or malfunction. This output is internally pulled up to the supply voltage (either 24 V or 48 V) using a 10 k Ω resistor and is active low. The maximum sink current is 10 mA.

There are five possible fault sources. By default, some of these fault sources are monitored by the controller and others are ignored. Any of the five fault sources can be selectively enabled or ignored by using the serial RS485 and Ethernet interfaces (see sections [Operation with Modbus](#) and [Operation with a web browser](#)).

If a fault condition is detected the controller output is turned off, the fault output is driven low and the FAULT red LED on the top panel is continuously lit. The controller will automatically resume regular operation once the fault condition is removed.

The controller fault output signal can be monitored using pin 1 (FLT) and pin 6 (V0) of the screw terminals of connector P1. Connector pinout, ordered from left to right, is listed in *Table 2: pinout of connector P1*.

The fault condition can also be monitored by using the serial RS485 and Ethernet interfaces (see sections [Operation with Modbus](#) and [Operation with a web browser](#)).

8.3. Light output, thermal sensor input and fan output

Connector P2 (see *Figure 1: connectors on the controller*) is used for the light output, for an auxiliary external thermal sensor in the light (if available) and to drive an auxiliary cooling fan in the light (if available).

Connector pinout, ordered from left to right, is listed in *Table 3: pinout of connector P2*.

Number	Name	Description	Note
1	LD+	LED anode	
2	LD-	LED cathode	
3	NTC	Thermal sensor	
4	NTC	Thermal sensor	
5	FAN-	Negative terminal for cooling fan	
6	FAN+	Positive terminal for cooling fan	

Table 3: pinout of connector P2

See the following section for more information about connector P2.

8.3.1. Light output

Light output is available on pin 6 (LD+) and pin 5 (LD-) screw terminals of connector P2. Connector pinout, ordered from left to right, is listed in *Table 3: pinout of connector P2*.

Before powering up the controller, ensure that the polarity of LD+ and LD- is correct. Make also sure you set the correct current range for the light (see section [Operation](#)). See the light datasheet and manual for the electrical ratings of the light.

8.3.2. Auxiliary thermal sensor input

The controller allows for the connection of an auxiliary external temperature sensor. The intended temperature sensing element is a NTC (Negative Temperature Coefficient) thermistor with coefficients $R_{25} = 10 \text{ k}\Omega$ and $B_{25/85} = 3610 \text{ K}$. A suitable component is the Vishay NTCS0603E3103FMT.

The thermistor signals are available on pin 3 (NTC) and pin 4 (NTC) screw terminals of connector P2. Connector pinout, ordered from left to right, is listed in *Table 3: pinout of connector P2*.

The two terminals can be connected freely to the external thermistor, as the component is not polarized.

The light temperature, as well as the controller temperature, can be monitored by using the serial RS485 and Ethernet interfaces (see sections [Operation with Modbus](#) and [Operation with a web browser](#)).

8.3.3. Auxiliary cooling fan output

The controller allows for the connection of an auxiliary external cooling fan with a nominal voltage of 24 V and a maximum supply current of 1 A. Inside the controller there is a PWM (Pulse-Width Modulation) driver that allows to control the fan speed indirectly by modulating the pulse width of the fan power supply.

The duty cycle of the fan power supply is generally automatically varied by the controller between a minimum value and a maximum value according to the actual temperature of the light, as measured by the auxiliary thermal sensor, and user settings.

The controller implements three fan control algorithms as follows:

- Proportional (factory default)
- Hysteretic
- Manual

Every detail regarding fan control can be configured by using the serial RS485 and Ethernet interfaces (see sections [Operation with Modbus](#) and [Operation with a web browser](#)).

The following two diagrams show how the proportional and hysteretic fan control algorithms work. Parameters T_{Low} , T_{High} , $DutyInf$, $DutySup$, $DutyMin$ and $DutyMax$ are explained in the next sections.

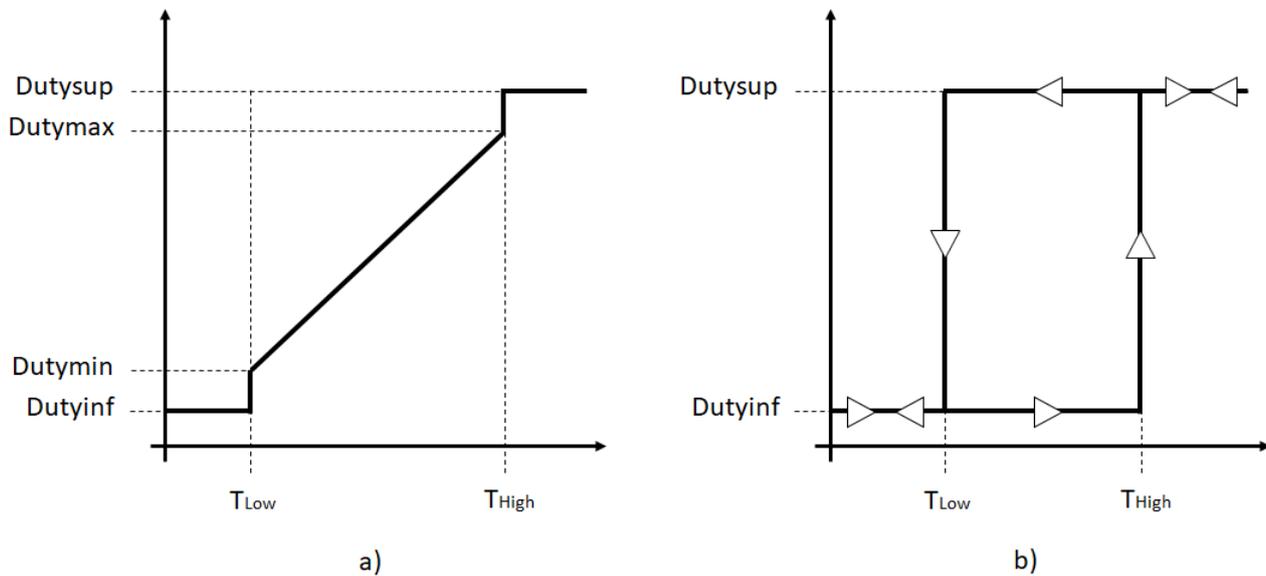


Figure 2: fan control algorithms: a) proportional; b) hysteretic

The fan output signals are available on pin 5 (FAN-) and pin 6 (FAN+) screw terminals of connector P2. Connector pinout, ordered from left to right, is listed in *Table 3: pinout of connector P2*.

8.3.3.1. Proportional algorithm

When the proportional algorithm is selected, the controller drives the fan with a pulse-width modulated power supply whose duty cycle is proportional to the measured temperature (see *Figure 2: fan control algorithms: a) proportional; b) hysteretic*).

There are six parameters that can be set to adapt the controller to a specific fan or application. These parameters are:

- T_{Low} is the low temperature limit
- T_{High} is the high temperature limit
- $DutyInf$ is the duty cycle of the fan power supply for temperature lower than T_{Low}
- $DutySup$ is the duty cycle of the fan power supply for temperature higher than T_{High}
- $DutyMin$ is the duty cycle of the fan power supply for temperature T_{Low}
- $DutyMax$ is the duty cycle of the fan power supply for temperature T_{High}

For temperature between T_{Low} and T_{High} the duty cycle of the fan power supply is calculated as an intermediate value between $DutyMin$ and $DutyMax$, proportional to the measured temperature.

The proportional algorithm is the factory default. Other factory settings are as follow:

- $T_{Low} = 70\text{ }^{\circ}\text{C}$
- $T_{High} = 85\text{ }^{\circ}\text{C}$
- $Duty_{Min}$ and $Duty_{Inf}$ are the same and correspond to 50 %
- $Duty_{Max}$ and $Duty_{Sup}$ are the same and correspond to 100 %

8.3.3.2. Hysteretic algorithm

When the hysteretic algorithm is selected, the controller drives the fan with a pulse-width modulated power supply whose duty cycle toggles between two different, discrete values according to the measured temperature history (see *Figure 2: fan control algorithms: a) proportional; b) hysteretic*).

There are four parameters that can be set to adapt the controller to a specific fan or application. These parameters are:

- T_{Low} is the low temperature limit
- T_{High} is the high temperature limit
- $Duty_{Inf}$ is the duty cycle of the fan power supply for temperature lower than T_{Low}
- $Duty_{Sup}$ is the duty cycle of the fan power supply for temperature higher than T_{High}

For temperature between T_{Low} and T_{High} the duty cycle of the fan power supply is selected between $Duty_{Inf}$ and $Duty_{Sup}$, according to the measured temperature history.

8.3.3.3. Manual algorithm

When manual algorithm is selected, the controller drives the fan with a pulse-width modulated power supply whose duty cycle is programmed by the user.

8.4. Serial RS485 interface

The serial RS485 interface is available on the D+, D- and GND terminals of connector P4. These signals are listed in *Table 4: pinout of serial interface in connector P4*. Be careful not to cross-connect the serial interface signals.

Pin number	Name	Description	Note
1	D+	RS485 data signal. Positive terminal	-
2	D-	RS485 data signal. Negative terminal	-
3	GND	RS485 reference ground	-

Table 4: pinout of serial interface in connector P4

The serial RS485 interface is electrically isolated. Note that GND is not the same as 0V.

8.5. Cable size and length

The actual connecting cables must be chosen on the basis of their load sinking current, the length, the working voltage and the cable materials characteristics. Special ambient conditions may further restrict the choice to a specific kind of cable.

The *Table 5: cable wire size and length* lists the recommended wire sizes and maximum allowed lengths for all the cables coming to and leaving from the controller. American Wire Gauge (AWG) is

the wire measurement system used by the United States and Canada, while mm is the metric system of measurement used across Europe and in most of the world.

Port	Recommended wire size		Maximum length [m]
	mm ²	AWG	
Power supply	1.5	15	5
Light output	0.75	18	5
Control and status signals	0.25	24	5
External fan	0.5	20	5
External temperature sensor	0.25	24	5

Table 5: cable wire size and length

9. Communication interfaces

There are several ways to configure the controller.

A first option is to use the serial RS485 interface. To support this interface the controller implements a subset of the Modbus/RTU (Remote Terminal Unit) slave protocol.

A second option is to use the Ethernet interface. Supported Ethernet speeds are 10 Mbit/s and 100 Mbit/s with auto negotiation. The Ethernet interface allows to configure the controller using the Modbus/TCP (Transmission Control Protocol) slave protocol, the Modbus/UDP (User Datagram Protocol) slave protocol or the HTTP (Hyper Text Transfer Protocol) protocol. For supporting the latter, the controller provides an internal web server accessible by most common web browsers.

The Modbus/RTU, Modbus/TCP and Modbus/UDP protocols are implemented by most programmable logic controllers (PLCs) with a suitable interface.

The availability of two physical interfaces and four logical protocols makes it easy to integrate the controller in most vision applications.

See [chapter 13](#) for details on operation with both Modbus and web browser.

9.1. Serial RS485 interface

For the serial RS485 interface, the controller implements a subset of the Modbus/RTU slave protocol and operates, by default, at 9600 bits per second with even parity. The factory set Modbus address is 32 and it is saved in the controller non-volatile memory.

The Modbus address is one of the controller parameters and can be changed using any of the available interfaces. The factory set Modbus address can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

Please note valid Modbus addresses for slave devices are in the range 1 to 247; remaining addresses are reserved by the standard for special purposes and must not be used. It is of great importance to ensure, at the time of assigning the slave address, that there are not two devices with the same address. In such a case, an abnormal behaviour of the whole serial bus can occur, the master being then in the impossibility to communicate with all the slaves present on the bus.

The activity of the serial RS485 interface is shown by the fast flashing of the green POWER LED on the controller top panel (see section [Visual indicators](#)).

9.2. Ethernet interface

The Ethernet interface allows to configure the controller using the Modbus/TCP slave protocol, the

Modbus/UDP slave protocol or the HTTP protocol. For the last option, the controller provides an internal web server accessible by most common web browsers.

To use the interface, connect the controller using a standard Ethernet cable. The default parameters for the communication are listed in *Table 6: default parameters for Ethernet communication*.

Parameter	Default Value
Host name	LTIC1CH-D1-4
DHCP	Disabled
IP address	192.168.0.32
Subnet mask	255.255.255.0
Default gateway	192.168.0.1
Preferred DNS server	192.168.0.2
Alternate DNS server	192.168.0.2
Modbus address	32
Modbus/TCP port	502
Modbus/UDP port	502

Table 6: default parameters for Ethernet communication

The IP address, subnet mask and DHCP use flag are some of the controller parameters and can be changed using any of the available interfaces. The factory configuration uses the static IP address 192.168.0.32. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

10. Visual indicators

There are three LEDs on the top panel of the controller (see *Figure 1: connectors on the controller*) and two LEDs embedded in the Ethernet RJ45 jack.

The exact meaning of each of the LEDs on the top panel is listed in *Table 7: meaning of the LEDs*. The LEDs of the top panel of the controller are identified by a unique label printed next to them. The Ethernet ACT and LINK LEDs are identified by their position relative to the Ethernet RJ45 jack. The ACT LED is at the left of the jack, while the LINK LED is at the right.

Number	Name	Colour	Description
1	POWER	Green	Blinks periodically when power supply is present. Blinks at a faster rate during serial RS485 communication or when loading factory settings
2	ENABLE	Yellow	Stable when controller is enabled
3	FAULT	Red	Stable when an error condition is detected

Table 7: meaning of the LEDs

The green RUN LED blinks periodically when the power supply is present. The LED blinks at a faster rate during serial RS485 communication or when loading the factory settings. The controller reverts to using the factory settings at power up due to user activation of the INIT button (see next section for more information on the INIT button) or the detection of a corruption in the stored customer settings.

11. INIT button functions

The INIT button is used either to restore the factory settings or to activate the firmware update procedure.

To restore the factory settings, follow these steps:

1. Switch off the device and wait 30 seconds
2. Push and hold down the INIT button
3. Switch on the device
4. Release the INIT button
5. Wait 10 seconds

After the ten seconds interval the settings are restored to the factory values and the controller resumes normal operation.

To activate the firmware update, follow these steps:

1. Switch off the device and wait 30 seconds
2. Push and hold down the INIT button
3. Switch on the device
4. Release the INIT button
5. Launch a firmware update (according to chapter 17) within 10 seconds.

Note the INIT button is sampled only once at power-up.

During the ten seconds interval, the RUN and ERR LEDs blink at a high rate to emphasize the circumstance. In the meantime, the use of the RS485 serial interface is restricted to the firmware update and the Modbus/TCP, Modbus/UDP, and HTTP protocols are not available.

The INIT button is concealed by a hole located between the USB port and the shell connector.

12. Wiring diagrams

The following wiring diagrams describe some of the possible configurations for the controller.

12.1. Wiring example #1: basic connection

In the schematic diagram of *Figure 3: example schematic #1* the controller drives a light and a fan (generally mounted in the same light).

As shown, if required for the application, a safety switch may be included in the circuit to disable the controller output in order to protect the end user from photobiological hazard. The same switch can be used to turn the light on and off as needed by the application.

In this example the maximum output current is selected by the 4-way dip switch as 1200 mA. It can then be adjusted from zero to the maximum value using the knob on the front panel.

All these settings can also be modified using the serial RS485 and Ethernet interfaces (see sections [Operation with Modbus](#) and [Operation with a web browser](#)).

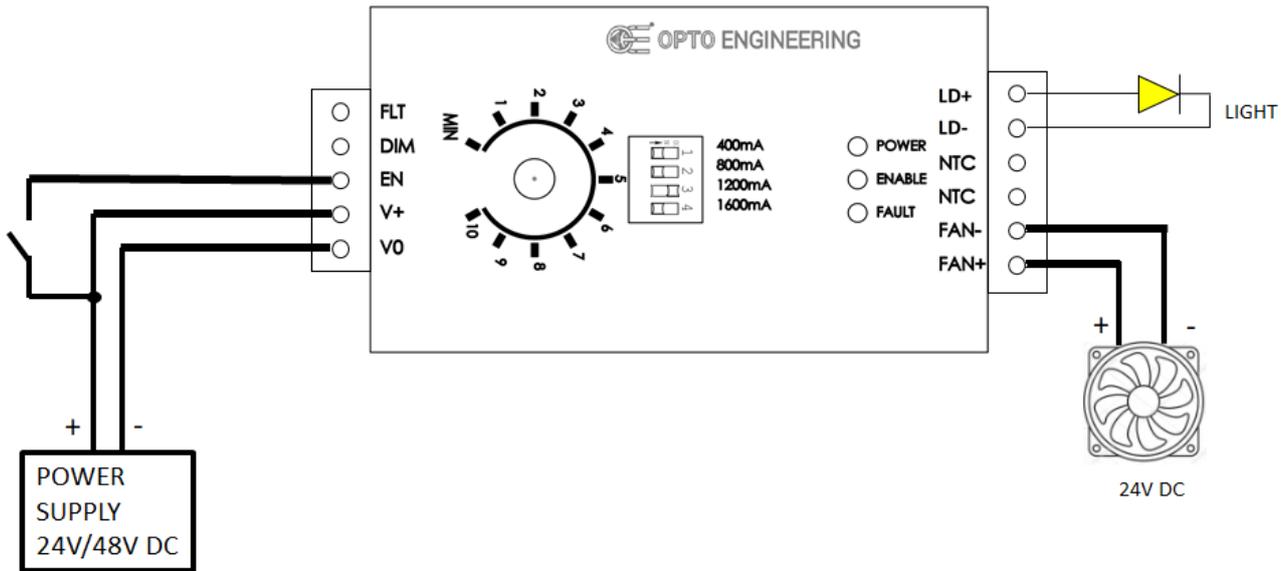


Figure 3: example schematic #1

12.2. Wiring example #2: advanced connection

In the schematic diagram of *Figure 4: example schematic #2* the controller drives a light and a fan while monitoring the light temperature using a thermal sensor. Generally, the thermal sensor is mounted in the light.

As shown, in this case the controller is always enabled (will be automatically disabled only during fault conditions).

In this example the maximum output current selected by the 4-way dip switch is 2800 mA (= 1200 mA + 1600 mA). The intensity of the light can then be adjusted using the depicted external voltage source (possibly variable between 0 V and 10 V) connected to the external dimming input.

All these settings can also be modified using the serial RS485 and Ethernet interfaces (see sections [Operation with Modbus](#) and [Operation with a web browser](#)).

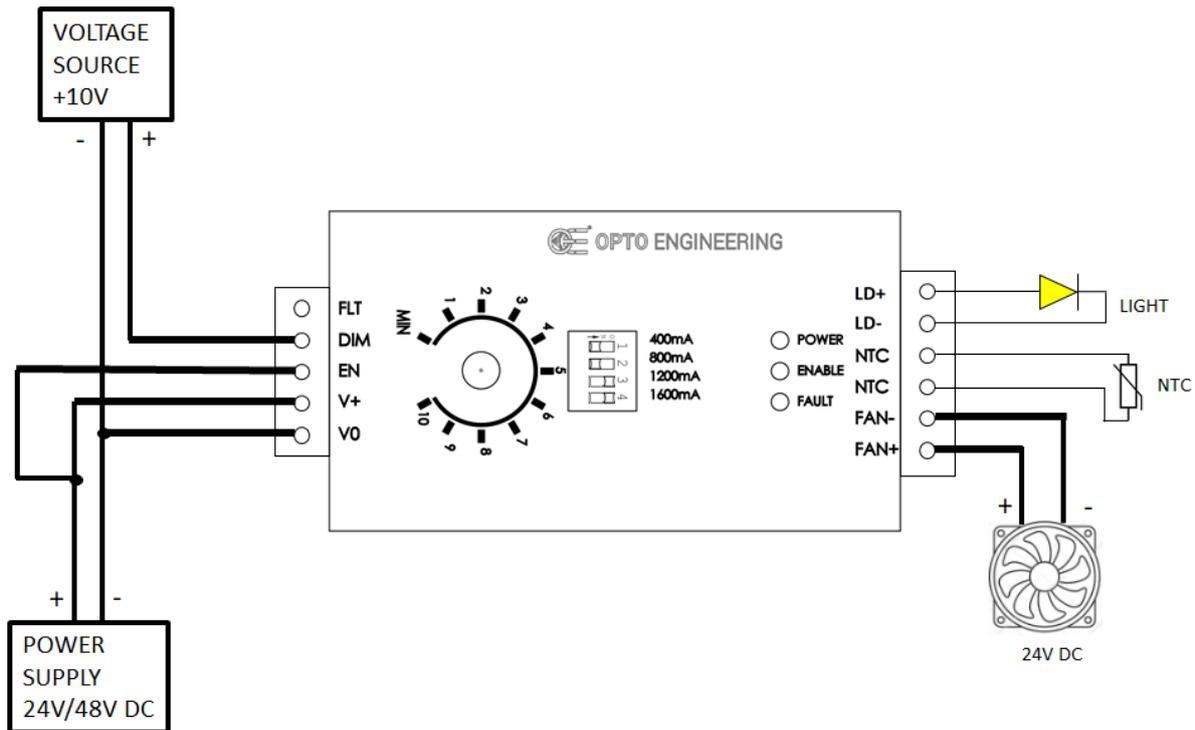


Figure 4: example schematic #2

13. Operation

The controller can be powered with a supply voltage of either 24 V or 48 V. As stated before, intermediate voltages are not allowed and must be avoided. As a general indication, use either 24 V or 48 V if the light power is less than 50 W and use 48 V if the light power is 50 W or more. Use of a 24 V supply with a light power of 50 W or more is highly discouraged and can result in the controller overheating and malfunctioning.

Before powering the controller ensure that all the connections are correct and the current range is adequate for the light (see the light datasheet and manual for details on this topic).

For better results be careful that the selected current range is the lowest range capable of delivering the required output current. Such a selection reduces noise in the electronics and thus flickering in the light is minimized.

There are several ways to configure and use the controller. Some examples follow:

- Locally, by using the 4-way dip switch and the knob on the top panel with the digital enable input tied directly to the supply voltage
- Remotely, by using the 4-way dip switch, the external analogue dimming input and the digital enable input
- Remotely, by using the serial RS485 interface. To support the serial RS485 interface the controller implements a subset of the Modbus/RTU slave protocol
- Remotely, by using the Ethernet interface. Supported Ethernet speeds are 10 Mbit/s and 100 Mbit/s with auto negotiation. The Ethernet interface allows to configure the controller using the Modbus/TCP slave protocol, the Modbus/UDP slave protocol or the HTTP protocol. For supporting the latter, the controller provides an internal web server accessible by most common web browsers

In the next sections, an overview of the operation modes and of the Modbus/RTU, Modbus/TCP and Modbus/UDP protocols is given.

13.1. Operation in continuous mode

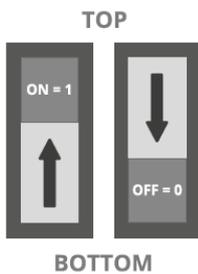
This controller normally operates in continuous mode.

There are four basic output current ranges which can be selected either by using the 4-way dip switch or the serial RS485 and Ethernet interfaces (see sections [Operation with Modbus](#) and [Operation with a web browser](#)). These are:

- 0 – 400 mA
- 0 – 800 mA
- 0 – 1200 mA
- 0 – 1600 mA

These four ranges can be combined together to obtain ten different output current ranges, nominally from zero up to 4 A. All the sixteen possible combination of the 4-way dip switch are listed in *Table 8: dip switch combination and output current ranges*. Note combination “0000” is not allowed and must not be used at any time.

Each of these dip switches is enabled (1 state) when its corresponding mechanical lever is in the ON position (i.e. moved toward the top of the controller), while it is disabled (0 state) when its corresponding mechanical lever is in the OFF position (i.e. moved toward the bottom of the controller).



Dip switch states:
 ON position (1 state): lever moved towards the top
 OFF position (0 state): lever moved towards the bottom

Dip switch				Max output current (mA)
1 (400 mA)	2 (800 mA)	3 (1200 mA)	4 (1600 mA)	
0	0	0	0	Not allowed
1	0	0	0	400
0	1	0	0	800
1	1	0	0	1200
0	0	1	0	
1	0	1	0	1600
0	0	0	1	
1	0	0	1	2000

0	1	1	0	
1	1	1	0	2400
0	1	0	1	
1	1	0	1	2800
0	0	1	1	
1	0	1	1	3200
0	1	1	1	3600
1	1	1	1	4000

Table 8: dip switch combination and output current ranges

For each selected range the current can be linearly adjusted in three different ways. The basic way is to use the knob mounted on the controller front panel. The alternative way is to use the external analogue dimming input (see section [Dimming input](#)). The third way is to use the serial RS485 and Ethernet interfaces (see sections [Operation with Modbus](#) and [Operation with a web browser](#)).

Both the knob and the external analogue dimming input allow to adjust the output current from zero up to the maximum current value selected by the 4-way dip switch. To be able to adjust the output current via the knob, the external analogue dimming input signal must be left floating.

If an external dimming signal is used, the output current can be adjusted by applying an analogue signal (from 0 V up to 10 V) at the dimming input (see section [Dimming input](#)). In this case, it is important that the external driving source has an output resistance lower than 470 Ω in order to override the contribution of the knob.

By using the serial RS485 and Ethernet interfaces (see sections [Operation with Modbus](#) and [Operation with a web browser](#)) it is possible to override both the current range selected by the 4-way dip switch and the current adjustment made with the knob or the external analogue dimming input signal.

13.2. Operation in pulsed mode

Although primarily intended for continuous mode operation, this controller also allows slow speed operation in pulsed mode by sequentially turning the enable input signal on and off.

In case of operation in pulsed mode, it is important to note that the controller is not exactly performing such as a strobe controller. In fact, the response time after a change on the enable input signal is approximately 560 μs for light activation and 15 μs for light deactivation, while the output current rising edge takes approximately 140 μs and the falling edge takes approximately 85 μs. This information is collected in the time diagram of *Figure 5: time diagram of the controller output after a change on the enable input*

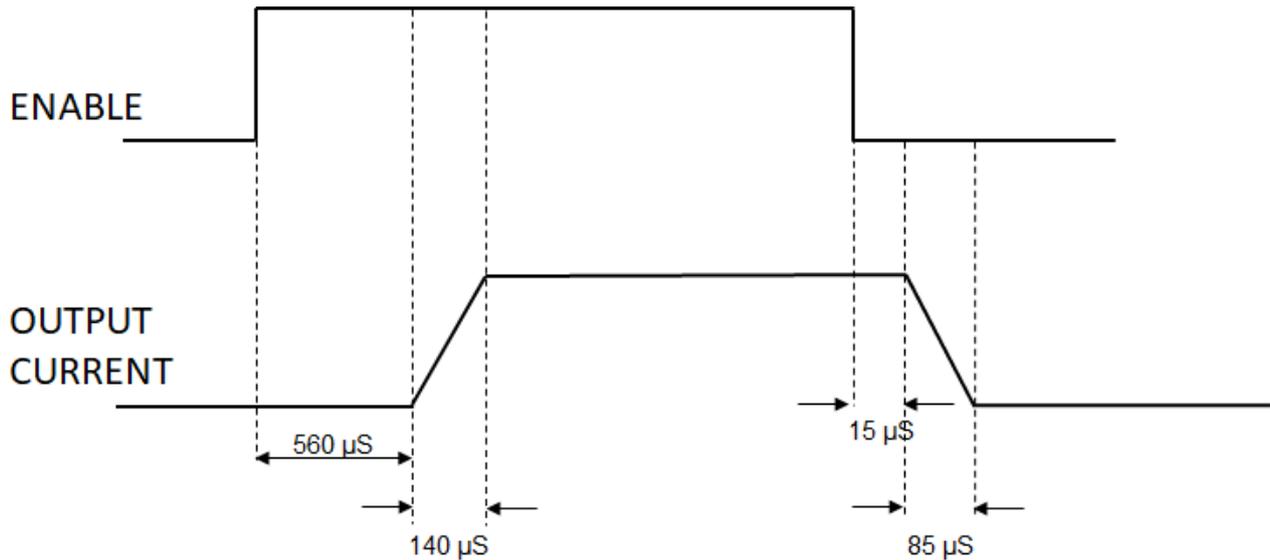


Figure 5: time diagram of the controller output after a change on the enable input

As for the continuous mode, the peak value of the output current pulse can be programmed by using the 4-way dip switch (in order to select the output current range) and the knob, or by using the 4-way dip switch and the external analogue dimming input signal. A third way is by using the serial RS485 and Ethernet interfaces ([see sections Operation with Modbus and Operation with a web browser](#)).

The maximum working frequency for pulse operation must be limited to 15 Hz. Higher working frequencies may cause damage or malfunction of the controller.

13.3. Flickering optimization

The power stage is implemented using a switching converter and the output current has a small degree of superimposed switching noise that may result in the flickering of the light at extremely low exposure times.

- In order to minimize this superimposed switching noise, take care to select the lowest current range capable of delivering the output current required by the application
- The switching noise generally decreases as the output voltage increases. To minimize the switching noise for light voltages around 2 V or less it is advisable to artificially increase the controller output voltage by inserting an extra resistor in series with the output. The required resistance may be easily calculated using the Ohm's law

13.4. Operation with Modbus

The Modbus/RTU, Modbus/TCP and Modbus/UDP protocols are supported by most programmable logic controllers (PLCs) with a suitable communication port. The controller can also be configured by any PC with a proper interface.

Being simple and robust, over the years Modbus became a well-known communication protocol and it is now a commonly available means of connecting industrial electronic devices. The development and update of Modbus protocols has been managed by the Modbus Organization since April 2004. The Modbus Organization is an association of users and suppliers of Modbus compliant devices that seeks to drive the adoption and evolution of Modbus.

The organization web site is:

<http://www.modbus.org>

More information, including Modbus specifications, implementation guides and code fragments can be downloaded from:

<http://www.modbus.org/specs.php>

13.4.1. Comparison of Modbus/RTU, Modbus/TCP and Modbus/UDP

The Modbus/RTU, Modbus/TCP and Modbus/UDP are pretty similar. The main difference is that Modbus/RTU is used on serial lines, while Modbus/TCP and Modbus/UDP are used on Ethernet connections. Modbus/TCP is connection-oriented and is implemented using TCP packets, while Modbus/UDP is connectionless and is implemented using UDP packets.

The controller implements Modbus/RTU with a serial RS485 interface (by default operating at 9600 bits per second, with even parity). The controller implements Modbus/TCP and Modbus/UDP with an Ethernet interface operating at 10 Mbit/s or 100 Mbit/s with auto negotiation.

13.4.2. Supported function codes

Modbus is a request/reply protocol and offers services specified by function codes.

The controller implements a restricted subset of the Modbus slave protocol. The list in *Table 9: function codes supported by the controller* summarizes the Modbus function codes supported by the controller with the current firmware.

Function name	Function code	Note
Read Holding Registers	0x03	
Write Single Register	0x06	
Write Multiple Registers	0x10	

Table 9: function codes supported by the controller

Any Modbus request containing an unimplemented function code is silently ignored by the controller and no response, of any kind, is given back to the master. These supported function codes can be used to access the controller internal register file, organized as an array of sixteen-bit (two bytes) values. These function codes are briefly described in the following sections.

13.4.3. Read Holding Registers (0x03)

This function code is used to read the contents of a contiguous block of registers from the controller register file. The master specifies the starting register address and the number of registers to be read. Registers are addressed starting at zero. The register data in the response message are packed as two bytes per register. For each register, the first byte contains the high order bits and the second contains the low order bits.

13.4.4. Write Single Register (0x06)

This function code is used to write a single register in the controller register file. The master specifies the address of the register to be written and the actual data to be written. Registers are addressed starting at zero. The register data in the request message are packed as two bytes per register. For each register, the first byte contains the high order bits and the second contains the low order bits. The normal response is an echo of the request, returned after the register contents have been

written.

13.4.5. Write Multiple Registers (0x10)

This function code is used to write a block of contiguous registers (1 to 123 registers) in the controller register file. The master specifies the starting register address, the number of registers and the actual data to be written. Registers are addressed starting at zero. The register data in the request message are packed as two bytes per register. For each register, the first byte contains the high order bits and the second contains the low order bits. The normal response returns the function code, starting address, and quantity of registers written.

13.5. Register file

The list in *Table 10: controller register file* summarizes all the Modbus registers implemented in the controller. There are 512 registers, each of sixteen-bits (two bytes) in size. The registers are mapped at contiguous addresses starting at 0 and ending at 511.

Address	Name	Type	Range	Description
0	DEVICE_TYPE	R	0-65535	Device type
1	BOOTLOADER_VERSION	R	0-65535	Bootloader firmware version
2	MCU_VERSION	R	0-65535	Microcontroller firmware version
3	UNUSED	N/A	N/A	
4	BOARD_VERSION	R	0-65535	Board version
5	ENABLE_CONTROL	RW	0-2	Enable control mode
6	CURRENT_CONTROL	RW	0-1	Output current control mode
7	CURRENT_RANGE	RW	0-11	Output current range select
8	CURRENT_PRESET	RW	0-4000	Output current preset value
9	FAULT_CONTROL	RW	0-31	Fault control word
10	FAULT_BOARD_TMP_LOW	RW	-200 to +1000	Low board temperature limit
11	FAULT_BOARD_TMP_HIGH	RW	-200 to +1000	High board temperature limit
12	FAULT_LIGHT_TMP_LOW	RW	-200 to +1000	Low light temperature limit
13	FAULT_LIGHT_TMP_HIGH	RW	-200 to +1000	High light temperature limit
14	FAN_CONTROL	RW	0-2	Fan operation mode
15	FAN_TMP_LOW	RW	-200 to +1000	Low fan temperature limit

16	FAN_TMP_HIGH	RW	-200 to +1000	High fan temperature limit
17	FAN_DUTY_INF	RW	0-100	Duty cycle of the fan power supply for temperature lower than FAN_TMP_LOW
18	FAN_DUTY_SUP	RW	0-100	Duty cycle of the fan power supply for temperature higher than FAN_TMP_HIGH
19	FAN_DUTY_MIN	RW	0-100	Duty cycle of the fan power supply for temperature equal to FAN_TMP_LOW
20	FAN_DUTY_MAX	RW	0-100	Duty cycle of the fan power supply for temperature equal to FAN_TMP_HIGH
21	FAN_DUTY_MANUAL	RW	0-100	Duty cycle of the fan power supply when fan operating in manual mode
22	FAN_PWM_FREQUENCY	RW	100-50000	Frequency of the fan power supply
23	OSC_CONTROL	RW	0-1	Oscillator operation mode
24	OSC_FREQUENCY	RW	2000-7000	Oscillator working frequency
25	UNUSED	N/A	N/A	
26	UNUSED	N/A	N/A	
27	UNUSED	N/A	N/A	
28	UNUSED	N/A	N/A	
29	UNUSED	N/A	N/A	
30	UNUSED	N/A	N/A	
31	UNUSED	N/A	N/A	
32	UNUSED	N/A	N/A	
33	UNUSED	N/A	N/A	
34	UNUSED	N/A	N/A	
35	UNUSED	N/A	N/A	
36	UNUSED	N/A	N/A	
37	UNUSED	N/A	N/A	
38	UNUSED	N/A	N/A	
39	UNUSED	N/A	N/A	
40	UNUSED	N/A	N/A	
41	RS485_MODBUS_ADDR	RW	1-247	Modbus address for serial RS485 interface
42	RS485_LINE_SPEED	RW	0-7	Speed for serial RS485 interface

43	RS485_LINE_PARITY	RW	0-2	Parity for serial RS485 interface
44	ETH_MAC_ADDR0	R	0-65535	Bytes 0 and 1 of Ethernet MAC address
45	ETH_MAC_ADDR1	R	0-65535	Bytes 2 and 3 of Ethernet MAC address
46	ETH_MAC_ADDR2	R	0-65535	Bytes 4 and 5 of Ethernet MAC address
47	ETH_HOSTNAME0	RW	0-65535	Bytes 0 and 1 of Ethernet host name
48	ETH_HOSTNAME1	RW	0-65535	Bytes 2 and 3 of Ethernet host name
49	ETH_HOSTNAME2	RW	0-65535	Bytes 4 and 5 of Ethernet host name
50	ETH_HOSTNAME3	RW	0-65535	Bytes 6 and 7 of Ethernet host name
51	ETH_HOSTNAME4	RW	0-65535	Bytes 8 and 9 of Ethernet host name
52	ETH_HOSTNAME5	RW	0-65535	Bytes 10 and 11 of Ethernet host name
53	ETH_HOSTNAME6	RW	0-65535	Bytes 12 and 13 of Ethernet host name
54	ETH_HOSTNAME7	RW	0-65535	Bytes 14 and 15 of Ethernet host name
55	ETH_DHCP_ENABLE	RW	0-1	Ethernet DHCP enable/disable flag
56	ETH_IP_ADDR_HI	RW	0-65535	High order word of Ethernet IP address
57	ETH_IP_ADDR_LO	RW	0-65535	Low order word of Ethernet IP address
58	ETH_SUBNET_MASK_HI	RW	0-65535	High order word of Ethernet subnet mask
59	ETH_SUBNET_MASK_LO	RW	0-65535	Low order word of Ethernet subnet mask
60	ETH_DEF_GATEWAY_HI	RW	0-65535	High order word of Ethernet default gateway
61	ETH_DEF_GATEWAY_LO	RW	0-65535	Low order word of Ethernet default gateway
62	ETH_PRI_DNS_HI	RW	0-65535	High order word of Ethernet primary DNS address
63	ETH_PRI_DNS_LO	RW	0-65535	Low order word of Ethernet primary DNS address
64	ETH_SEC_DNS_HI	RW	0-65535	High order word of Ethernet secondary DNS address

65	ETH_SEC_DNS_LO	RW	0-65535	Low order word of Ethernet secondary DNS address
66	ETH_MODBUS_ADDR	RW	1-247	Modbus address for Ethernet interface
67	ETH_MODBUS_TCP_PORT	RW	1-65535	Ethernet port for Modbus/TCP
68	ETH_MODBUS_UDP_PORT	RW	1-65535	Ethernet port for Modbus/UDP
69	WEB_PASSWORD0	RW	0-65535	Bytes 0 and 1 of web password
70	WEB_PASSWORD1	RW	0-65535	Bytes 2 and 3 of web password
71	WEB_PASSWORD2	RW	0-65535	Bytes 4 and 5 of web password
72	WEB_PASSWORD3	RW	0-65535	Bytes 6 and 7 of web password
73	BOARD_TEMPERATURE	R	-200 to +1000	Board temperature measured at the internal power driver
74	LIGHT_TEMPERATURE	R	-200 to +1000	Light temperature measured by external thermal sensor
75	SUPPLY_VOLTAGE	R	0-528	Measured controller supply voltage
76	DRIVER_CURRENT	R	0-5000	Measured current of the internal power driver
77	FAN_VOLTAGE	R	0-264	Measured voltage of fan output
78	INTERNAL_VOLTAGE	R	0-132	Measured voltage of internal analogue power supply
79	LIGHT_VOLTAGE	R	0-480	Measured voltage of light output
80	LIGHT_CURRENT	R	0-4000	Measured current of light output
81	DIMMING_VOLTAGE	R	0-100	Measured voltage of the external analogue dimming input signal
82	DIPSWITCH_SETTING	R	0-15	Detected setting of 4-way dip switch
83	ENABLE_INPUT	R	0-1	State of the digital enable input signal
84	FAULT_FLAG	R	0-31	Detected fault conditions
85	FAULT_OUTPUT	R	0-1	State of the digital fault output signal
86	UNUSED	N/A	N/A	
87	UNUSED	N/A	N/A	
88	UNUSED	N/A	N/A	
89	FAN_DUTY	R	0-100	Actual duty cycle of the fan power supply
90	ERROR_WORD	R	0-	Composite error word

			65535	
91-126	SPARE[0:34]	N/A	N/A	Reserved for future use
127	BOARD_COMMAND	RW	0-3	Board command

Table 10: controller register file

As indicated in the table, most of the registers can be both read and written (type RW), some registers are read only (type R) and some others are unused or reserved (type N/A). Registers marked as unused or reserved must not be accessed, either in reading and writing, at any time. Failure to comply with this requirement may lead to device malfunction. The accessible registers are described in the following sections.

Changes to the register file are not saved in non-volatile memory until a specific command is issued to the controller. See the description of register **BOARD_COMMAND** in the next sections for more information on this subject.

In the following sections, the 0x prefix is used to denote a hexadecimal number. The prefix 0x is used in C and related languages.

13.5.1. Register DEVICE_TYPE

This register contains the device type. This information is encoded as a 16-bit unsigned number. For the standard LTIC1CH-D1-4 the device type is 0x0009.

13.5.2. Register BOOT_VERSION

This register contains the bootloader firmware version for the microcontroller. This information is encoded as a 16-bit unsigned number.

13.5.3. Register MCU_VERSION

This register contains the application firmware version for the microcontroller. This information is encoded as a 16-bit unsigned number.

13.5.4. Register ENABLE_CONTROL

Bit field [1:0] of this register selects how the controller output can be enabled and disabled.

Allowed values are in the range from 0 to 2 and are listed below. Avoid operation with non-listed values.

- When 0 the controller output is enabled and disabled according to the status of the enable input signal
- When 1 the controller output is permanently disabled
- When 2 the controller output is permanently enabled

Bit field [15:2] of this register is unused. When writing these bits, they must be set to zero.

The default value is 0, corresponding to the operation of the controller based on the status of the enable input signal. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.5. Register CURRENT_CONTROL

Bit 0 of this register selects how the controller output current can be adjusted.

Allowed values are in the range from 0 to 1 and are listed below. Avoid operation with non-listed

values.

- When set to 0 the output current can be adjusted with the knob on the top panel or the external analogue dimming input signal (see [section 13.1](#) for more information)
- When set to 1 the output current is fixed to the user preset output current value stored in Modbus register **CURRENT_PRESET**

Bit field [15:1] of this register is unused. When writing these bits, they must be set to zero.

The default value is 0, corresponding to the output current adjustment carried out by the knob on the top panel or the external analogue dimming input signal. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.6. Register CURRENT_RANGE

Bit field [3:0] of this register selects the range of the controller output current.

Allowed values are in the range from 0 to 11 and are listed below. Avoid operation with non-listed values.

- When 0 the output current range is selected through the 4-way dip switch on the top panel (see [section 13.1](#) for more information)
- When 1 the output current range is 400 mA
- When 2 the output current range is 800 mA
- When 3 the output current range is 1200 mA
- When 4 the output current range is 1600 mA
- When 5 the output current range is 2000 mA
- When 6 the output current range is 2400 mA
- When 7 the output current range is 2800 mA
- When 8 the output current range is 3200 mA
- When 9 the output current range is 3600 mA
- When 10 the output current range is 4000 mA
- When 11 the output current range is automatically selected by the controller according to the user preset output current value stored in Modbus register **CURRENT_PRESET**

Bit field [15:4] of this register is unused. When writing these bits, they must be set to zero.

The default value is 0, corresponding to the output current range selected by the 4-way dip switch. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.7. Register CURRENT_PRESET

Bit field [11:0] of this register defines the preset value for the controller output current.

Allowed values are in the range from zero to 4000 and are expressed in mA, so the required current can range from zero up to 4000 mA with a resolution of 1 mA. Avoid operation with non-allowed values.

Bit field [15:12] of this register is unused. When writing these bits, they must be set to zero.

The default value is 0, corresponding to a preset current value of zero. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.8. Register FAULT_CONTROL

The controller has a fault output that is used to signal overheating or malfunction. The fault output is internally pulled up to the supply voltage (either 24 V or 48 V) and is active low.

There are five possible fault sources. By default, some of these fault sources are monitored by the controller and others are ignored. Any of the five fault sources can be selectively enabled or ignored.

If fault condition is detected the controller output is turned off, the fault output is driven low and the FAULT red LED on the top panel is continuously lit. The controller will automatically resume regular operation once the fault condition is removed.

Bit field [4:0] of this register selects which fault sources are monitored and which fault sources are ignored. The meaning of each of the bits is listed below.

- When bit 0 is set to 1 the controller generates a fault output if an out-of-range supply voltage is detected. A valid supply voltage is either 24 V \pm 10% or 48 V \pm 10%
- When bit 1 is set to 1 the controller generates a fault output if board overtemperature is detected
- When bit 2 is set to 1 the controller generates a fault output if light overtemperature is detected
- When bit 3 is set to 1 the controller generates a fault output if a short circuit on the light output is detected
- When bit 4 is set to 1 the controller generates a fault output if an open circuit on the light output is detected

Bit field [15:5] of this register is unused. When writing these bits, they must be set to zero.

The default value is 0x7, corresponding to the monitoring of the supply voltage, the temperature of the board, the temperature of the light and to the ignoring of a short circuit and an open circuit on the light output. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.9. Register FAULT_BOARD_TMP_LOW

The controller is able to monitor the temperature of the internal board and to generate a fault output in case an overtemperature is detected. To reduce the frequency of activation of the fault output and improve immunity to small temperature changes a hysteretic overtemperature detection is implemented.

This register sets the lower temperature limit for the hysteretic temperature comparison. Given a temperature, the corresponding register value is expressed as a signed sixteen bits integer and can be calculated as follow:

$$FAULT_BOARD_TMP_LOW = signedInteger(BoardTemperatureLow * 10)$$

Where *BoardTemperatureLow* is the required temperature expressed in °C.

Allowed values for the register are in the range from -200 to +1000 and are expressed in tenths of °C, so the required temperature can range from -20.0 °C up to +100.0 °C with a resolution of 0.1 °C. Avoid operation with non-allowed values.

The default value is 700, corresponding to a temperature of 70 °C. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.10. Register FAULT_BOARD_TMP_HIGH

The controller is able to monitor the temperature of the internal board and to generate a fault output

in case an overtemperature is detected. To reduce the frequency of activation of the fault output and improve immunity to small temperature changes a hysteretic overtemperature detection is implemented.

This register sets the upper temperature limit for the hysteretic temperature comparison. Given a temperature, the corresponding register value is expressed as a signed sixteen bits integer and can be calculated as follow:

$$\mathbf{FAULT_BOARD_TMP_HIGH} = \mathit{signedInteger}(\mathit{BoardTemperatureHigh} * 10)$$

Where *BoardTemperatureHigh* is the required temperature expressed in °C.

Allowed values for the register are in the range from -200 to +1000 and are expressed in tenths of °C, so the required temperature can range from -20.0 °C up to +100.0 °C with a resolution of 0.1 °C. Avoid operation with non-allowed values.

The default value is 900, corresponding to a temperature of 90 °C. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.11. Register **FAULT_LIGHT_TMP_LOW**

The controller is able to monitor the temperature of the connected light and to generate a fault output in case an overtemperature is detected. To reduce the frequency of activation of the fault output and improve immunity to small temperature changes a hysteretic overtemperature detection is implemented.

This register sets the lower temperature limit for the hysteretic temperature comparison. Given a temperature, the corresponding register value is expressed as a signed sixteen bits integer and can be calculated as follow:

$$\mathbf{FAULT_LIGHT_TMP_LOW} = \mathit{signedInteger}(\mathit{LightTemperatureLow} * 10)$$

Where *LightTemperatureLow* is the required temperature expressed in °C.

Allowed values for the register are in the range from -200 to +1000 and are expressed in tenths of °C, so the required temperature can range from -20.0 °C up to +100.0 °C with a resolution of 0.1 °C. Avoid operation with non-allowed values.

The default value is 700, corresponding to a temperature of 70 °C. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.12. Register **FAULT_LIGHT_TMP_HIGH**

The controller is able to monitor the temperature of the connected light and to generate a fault output in case an overtemperature is detected. To reduce the frequency of activation of the fault output and improve immunity to small temperature changes a hysteretic overtemperature detection is implemented.

This register sets the upper temperature limit for the hysteretic temperature comparison. Given a temperature, the corresponding register value is expressed as a signed sixteen bits integer and can be calculated as follow:

$$\mathbf{FAULT_LIGHT_TMP_HIGH} = \mathit{signedInteger}(\mathit{LightTemperatureHigh} * 10)$$

Where *LightTemperatureHigh* is the required temperature expressed in °C.

Allowed values for the register are in the range from -200 to +1000 and are expressed in tenths of °C, so the required temperature can range from -20.0 °C up to +100.0 °C with a resolution of 0.1 °C. Avoid operation with non-allowed values.

The default value is 900, corresponding to a temperature of 90 °C. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.13. Register FAN_CONTROL

Bit field [1:0] of this register selects the fan operation mode. Allowed values are in the range from 0 to 2 and are listed below (see [section 8.3.3](#) for more information). Avoid operation with non-listed values.

- When 0 the proportional control algorithm is selected
- When 1 the hysteretic control algorithm is selected
- When 2 the manual control is selected

Bit field [15:2] of this register is unused. When writing these bits, they must be set to zero.

The default value is 0, corresponding to the selection of proportional control algorithm. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.14. Register FAN_TMP_LOW

This register sets the *TLow* parameter described in [section 8.3.3](#), used for the control of the fan mounted in the light (if available). Given a temperature, the corresponding register value is expressed as a signed sixteen bits integer and can be calculated as follow:

$$FAN_TMP_LOW = signedInteger(TLow * 10)$$

Where *TLow* is the required temperature expressed in °C.

Allowed values for the register are in the range from -200 to +1000 and are expressed in tenths of °C, so the required temperature can range from -20.0 °C up to +100.0 °C with a resolution of 0.1 °C. Avoid operation with non-allowed values.

The default value is 700, corresponding to a temperature of 70 °C. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.15. Register FAN_TMP_HIGH

This register sets the *THigh* parameter described in [section 8.3.3](#), used for the control of the fan mounted in the light (if available). Given a temperature, the corresponding register value is expressed as a signed sixteen bits integer and can be calculated as follow:

$$FAN_TMP_HIGH = signedInteger(THigh * 10)$$

Where *THigh* is the required temperature expressed in °C.

Allowed values for the register are in the range from -200 to +1000 and are expressed in tenths of °C, so the required temperature can range from -20.0 °C up to +100.0 °C with a resolution of 0.1 °C. Avoid operation with non-allowed values.

The default value is 850, corresponding to a temperature of 85 °C. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.16. Register FAN_DUTY_INF

This register sets the *DutyInf* parameter described in [section 8.3.3](#), used for the control of the fan mounted in the light (if available). Given a duty cycle, the corresponding register value is expressed as an unsigned sixteen bits integer and can be calculated as follow:

$$FAN_DUTY_INF = unsignedInteger(DutyInf)$$

Where *DutyInf* is the required duty cycle expressed in %.

Allowed values for the register are in the range from 0 to 100 and are expressed in units of %, so

the required duty cycle can range from 0 % up to 100 % with a resolution of 1 %. Avoid operation with non-allowed values.

The default value is 50, corresponding to a duty cycle of 50 %. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.17. Register FAN_DUTY_SUP

This register sets the *DutySup* parameter described in [section 8.3.3](#), used for the control of the fan mounted in the light (if available). Given a duty cycle, the corresponding register value is expressed as an unsigned sixteen bits integer and can be calculated as follow:

$$FAN_DUTY_SUP = \text{unsignedInteger}(DutySup)$$

Where *DutySup* is the required duty cycle expressed in %.

Allowed values for the register are in the range from 0 to 100 and are expressed in units of %, so the required duty cycle can range from 0 % up to 100 % with a resolution of 1 %. Avoid operation with non-allowed values.

The default value is 100, corresponding to a duty cycle of 100 %. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.18. Register FAN_DUTY_MIN

This register sets the *DutyMin* parameter described in [section 8.3.3](#), used for the control of the fan mounted in the light (if available). Given a duty cycle, the corresponding register value is expressed as an unsigned sixteen bits integer and can be calculated as follow:

$$FAN_DUTY_MIN = \text{unsignedInteger}(DutyMin)$$

Where *DutyMin* is the required duty cycle expressed in %.

Allowed values for the register are in the range from 0 to 100 and are expressed in units of %, so the required duty cycle can range from 0 % up to 100 % with a resolution of 1 %. Avoid operation with non-allowed values.

The default value is 50, corresponding to a duty cycle of 50 %. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.19. Register FAN_DUTY_MAX

This register sets the *DutyMax* parameter described in [section 8.3.3](#), used for the control of the fan mounted in the light (if available). Given a duty cycle, the corresponding register value is expressed as an unsigned sixteen bits integer and can be calculated as follow:

$$FAN_DUTY_MAX = \text{unsignedInteger}(DutyMax)$$

Where *DutyMax* is the required duty cycle expressed in %.

Allowed values for the register are in the range from 0 to 100 and are expressed in units of %, so the required duty cycle can range from 0 % up to 100 % with a resolution of 1 %. Avoid operation with non-allowed values.

The default value is 100, corresponding to a duty cycle of 100 %. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.20. Register FAN_DUTY_MANUAL

This register sets the manual duty cycle described in [section 8.3.3](#), used for the control of the fan mounted in the light (if available). Given a duty cycle *DutyManual*, the corresponding register value is expressed as an unsigned sixteen bits integer and can be calculated as follow:

$$FAN_DUTY_MANUAL = \text{unsignedInteger}(DutyManual)$$

Where *DutyManual* is the required duty cycle expressed in %.

Allowed values for the register are in the range from 0 to 100 and are expressed in units of %, so the required duty cycle can range from 0 % up to 100 % with a resolution of 1 %. Avoid operation with non-allowed values.

The default value is 50, corresponding to a duty cycle of 50 %. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.21. Register FAN_PWM_FREQUENCY

This register sets the frequency of the fan power supply. Allowed values are in the range from 100 to 50000, corresponding to a frequency range from 100 Hz up to 50 kHz. Avoid operation with non-allowed values.

The default value is 250, corresponding to a frequency of 250 Hz. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.22. Register OSC_CONTROL

Bit 0 of this register controls the operation of the internal oscillator driving the controller power stage.

Allowed values are in the range from 0 to 1 and are listed below. Avoid operation with non-listed values.

- When set to 0 the internal oscillator is free-running at a frequency of approximately 300 kHz
- When set to 1 the internal oscillator is tuned to a specified frequency. Available tuning frequencies are in the range from 200 kHz up to 700 kHz

Bit field [15:1] of this register is unused. When writing these bits, they must be set to zero.

The default value is 0, corresponding to free-running operation. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.23. Register OSC_FREQUENCY

This register sets the tuning frequency for the internal oscillator driving the controller power stage.

Given a frequency *OscFrequency*, the corresponding register value is expressed as an unsigned sixteen bits integer and can be calculated as follow:

$$OSC_FREQUENCY = \text{unsignedInteger}(OscFrequency / 100)$$

Where *OscFrequency* is the required frequency expressed in Hz.

Allowed values for the register are in the range from 2000 to 70000 and are expressed in hundreds of Hz, so the required frequency can range from 200 kHz up to 700 kHz with a resolution of 100 Hz. Avoid operation with non-allowed values.

The default value is 2000, corresponding to a tuning frequency of 200 kHz. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.24. Register RS485_MODBUS_ADDR

This register contains the Modbus address of the controller for the serial RS485 interface. The default value is 32. Permitted values are in the interval between 1 and 247. Avoid operation with non-allowed values.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.25. Register RS485_LINE_SPEED

Bit field [2:0] of this register holds the speed setting for the serial RS485 interface.

Allowed values are in the range from 0 to 7 and are listed below. Avoid operation with non-listed values.

- When 0 the selected baud rate is 1200 bits per second
- When 1 the selected baud rate is 2800 bits per second
- When 2 the selected baud rate is 4800 bits per second
- When 3 the selected baud rate is 9600 bits per second (default value)
- When 4 the selected baud rate is 19200 bits per second
- When 5 the selected baud rate is 38400 bits per second
- When 6 the selected baud rate is 57600 bits per second
- When 7 the selected baud rate is 115200 bits per second

Bit field [15:3] of this register is unused. When writing these bits, they must be set to zero.

The default value is 3, corresponding to 9600 bits per second. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

Note the firmware supports only 9600 bits per second. Setting a different rate in this register will cause the serial interface to operate at 9600 bits per second.

13.5.26. Register RS485_LINE_PARITY

Bit field [1:0] of this register holds the parity setting for the serial RS485 interface.

Allowed values are in the range from 0 to 2 and are listed below. Avoid operation with non-listed values.

- When 0 there is no parity bit
- When 1 the parity bit is even (default value)
- When 2 the parity bit is odd

Bit field [15:2] of this register is unused. When writing these bits, they must be set to zero.

The default value is 1, corresponding to even parity bit. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

Note the firmware supports only even parity. Setting a different option in this register will cause the serial interface to operate with even parity.

13.5.27. Registers ETH_MAC_ADDR[0:2]

These registers contain the Ethernet MAC address of the controller.

- **ETH_MAC_ADDR0**: bytes 0 and 1 of the Ethernet MAC address
- **ETH_MAC_ADDR1**: bytes 2 and 3 of the Ethernet MAC address
- **ETH_MAC_ADDR2**: bytes 4 and 5 of the Ethernet MAC address

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.28. Registers ETH_HOSTNAME[0-7]

These registers contain the Ethernet host name of the controller.

- **ETH_HOSTNAME0**: bytes 0 and 1 of the Ethernet host name
- **ETH_HOSTNAME1**: bytes 2 and 3 of the Ethernet host name
- **ETH_HOSTNAME2**: bytes 4 and 5 of the Ethernet host name
- **ETH_HOSTNAME3**: bytes 6 and 7 of the Ethernet host name
- **ETH_HOSTNAME4**: bytes 8 and 9 of the Ethernet host name
- **ETH_HOSTNAME5**: bytes 10 and 11 of the Ethernet host name
- **ETH_HOSTNAME6**: bytes 12 and 13 of the Ethernet host name
- **ETH_HOSTNAME7**: bytes 14 and 15 of the Ethernet host name

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.29. Register ETH_DHCP_ENABLE

Bit 0 of these registers holds the enable flag for the DHCP.

- When set to 0 the DHCP is disabled (default value)
- When set to 1 the DHCP is enabled

Bit field [15:1] of this register is unused. When writing these bits, they must be set to zero.

13.5.30. Register ETH_IP_ADDR_HI

This register contains the sixteen high order bits of the IP address for Ethernet communication of the controller. These are the bits [31:16]. The default IP address is 192.168.0.32.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.31. Register ETH_IP_ADDR_LO

This register contains the sixteen low order bits of the IP address for Ethernet communication of the controller. These are the bits [15:0]. The default IP address is 192.168.0.32.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.32. Register ETH_SUBNET_MASK_HI

This register contains the sixteen high order bits of the subnet mask for Ethernet communication of the controller. These are the bits [31:16]. The default subnet mask is 255.255.255.0.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.33. Register ETH_SUBNET_MASK_LO

This register contains the sixteen low order bits of the subnet mask for Ethernet communication of the controller. These are the bits [15:0]. The default subnet mask is 255.255.255.0.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.34. Register ETH_DEF_GATEWAY_HI

This register contains the sixteen high order bits of the default gateway for Ethernet communication of the controller. These are the bits [31:16]. The default gateway is 192.168.0.1.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.35. Register ETH_DEF_GATEWAY_LO

This register contains the sixteen low order bits of the default gateway for Ethernet communication of the controller. These are the bits [15:0]. The default gateway is 192.168.0.1.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.36. Register ETH_PRI_DNS_HI

This register contains the sixteen high order bits of the primary (preferred) DNS address for Ethernet communication of the controller. These are the bits [31:16]. The primary DNS address is 192.168.0.2.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.37. Register ETH_PRI_DNS_LO

This register contains the sixteen low order bits of the primary (preferred) DNS address for Ethernet communication of the controller. These are the bits [15:0]. The primary DNS address is 192.168.0.2.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.38. Register ETH_SEC_DNS_HI

This register contains the sixteen high order bits of the secondary (alternate) DNS address for Ethernet communication of the controller. These are the bits [31:16]. The secondary DNS address is 192.168.0.2.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.39. Register ETH_SEC_DNS_LO

This register contains the sixteen low order bits of the secondary (alternate) DNS address for Ethernet communication of the controller. These are the bits [15:0]. The secondary DNS address is 192.168.0.2.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.40. Register ETH_MODBUS_ADDR

This register contains the Modbus address of the controller for the Ethernet interface. The default value is 32. Permitted values are in the interval between 1 and 247. Avoid operation with non-allowed values.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.41. Register ETH_MODBUS_TCP_PORT

This register contains the TCP port number for Modbus/TCP communication of the controller. The controller integrates a TCP server that accepts and produces TCP packets containing structured datagrams according to the Modbus/TCP protocol. The default TCP port is 502.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.42. Register ETH_MODBUS_UDP_PORT

This register contains the UDP port number for Modbus/UDP communication of the controller. The controller integrates a UDP server that accepts and produces UDP packets containing structured datagrams according to the Modbus/UDP protocol. The default UDP port is 502.

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.43. Registers WEB_PASSWORD[0-3]

These registers contain the web password of the controller.

- **WEB_PASSWORD0**: bytes 0 and 1 of the web password
- **WEB_PASSWORD1**: bytes 2 and 3 of the web password
- **WEB_PASSWORD2**: bytes 4 and 5 of the web password
- **WEB_PASSWORD3**: bytes 6 and 7 of the web password

The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

13.5.44. Register BOARD_TEMPERATURE

The controller integrates a temperature sensor placed near the internal power driver. This register contains the measured temperature, expressed as a signed sixteen bits integer, with a resolution of 0.1 °C. Given the register value, the corresponding actual temperature can be calculated as follows:

$$BoardTemperature [^{\circ}C] = value(BOARD_TEMPERATURE) * 0.1$$

This register is continuously updated.

13.5.45. Register LIGHT_TEMPERATURE

The controller allows for the connection of an external temperature sensor to measure the light temperature (see [section 8.3.2](#) for more information).

This register contains the measured temperature, expressed as a signed sixteen bits integer, with a resolution of 0.1 °C. Given the register value, the corresponding actual temperature for the remote sensor in the light can be calculated as follows:

$$LightTemperature [^{\circ}C] = value(LIGHT_TEMPERATURE) * 0.1$$

In case the external temperature sensor is not connected, the value in the register is undefined. This register is continuously updated.

13.5.46. Register SUPPLY_VOLTAGE

The controller is capable of measuring its actual supply voltage.

This register contains the measured supply voltage, expressed as an unsigned sixteen bits integer, with a resolution of 0.1 V. Given the register value, the corresponding actual supply voltage can be

calculated as follows:

$$\text{SupplyVoltage [V]} = \text{value}(\text{SUPPLY_VOLTAGE}) * 0.1$$

This register is continuously updated. The value is approximate.

13.5.47. Register DRIVER_CURRENT

The controller is capable of measuring the actual current drawn by the internal power driver.

This register contains the measured current, expressed as an unsigned sixteen bits integer, with a resolution of 1 mA. Given the register value, the corresponding actual supply current can be calculated as follows:

$$\text{DriverCurrent [mA]} = \text{value}(\text{DRIVER_CURRENT}) * 0.1$$

This register is continuously updated. The value is approximate.

13.5.48. Register FAN_VOLTAGE

The controller allows for the connection of an auxiliary external cooling fan (see [section 8.3.3](#) for more information) and is capable of measuring its actual supply voltage.

This register contains the measured fan voltage, expressed as an unsigned sixteen bits integer, with a resolution of 0.1 V. Given the register value, the corresponding actual fan voltage can be calculated as follows:

$$\text{FanVoltage [V]} = \text{value}(\text{FAN_VOLTAGE}) * 0.1$$

This register is continuously updated. The value is approximate.

13.5.49. Register INTERNAL_VOLTAGE

The controller is capable of measuring the actual voltage of the internal analogue power supply.

This register contains the measured internal voltage, expressed as an unsigned sixteen bits integer, with a resolution of 0.1 V. Given the register value, the corresponding actual voltage for the internal logic can be calculated as follows:

$$\text{InternalVoltage [V]} = \text{value}(\text{INTERNAL_VOLTAGE}) * 0.1$$

This register is continuously updated. The value is approximate.

13.5.50. Register LIGHT_VOLTAGE

The controller is capable of measuring the actual voltage of the light output.

This register contains the measured output voltage, expressed as an unsigned sixteen bits integer, with a resolution of 0.1 V. Given the register value, the corresponding actual voltage for the light output can be calculated as follows:

$$\text{LightVoltage [V]} = \text{value}(\text{LIGHT_VOLTAGE}) * 0.1$$

This register is continuously updated. The value is approximate.

13.5.51. Register LIGHT_CURRENT

The controller is capable of measuring the actual current of the light output.

This register contains the measured output current, expressed as an unsigned sixteen bits integer, with a resolution of 0.1 A. Given the register value, the corresponding actual current for the light output can be calculated as follows:

$$\text{LightCurrent [A]} = \text{value}(\text{LIGHT_CURRENT}) * 0.1$$

This register is continuously updated. The value is approximate.

13.5.52. Register DIMMING_VOLTAGE

The controller is capable of measuring the actual voltage of the external analogue dimming input.

This register contains the measured dimming voltage, expressed as an unsigned sixteen bits integer, with a resolution of 0.1 V. Given the register value, the corresponding actual dimming voltage can be calculated as follows:

$$\text{DimmingVoltage [V]} = \text{value}(\text{DIMMING_VOLTAGE}) * 0.1$$

This register is continuously updated. The value is approximate.

13.5.53. Register DIPSWITCH_SETTING

Bit field [3:0] of this register contains the actual setting of the 4-way dip switch.

- Bit 0 is 1 when the 400 mA switch is set to on, it is 0 when the switch is set to off
- Bit 1 is 1 when the 800 mA switch is set to on, it is 0 when the switch is set to off
- Bit 2 is 1 when the 1200 mA switch is set to on, it is 0 when the switch is set to off
- Bit 3 is 1 when the 1600 mA switch is set to on, it is 0 when the switch is set to off

These four ranges can be combined together to obtain ten different output current ranges, from zero up to 4000 mA (see [section 13.1](#) for more information).

Bit field [15:5] of this register is unused. This register is continuously updated.

13.5.54. Register ENABLE_INPUT

Bit 0 of this register contains the state of the digital enable input signal.

- Bit 0 is 1 when the digital enable input signal is detected
- Bit 0 is 0 when the digital enable input signal is not detected

Bit field [15:1] of this register is unused. This register is continuously updated.

13.5.55. Register FAULT_FLAG

Bit field [4:0] of this register contains information about the currently detected fault conditions (see [section 8.2.4](#) for more information).

- Bit 0 is 1 when the supply voltage is out-of-range. A valid supply voltage is either 24 V \pm 10% or 48 V \pm 10%
- Bit 1 is 1 when a board overtemperature is detected
- Bit 2 is 1 when a light overtemperature is detected
- Bit 3 is 1 when a short circuit on the light output is detected
- Bit 4 is 1 when an open circuit on the light output is detected

Bit field [15:5] of this register is unused. This register is continuously updated.

13.5.56. Register FAULT_OUTPUT

Bit 0 of this register contains the state of the digital fault output (see [section 8.2.4](#) for more information).

- Bit 0 is 1 when the digital fault output is driven

- Bit 0 is 0 when the digital fault output is not driven

Bit field [15:1] of this register is unused. This register is continuously updated.

13.5.57. Register FAN_DUTY

This register contains the actual duty cycle of the fan power supply (see [section 8.3.3](#) for more information).

The fan duty cycle is expressed as an unsigned sixteen bits integer, with a resolution of 1 %. Given the register value, the corresponding actual fan duty cycle can be calculated as follows:

$$\text{FanDuty [\%]} = \text{value}(\text{FAN_DUTY})$$

This register is continuously updated.

13.5.58. Register ERROR_WORD

This register contains several error flags in its lower bits. The register may be read and these flags then checked to evaluate the controller health. The meaning of each of the bits is as follows.

- Bit 0 is 1 if controller power supply voltage is out-of-range
- Bit 1 is 1 if a board overtemperature is detected
- Bit 2 is 1 if a light overtemperature is detected
- Bit 3 is 1 if a short circuit on the light output is detected
- Bit 4 is 1 if an open circuit on the light output is detected

When the controller power supply voltage is out-of-range (either $24\text{ V} \pm 10\%$ or $48\text{ V} \pm 10\%$) the relevant error bit is set (bit 0) and the output driver is shut down. The output driver is then reactivated when the power supply voltage is within range.

When the thermal sensor on the board reports an excessive temperature (above $90\text{ }^{\circ}\text{C}$), the relevant error bit is set (bit 1) and the output driver is shut down. The output driver is then reactivated when the temperature drops sufficiently (below $70\text{ }^{\circ}\text{C}$).

When the thermal sensor in the light reports an excessive temperature (above $90\text{ }^{\circ}\text{C}$), the relevant error bit is set (bit 2) and the output driver is shut down. The output driver is then reactivated when the temperature drops sufficiently (below $70\text{ }^{\circ}\text{C}$).

When a hardware problem in the light is detected (i.e. a short circuit or an open circuit) the relevant error bit is set (bit 3 or 4) and the output driver is shut down. The output driver is then reactivated when the hardware problem is solved.

Bit field [15:5] of this register is unused. This register is continuously updated.

13.5.59. Register BOARD_COMMAND

This register is used to execute special actions. Allowed values are listed below. Avoid operation with non-listed values.

- When 0 no action to perform
- When 1 register file is read from non-volatile memory (current contents will be overridden)
- When 2 register file is written to non-volatile memory (current contents will not be affected)
- When 3 the controller is rebooted

Changes to the register file are saved in non-volatile memory only when a specific command is issued to the controller using this register. The register is automatically set back to zero after the requested action has been completed.

13.6. Operation with a web browser

The controller has an Ethernet interface that allows it to be easily configured using a web browser.

The default configuration uses the static IP address 192.168.0.32. It is possible to change this factory configuration using either the serial RS485 interface with Modbus/RTU or the Ethernet interface with Modbus/TCP, Modbus/UDP or the web pages. The factory settings can be restored using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

For more information about the Modbus/RTU, Modbus/TCP and Modbus/UDP protocols and the Modbus register file please refer to [chapter 13.3](#) and [chapter 13.4](#).

To access the web pages of the controller it is necessary to connect the controller to a PC using a standard Ethernet cable and to configure the PC to operate in the same local network used for the controller. The web interface can then easily be accessed by entering the controller IP address in the browser (factory IP address is 192.168.0.32). Microsoft Edge, Google Chrome and Mozilla Firefox are all supported web browsers. Other web clients have not been tested.

The following paragraphs detail the web interface of the controller.

13.6.1. Main page and navigation menu

Once the web browser is successfully connected to the controller, the main web page of the controller will be displayed (see the image in *Figure 6: the **Main** web page*).



Main page

Version information



Device type:	LTIC1CH-D1-4
Bootloader firmware version:	1.04
MCU firmware version:	1.00
FPGA firmware version:	N/A
Board version:	1.00

Current state

Board temperature:	25.8 °C	Light temperature:	N/A °C
Supply voltage:	23.8 V	Driver current:	0.00 A
Fan voltage:	24.1 V	Internal analog voltage:	12.0 V
Light voltage:	0.0 V	Light current:	0.00 A
Dimming voltage:	0.0 V	Enable input signal:	No
Fault flags:	0-0-0-0	Fault output signal:	No
Dipswitch setting:	1-1-1-1	Fan dutycycle:	10 %



Figure 6: the **Main** web page

The top part of the main web page contains the navigation menu that is used to access all the other pages necessary to configure and manage the controller. The navigation menu is always visible and accessible at the top of every page.

From the navigation menu, it is possible to switch to the following pages:

- **Main page**
- **Controller setup**
- **General setup**
- **Advanced setup**

In the bottom part of the main page, the following information is collected:

- **Version information:**
 - **Device type:** model of the controller
 - **Bootloader firmware version:** version of the bootloader firmware

- **MCU firmware version:** version of the application firmware
- **FPGA firmware version:** version of the FPGA firmware
- **Board version:** version of the hardware board
- **Current state:**
 - **Board temperature:** temperature of the controller output driver
 - **Light temperature:** temperature of the light (if a thermal sensor in the light is available and connected to the controller)
 - **Supply voltage:** actual supply voltage for the controller
 - **Driver current:** actual current drawn by the output driver
 - **Fan voltage:** actual supply voltage for the fan
 - **Internal analog voltage:** actual voltage of the internal analogue power supply
 - **Light voltage:** measured voltage of the light output
 - **Light current:** measured current of the light output
 - **Dimming voltage:** measured voltage of the external analogue dimming input
 - **Enable input signal:** actual state of the digital enable input signal
 - **Fault flags:** actual state of the five fault conditions
 - **Fault output signal:** actual state of the digital fault output signal
 - **Dipswitch setting:** actual setting of the 4-way dip switch
 - **Fan duty cycle:** actual duty cycle of the fan power supply

If the remote temperature sensor in the light is not available or simply not connected, the indication N/A, standing for Not Available, is displayed in the web page.

Generally, if a measurement is not available the indication N/A is displayed in the web page.

13.6.2. Controller setup

This page allows to inspect and change most of the settings related to the controller operation. For instance, from this page it is possible to set the output current to be driven into the light, to select how the input and output signals are to be handled and to specify the parameters necessary for the driving of the cooling fan in the light (if available). The **Controller setup** page is shown in the image of *Figure 7: the **Controller setup** web page*.

There are four main sections to set the functionality of the controller:

- **Main settings**
- **Fault detection**
- **Fan settings**
- **Oscillator settings**

The following paragraphs detail the **Controller setup** page of the controller.

Controller setup

Main settings

Enable control:
 Current control:
 Current range:
 Current preset: mA

Apply

Fault detection

When bad supply voltage:
 When board overtemperature:
 When light overtemperature:
 When light short:
 When light open:
 Board temperature low: °C
 Board temperature high: °C
 Light temperature low: °C
 Light temperature high: °C

Apply

Fan settings

Fan control:
 Fan temperature low: °C
 Fan temperature high: °C
 Fan dutycycle below temperature low: %
 Fan dutycycle above temperature high: %
 Fan dutycycle minimum: %
 Fan dutycycle maximum: %
 Fan dutycycle manual: %
 Fan PWM frequency: Hz

Apply

Oscillator settings

Oscillator control:
 Oscillator frequency: Hz

Apply

Figure 7: the **Controller setup** web page

13.6.2.1. Controller setup – Main settings

The fields under **Main settings** contain general settings that affect how the controller operates and reacts to the input and output signals. The **Main settings** section is shown in detail in the image of *Figure 8: the Main settings section*. The relevant fields are listed and described below.

- **Enable control** selects how the controller output can be enabled and disabled. It can be chosen between **Enable input signal** (default), **Force off** and **Force on**. When set to **Enable input signal** the controller output is enabled and disabled according to the external enable input signal. When set to **Force off** the controller output is always disabled. When set to **Force on** the controller output is always enabled
- **Current control** selects how the controller output current can be adjusted. It can be chosen between **Dimming input signal** (default) and **Preset value**. When set to **Dimming input signal** the output current can be adjusted with the knob on the top panel or the external analogue dimming input signal (see [chapter 13](#) for more information). When set to **Preset value** the output current is fixed to the user preset value specified in field **Current preset**
- **Current range** selects the range of the controller output current. It can be chosen between **Panel dip switch setting** (default), **400 mA**, **800 mA**, **1200 mA**, **1600 mA**, **2000 mA**, **2400 mA**, **2800 mA**, **3200 mA**, **3600 mA**, **4000 mA** and **According to current preset**. When set to **Panel dip switch setting** the controller abides to the current range specified with the 4-way dip switch on the top panel of the controller. When set to **400 mA**, **800 mA**, **1200 mA**, **1600 mA**, **2000 mA**, **2400 mA**, **2800 mA**, **3200 mA**, **3600 mA** or **4000 mA** the controller adapts to the same current range. When set to **According to current preset** the controller automatically selects the range according to the required current specified in field **Current preset**
- **Current preset** can be used to specify a fixed output current value. Factory value for this field is 0

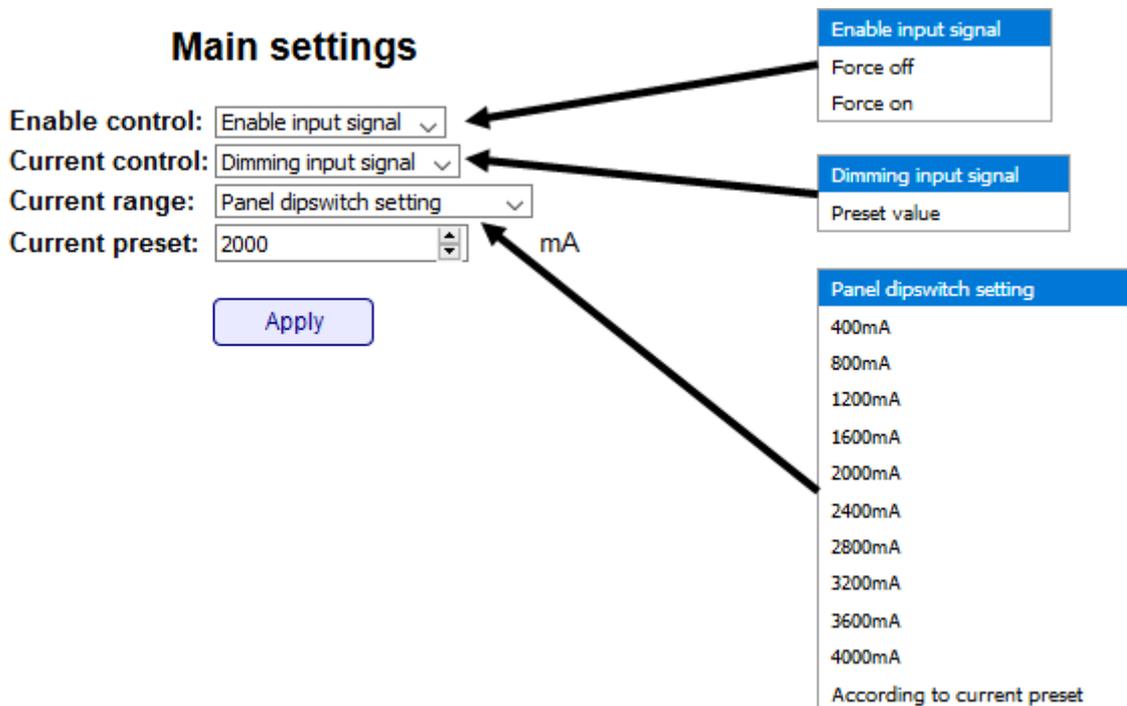


Figure 8: the Main settings section

13.6.2.2. Controller setup – Fault detection

The fields under **Fault detection** contain general settings that affect how the controller reacts to the various faults it can detect. The **Fault detection** section is shown in detail in the image of *Figure 9: the Fault detection section*. The relevant fields are listed and described below.

- **When bad supply voltage** selects how the controller reacts when a bad supply voltage is detected (neither $24\text{ V} \pm 10\%$ or $48\text{ V} \pm 10\%$). It can be chosen between **Ignore** and **Drive fault signal** (default). When set to **Ignore**, in case a bad supply voltage is detected the controller disables the output driver but does not drive the fault signal. When set to **Drive fault signal**, in case a bad supply voltage is detected the controller disables the output driver and drives the fault signal
- **When board overtemperature** selects how to controller reacts when an overtemperature is detected on the internal board. It can be chosen between **Ignore** and **Drive fault signal** (default). When set to **Ignore**, in case an overtemperature is detected on the internal board the controller disables the output driver but does not drive the fault signal. When set to **Drive fault signal**, in case an overtemperature is detected on the internal board the controller disables the output driver and drives the fault signal. To reduce the frequency of activation of the fault output and improve immunity to small temperature changes a hysteretic overtemperature detection is implemented and can be customized by changing fields **Board temperature low** and **Board temperature high** (see below)
- **When light overtemperature** selects how to controller reacts when an overtemperature is detected on the connected light. It can be chosen between **Ignore** and **Drive fault signal** (default). When set to **Ignore**, in case an overtemperature is detected on the connected light the controller disables the output driver but does not drive the fault signal. When set to **Drive fault signal**, in case an overtemperature is detected on the connected light the controller disables the output driver and drives the fault signal. To reduce the frequency of activation of the fault output and improve immunity to small temperature changes a hysteretic overtemperature detection is implemented and can be customized by changing fields **Light temperature low** and **Light temperature high** (see below)
- **When light short** selects how the controller reacts when a short circuit is detected in the light output. It can be chosen between **Ignore** (default) and **Drive fault signal**. When set to **Ignore**, in case a short circuit is detected in the light output the controller disables the output driver but does not drive the fault signal. When set to **Drive fault signal**, in case a short circuit is detected in the light output the controller disables the output driver and drives the fault signal
- **When light open** selects how the controller reacts when an open circuit is detected in the light output. It can be chosen between **Ignore** (default) and **Drive fault signal**. When set to **Ignore**, in case a short circuit is detected in the light output the controller disables the output driver but does not drive the fault signal. When set to **Drive fault signal**, in case a short circuit is detected in the light output the controller disables the output driver and drives the fault signal
- **Board temperature low** is the low temperature threshold for the hysteretic overtemperature detection of the internal board
- **Board temperature high** is the high temperature threshold for the hysteretic overtemperature detection of the internal board
- **Light temperature low** is the low temperature threshold for the hysteretic overtemperature detection of the connected light
- **Light temperature high** is the high temperature threshold for the hysteretic overtemperature detection of the connected light

Fault detection

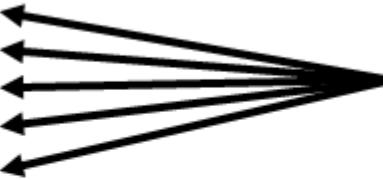
When bad supply voltage:	Drive fault signal	
When board overtemperature:	Drive fault signal	
When light overtemperature:	Drive fault signal	
When light short:	Ignore	
When light open:	Ignore	
Board temperature low:	80	°C
Board temperature high:	90	°C
Light temperature low:	85	°C
Light temperature high:	95	°C

Figure 9: the Fault detection section

13.6.2.3. Controller setup – Fan settings

The fields under **Fan settings** contain general settings that affect how the controller operates the fan in the light (if available). The **Fan settings** section is shown in detail in the image of *Figure 10: the Fan settings section*. The relevant fields are listed and described below.

- **Fan control** selects the fan operating mode. It can be chosen between **Proportional mode** (default), **Hysteretic mode** and **Manual mode**. When set to **Proportional mode** the duty cycle of the fan power supply is proportional to the temperature. When set to **Hysteretic mode** the duty cycle of the fan power supply toggles between two values according to the temperature history. When set to **Manual mode** the duty cycle of the fan power supply is a constant value specified by the user using field **Fan duty cycle manual** (see below)
- **Fan temperature low** is used to set parameter T_{Low} as described in [section 8.3.3](#).
- **Fan temperature high** is used to set parameter T_{High} as described in [section 8.3.3](#).
- **Fan duty cycle below temperature low** is used to set parameter $Duty_{Inf}$ as described in [section 8.3.3](#). Acceptable duty cycle is in the range from 0 % up to 100 %
- **Fan duty cycle above temperature high** is used to set parameter $Duty_{Sup}$ as described in [section 8.3.3](#). Acceptable duty cycle is in the range from 0 % up to 100 %
- **Fan duty cycle minimum** is used to set parameter $Duty_{Min}$ as described in [section 8.3.3](#). Acceptable duty cycle is in the range from 0 % up to 100 %
- **Fan duty cycle maximum** is used to set parameter $Duty_{Max}$ as described in [section 8.3.3](#). Acceptable duty cycle is in the range from 0 % up to 100 %
- **Fan duty cycle manual** is used to set the duty cycle when the fan is operated in manual mode. Acceptable duty cycle is in the range from 0 % up to 100 %
- **Fan PWM frequency** is used to set the frequency of the fan power supply. Allowed values are in the range from 100 Hz up to 50 kHz

Fan settings

Fan control:	Proportional mode ▾	←	Proportional mode
Fan temperature low:	60	°C	Hysteretic mode
Fan temperature high:	80	°C	Manual mode
Fan dutycycle below temperature low:	10	%	
Fan dutycycle above temperature high:	100	%	
Fan dutycycle minimum:	50	%	
Fan dutycycle maximum:	100	%	
Fan dutycycle manual:	75	%	
Fan PWM frequency:	250	Hz	

Figure 10: the Fan settings section

13.6.2.4. Controller setup – Oscillator settings

The fields under **Oscillator settings** contain general settings that affect how the controller operates the internal oscillator driving the power stage. The **Oscillator settings** section is shown in detail in the image of *Figure 11: the Oscillator settings section*. The relevant fields are listed and described below.

- **Oscillator control** selects if the oscillator is freerun or it is tuned to a specified frequency. It can be chosen between **Freerun mode** (default) and **Synchronized mode**. When set to **Freerun mode** the internal oscillator operates in freerun at a frequency of about 300 kHz. When set to **Synchronized mode** the internal oscillator is tuned to the frequency specified by field **Oscillator frequency** (see below)
- **Oscillator frequency** is used to specify the tuning frequency for the internal oscillator when operating in synchronized mode. Available tuning frequencies are in the range from 200 kHz up to 700 kHz

Oscillator settings

Oscillator control:	Freerun mode ▾	←	Freerun mode
Oscillator frequency:	200000	Hz	Synchronized mode

Figure 11: the Oscillator settings section

13.6.3. General setup

This page allows to inspect and change the password for the controller, the Ethernet interface parameters and the serial RS485 settings. The **General setup** page is shown in the image of *Figure 12: the General setup web page*.

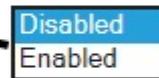
General setup

Enter unlock password

Current password:

Ethernet interface

MAC address:
Host name:
DHCP mode: 
IP address:
Subnet mask:
Gateway address:
Primary DNS:
Secondary DNS:
Modbus address: 
Modbus/TCP port: 
Modbus/UDP port: 



RS485 interface

Modbus address: 
Line speed: 
Line parity: 

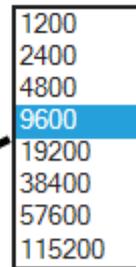
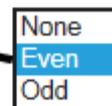



Figure 12: the **General setup** web page

A password can be used to deny the modification of the settings to unauthorized personnel. In case the password is set, it must be entered in the **Current password** field. If the entered password matches the saved password, the settings can be changed. In case the entered password does not match the saved password, the settings are displayed but they cannot be changed. The password can be modified in page **Advanced Setup** (see [section 13.5.4](#)).

Maximum length for the password is eight characters. The password is cleared when the settings are reset to the factory settings using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

The fields under **Ethernet interface** collect all the settings related to the Ethernet interface. It is possible to enable the use of a DHCP server, to change the IP address, the subnet mask, etc.

- **MAC address** is the unique MAC address of the Ethernet interface
- **Host name** is the name given to the controller
- **DHCP mode** enables or disables the use of a DHCP server
- **IP address** is the IP address assigned to the controller
- **Subnet mask** is the subnet mask
- **Gateway address** is the default gateway address
- **Primary DNS** is the address of the primary (preferred) DNS
- **Secondary DNS** is the address of the secondary (alternate) DNS
- **Modbus address** is the Modbus address to be used for the Modbus/TCP and Modbus/UDP protocols
- **Modbus/TCP port** is the TCP port used for the Modbus/TCP protocol
- **Modbus/UDP port** is the UDP port used for the Modbus/UDP protocol

The fields under **RS485 interface** collect all the settings related to the serial RS485 interface.

- **Modbus address** is the Modbus address to be used
- **Line speed** selects the speed to be used for the communication
- **Line parity** selects the parity to be used for the communication

Note the firmware supports only 9600 bits per second and even parity. Setting a different rate will cause the serial interface to operate at 9600 bits per second, while setting a different parity will cause the serial interface to operate with even parity.

13.6.4. Advanced setup

This page allows to set and modify the password used to protect the settings and to inspect or change the Modbus register file. The **Advanced setup** page is shown in the image of *Figure 13: the Advanced setup web page*.

Advanced setup

Change unlock password

Current password:

New password:

Repeat password:

Apply

Read Modbus register

Address:

Value:

Read

Write Modbus register

Address:

Value:

Write

Figure 13: the *Advanced setup* web page

The fields under **Change unlock password** can be used to set or modify the password employed to protect the settings of the controller from unauthorized modify.

- **Current password** is the current password (leave blank if there is not current password)
- **New password** is the new password to be used
- **Repeat password** is the new password to be used, repeated for safety

Maximum length for the password is eight characters. The password is cleared when the settings are reset to the factory settings using the INIT button (see [chapter 11](#) for a description of the INIT button functionalities).

The fields under **Read Modbus register** can be used to read the Modbus register file.

- **Address** is the address of the register to be read (decimal number)
- **Value** reports the contents of the register specified by the address (decimal number)

The fields under **Write Modbus register** can be used to write the Modbus register file.

- **Address** is the address of the register to be written (decimal number)
- **Value** contains the value that has to be written to the register specified by the address (decimal number)

Please refer to [chapter 13.4](#) for a comprehensive list of the Modbus registers available.

14. Operation with FabImage

FabImage Studio is a data-flow programming integrated development environment for machine vision applications. It provides a number of features and includes Modbus/TCP connectivity capabilities. The implementation of the Modbus protocol in FabImage Studio is complete, but for the use with this controller only a subset of functions, called filters, are necessary.

14.1. Filter selection

The filters are available in the top-left panel named **Toolbox** in the graphical interface. In this panel, under the **Program I/O** category, the **Modbus TCP I/O** group is available. See the following figure for reference.

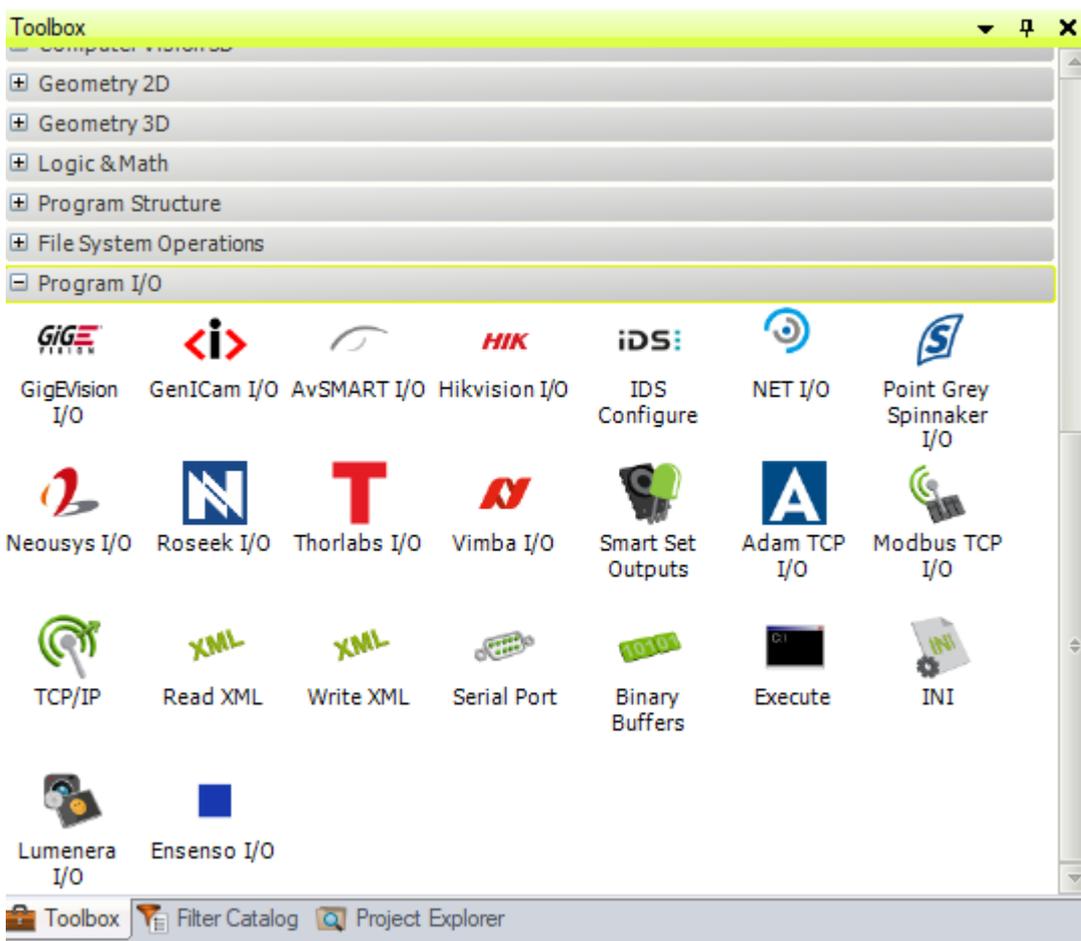


Figure 14. The **Program I/O** group of filters in FabImage

There are several Modbus TCP filters available. In order to see all of them and select those needed, drag and drop the **Modbus TCP I/O** icon to the central panel of the integrated development environment. A window called **Choose Filter Variant** prompting for the desired filter variant will open up.

The following figure shows the upper part of the window, where all the filters for reading data through Modbus are collected.

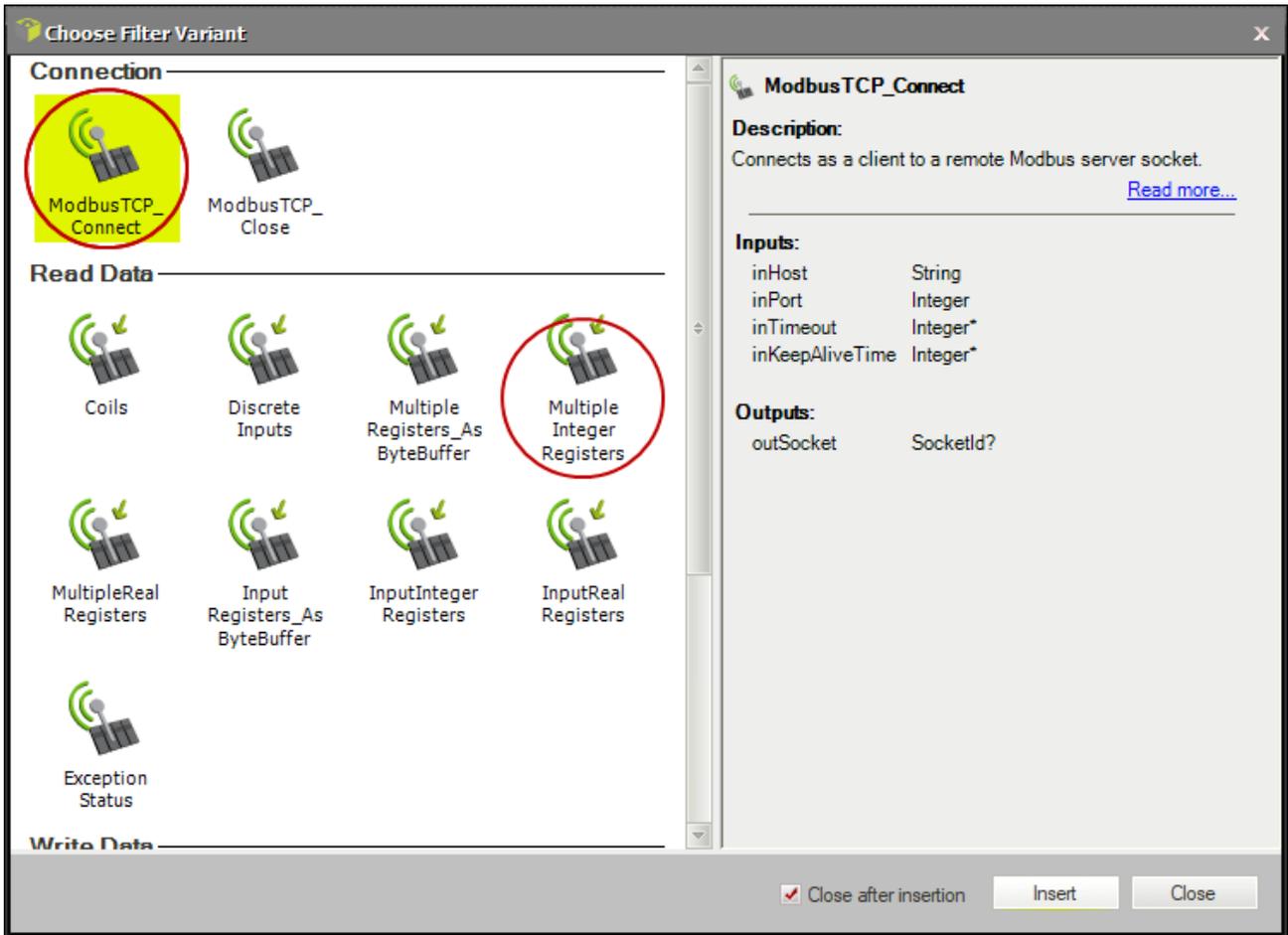


Figure 15. Selection of the Modbus TCP filter variant (top part)

The following figure shows the lower part of the window, where all the filters for writing data through Modbus are collected.

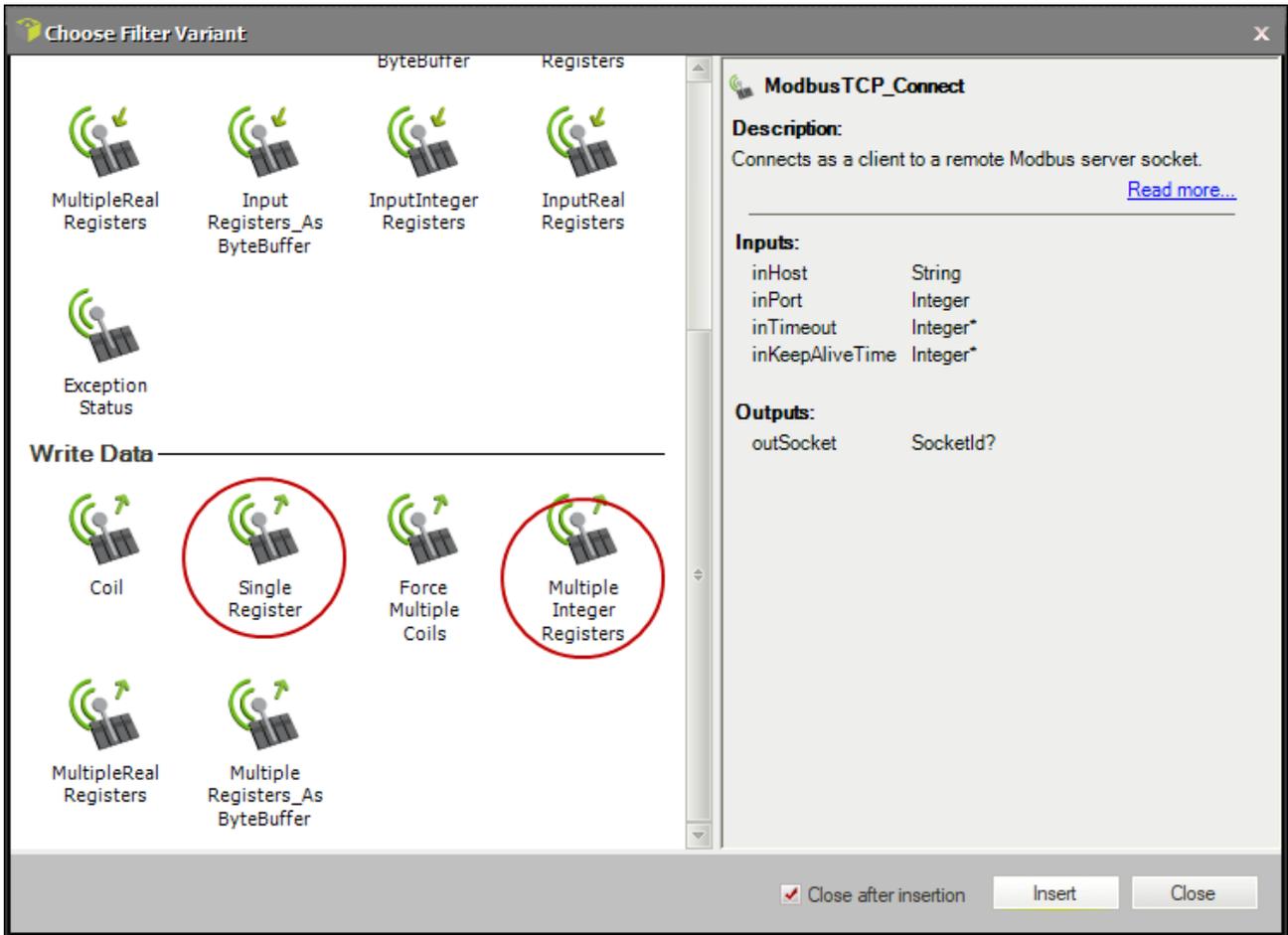


Figure 16. Selection of the Modbus TCP filter variant (bottom part)

In order to select one of the filter variants, click over it and then click on the **Insert** button in the lower right corner of the window. It is also possible to double-click on the desired variant. The filters useful for communicating with the controller are circled in red in the two previous pictures.

14.2. Designing a simple program for reading a register

With the explained procedure, build the data-flow code shown in the following figure.

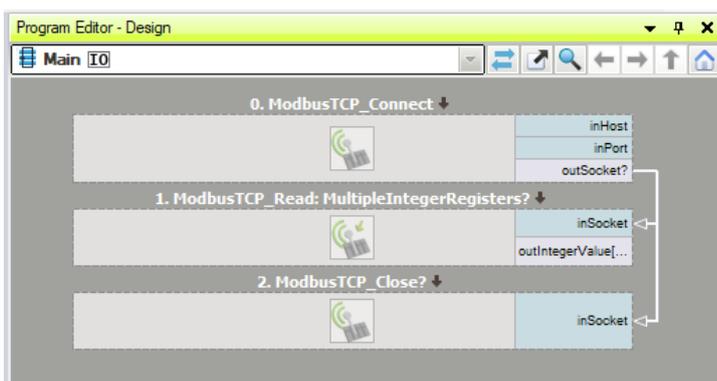


Figure 17. Example of data-flow code for reading a register

The connection between the **outSocket** output of filter **ModbusTCP_Connect** and the two **inSocket** inputs of filters **ModbusTCP_Read:MultipleIntegerRegisters?** and **ModbusTCP_Close** are made by drag and drop from the output to the inputs. Clicking on one filter instance will cause the filter property inspector to open on the bottom left panel of the user interface.

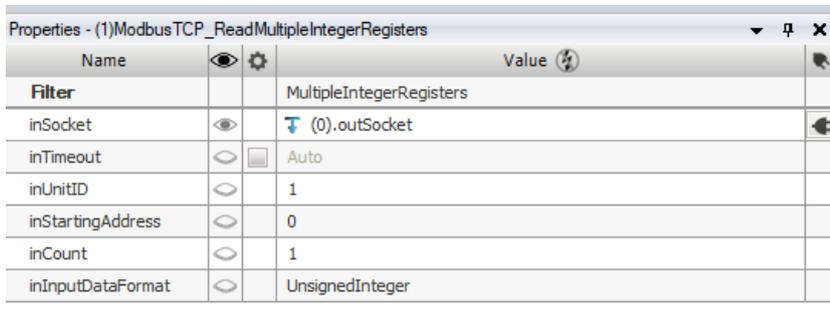
Configure the **ModbusTCP_Connect** filter as shown in the following figure.



Name	Visibility	Settings	Value
inHost	Visible		192.168.0.32
inPort	Visible		502
inTimeout	Visible	Auto	
inKeepAliveTime	Visible	Auto	

Figure 18. Configuration of **ModbusTCP_Connect** filter

Configure the **ModbusTCP_Read:MultipleIntegerRegisters?** filter as shown in the following figure.



Name	Visibility	Settings	Value
Filter			MultipleIntegerRegisters
inSocket	Visible		(0).outSocket
inTimeout	Visible	Auto	
inUnitID	Visible		1
inStartingAddress	Visible		0
inCount	Visible		1
inInputDataFormat	Visible		UnsignedInteger

Figure 19. Configuration of **ModbusTCP_Read:MultipleIntegerRegisters?** filter

The Modbus register at address 0 contains the device type as an unsigned 16-bit number. Changing the values of parameters **inStartingAddress** and **inCount** will allow the reading of different registers. Please refer to the Modbus register table of the controller for a list of all the registers available, the read and write capabilities of each one and the relevant allowed values.

After the read has completed ensure to close the connection with the controller using the **ModbusTCP_Close?** filter.

14.3. Designing a simple program for writing a register

With the explained procedure, build the data-flow code shown in the following figure.

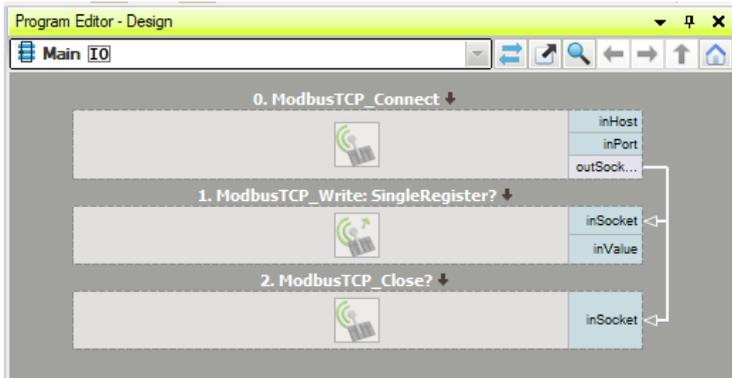


Figure 20. Example of data-flow code for writing a register

The connection between the **outSocket** output of filter **ModbusTCP_Connect** and the two **inSocket** inputs of filters **ModbusTCP_Write:SingleRegister?** and **ModbusTCP_Close** are made by drag and drop from the output to the inputs. Clicking on one filter instance will cause the filter property inspector to open on the bottom left panel of the user interface.

Configure the **ModbusTCP_Connect** filter as shown in the following figure.

Name	Visibility	Value
inHost	<input checked="" type="checkbox"/>	192.168.0.32
inPort	<input checked="" type="checkbox"/>	502
inTimeout	<input type="checkbox"/>	Auto
inKeepAliveTime	<input type="checkbox"/>	Auto

Figure 21. Configuration of **ModbusTCP_Connect** filter

Configure the **ModbusTCP_Write:SingleRegister?** filter as shown in the following figure.

Name	Visibility	Value
Filter		SingleRegister
inSocket	<input checked="" type="checkbox"/>	(0).outSocket
inTimeout	<input type="checkbox"/>	Auto
inUnitID	<input type="checkbox"/>	1
inRegisterAddress	<input type="checkbox"/>	0
inValue	<input checked="" type="checkbox"/>	0

Figure 22. Configuration of **ModbusTCP_Write:SingleRegister?** filter

The Modbus register at address 0 contains the device type as an unsigned 16-bit number. This register is defined as read only and any attempt to modify it will have no effect. Changing the values of parameters **inRegisterAddress** and **inValue** will allow the writing of different registers. Please refer to the Modbus register table of the controller for a list of all the registers available, the read and write capabilities of each one and the relevant allowed values.

Note that is possible to swap the **ModbusTCP_Write:SingleRegister?** filter with the **ModbusTCP_Write:MultipleIntegerRegisters?** filter. The latter one will allow the writing of multiple

Modbus registers located at sequential addresses.

After the write has completed ensure to close the connection with the controller using the **ModbusTCP_Close?** filter.

15. Electromagnetic compatibility

This product conforms to CENELEC EN 61326-1:2013 class A requirements for electromagnetic interference (EMI) suppression. EN 61326-1:2013 is equivalent to international standard IEC 61326-1, Ed. 2.0 (2012-07).

16. Firmware update procedure

The controller firmware can be updated using the serial RS485 interface and a specific PC application named **LTIC firmware updater**.

The first step is to connect the controller to be updated to the PC on which the LTIC firmware updater application will be run. If the PC does not have a native serial RS485 interface, a RS485-USB adapter like the ADPT001 may be used. On the controller end, the serial RS485 interface is available on the D+, D- and GND terminals of connector P4.

The image in *Figure 23: connection of RS485-USB adapter to the controller* shows the RS485-USB adapter connected to the controller using a Phoenix Contact 1803581 terminal block.



Figure 23: connection of RS485-USB adapter to the controller

The image in *Figure 24: main window of LTIC firmware updater application* presents the main window of the PC application.

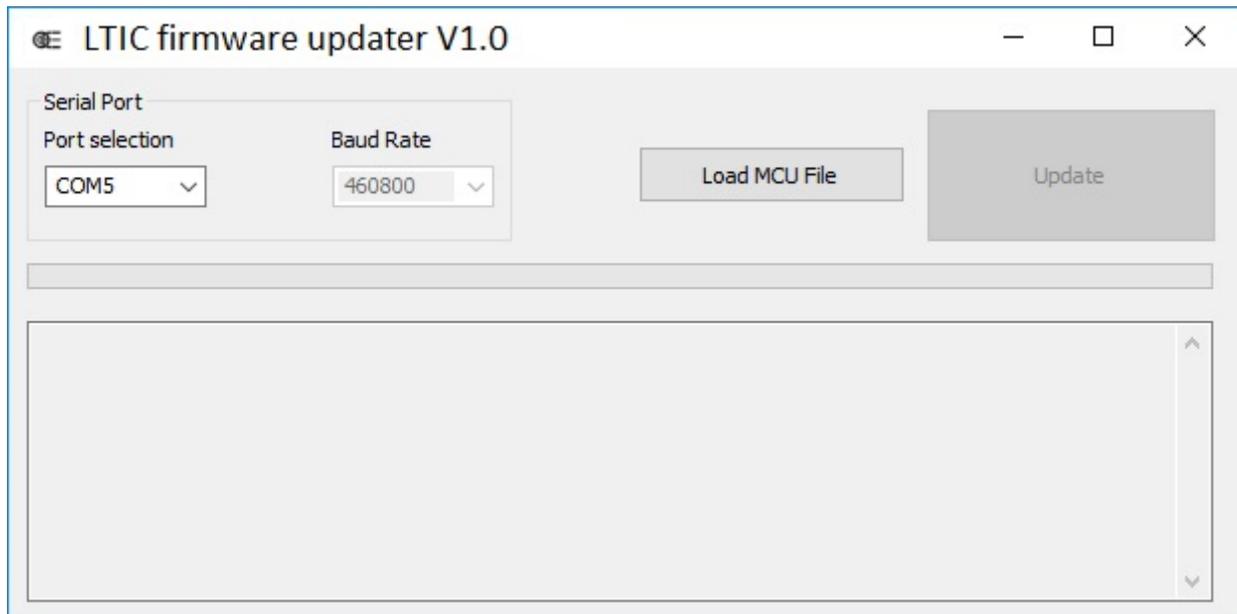


Figure 24: main window of LTIC firmware updater application

It is necessary to specify the communication port on the PC the controller is connected to. In the image **COM5** has been selected as the communication port. The baud rate is fixed at the maximum speed of 460,800 bits per second as indicated in the main window.

Inside the controller there is a programmable component, called MCU (Micro Controller Unit), that must be updated. The small button named **Load MCU file** at the right side of the window is used to locate the relevant programming file in the PC file system.

After pressing the **Load MCU file** button the modal dialog of *Figure 25: dialog used to locate the MCU firmware file* will be displayed, asking to specify the MCU file.

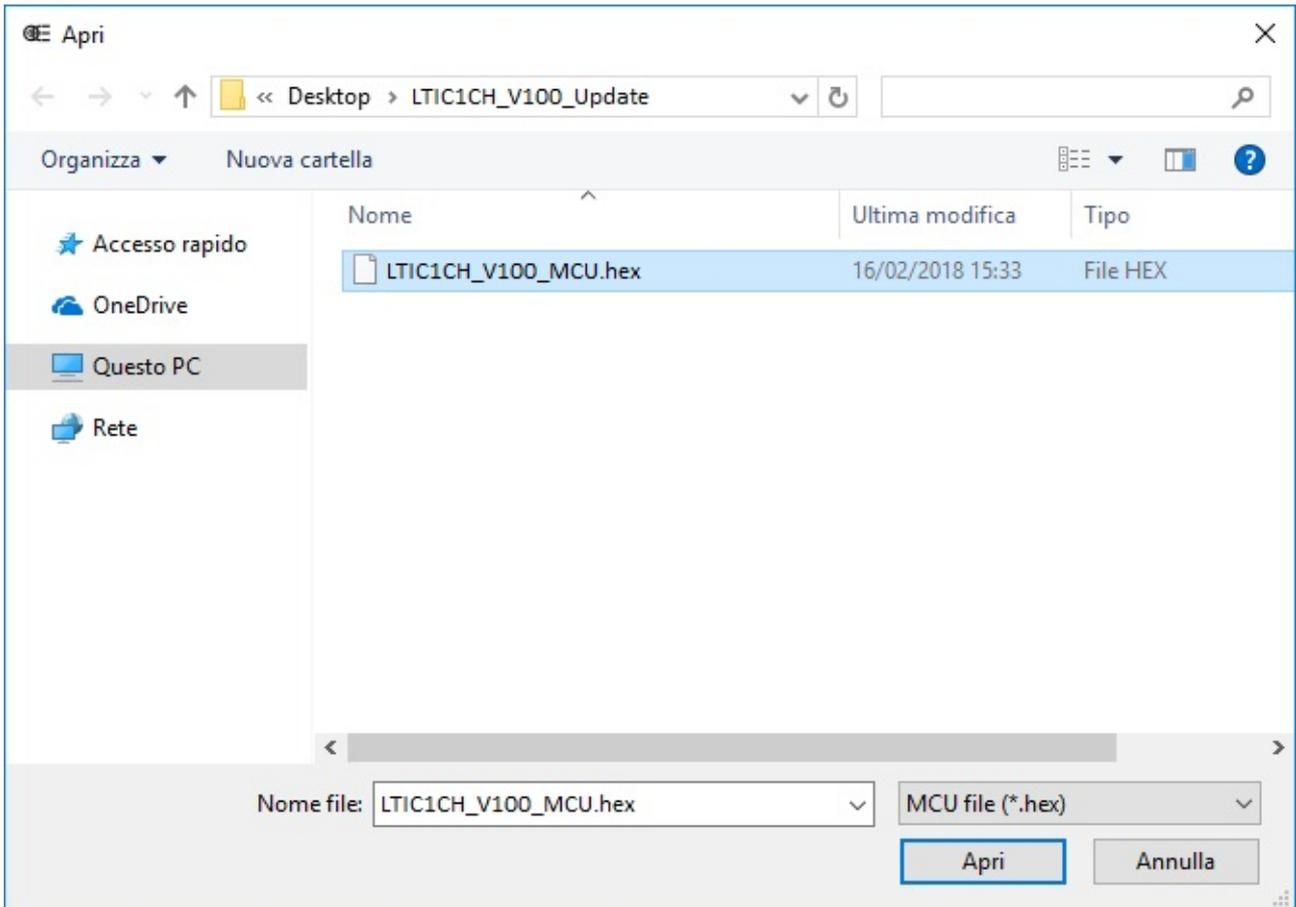


Figure 25: dialog used to locate the MCU firmware file

After the MCU file has been specified, the relative modal dialog is closed. The MCU firmware file will then be loaded from the PC disk to the PC RAM. The text box at the bottom of the main window will then be updated with some information regarding the loading process (see image in *Figure 26: main window after the MCU firmware file has been loaded*).

The **Update** button at the right side of the main window becomes active after the file has been loaded from the PC disk to the PC RAM.

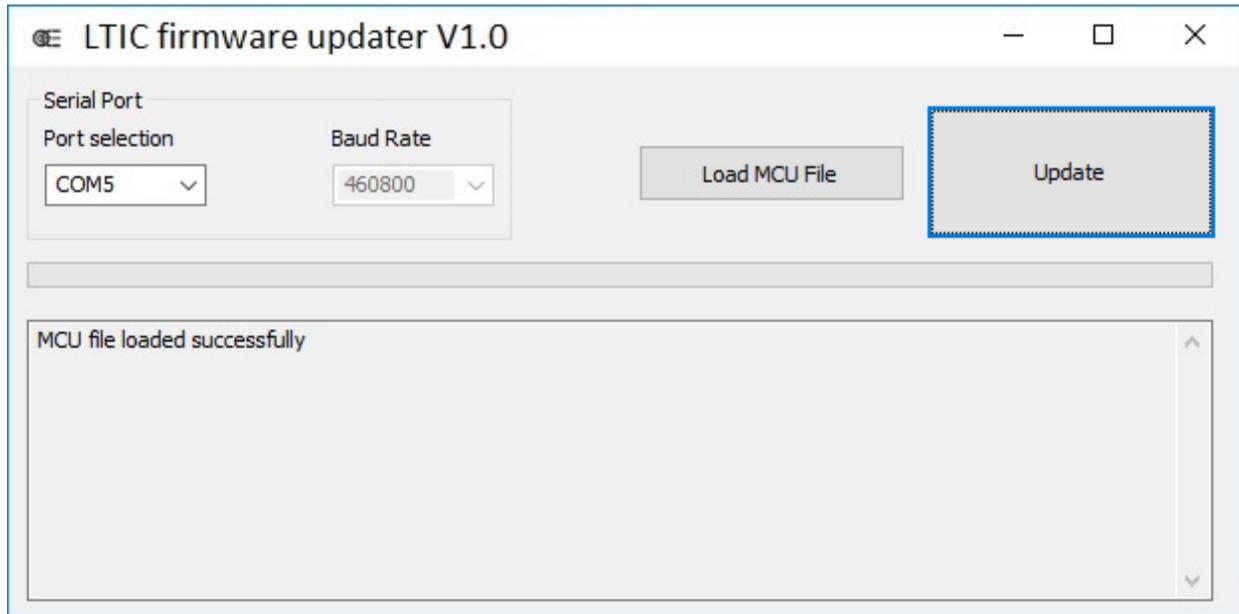


Figure 26: main window after the MCU firmware file has been loaded

At this point the PC application is ready to connect to the controller and transfer all the new firmware data. Now the controller must be switched off, if already powered, and then powered up with the INIT button held pressed for just a second. In this way the controller will enter a special state when, for ten seconds, will wait for new firmware data from the serial RS485 interface.

The firmware update sequence is then started by pressing the **Update** button. The PC application connects to the controller and starts the data transfer and programming. The progress bar in the middle of the main window keeps updating to show the advancement of the process. The text box at the bottom of the main window is updated with more information regarding the updating process.

The only data transferred to the controller is the MCU firmware (see image in *Figure 27: MCU firmware update sequence*).

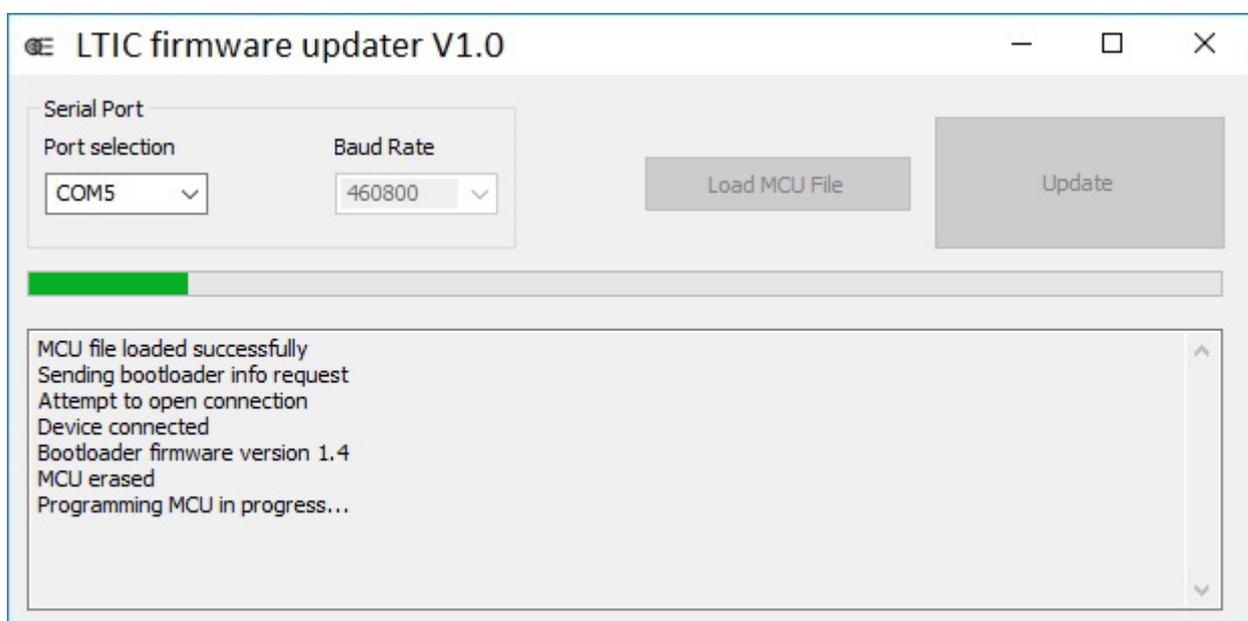


Figure 27: MCU firmware update sequence

The information in *Figure 28: main window after successful firmware update* is displayed when the update process is successfully finished.

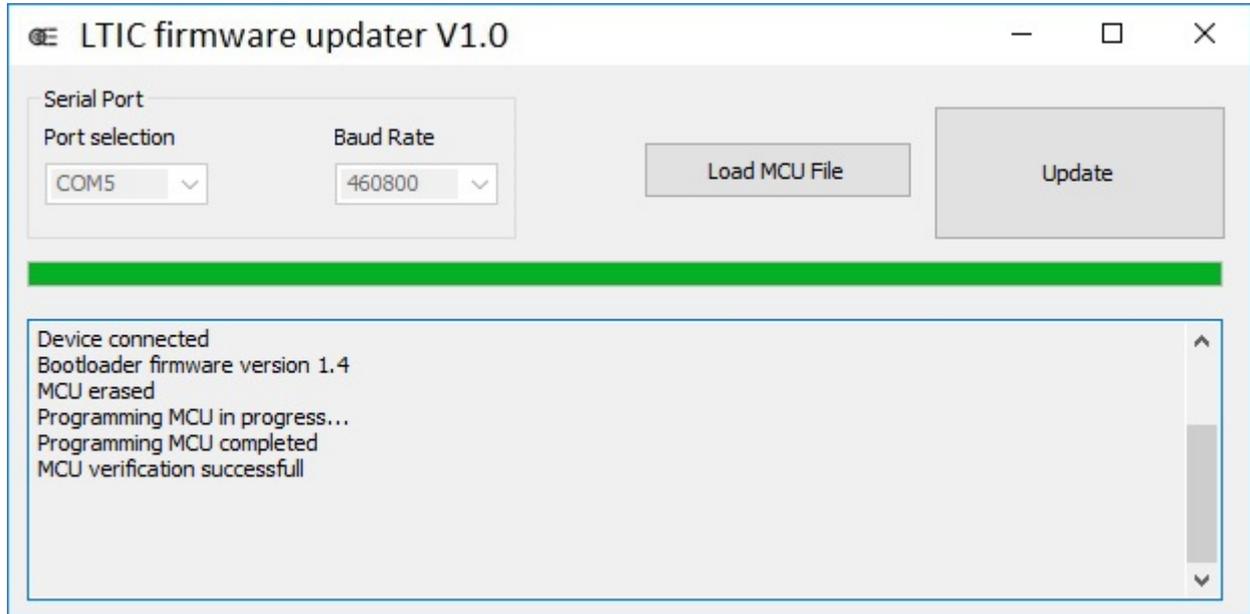


Figure 28: main window after successful firmware update

The whole update process takes about one minute to complete. In the unlikely event of failure it is advisable to repeat the process from the very beginning.



OPTO ENGINEERING

EUROPE

Opto Engineering
Europe Headquarters
Circonvallazione Sud, 15
46100 Mantova, IT
phone: +39 0376 699111
eu@opto-e.com

Opto Engineering
Germany
Marktplatz 3
82031 Grünwald
phone: +49 (0)89 693 9671-0
de@opto-e.com

Opto Engineering
Russia
official partner
ViTec Co., Ltd, Fontanka emb., 170
Saint-Petersburg, 198035, RU
phone: +7 812 5754591
info@vitec.ru

UNITED STATES

Opto Engineering
USA
11321 Richmond Ave
Suite M-105, Houston, TX 77082
phone: +1 832 2129391
us@opto-e.com

ASIA

Opto Engineering
China
Room 1903-1904, No.885, Renmin RD
Huangpu District 200010
Shanghai, China
phone: +86 21 61356711
cn@opto-e.com

Opto Engineering
Japan
official partner
Optart Corporation
4-54-5 Kameido Koto-ku
Tokyo, 136-0071 Japan
phone: +81 3 56285116
jp@opto-e.com

Opto Engineering
Korea
official partner
Far Island Corporation Ltd.
Seoil Building #703, 353 Sapyeong-daero,
Seocho-gu, Seoul, Korea 06542
phone: +82 70 767 86098
phone: +82 10 396 86098
kr@opto-e.com

WWW.OPTO-E.COM