



INSTRUCTIONS MANUAL

LTDVE2CH-20F

Strobe controller 2 CH – Firmware version 1.10



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
4.2. Operating mode.....	6
5. Getting started	7
6. Mechanical fixing.....	7
7. Heat dissipation	7
7.1. Calculating generated heat per channel.....	8
7.2. Reducing generated heat.....	8
8. Connections	8
8.1. Layout of connectors	9
8.2. Power and logic supply	10
8.3. Light outputs.....	10
8.4. Input/output synchronization	11
8.4.1. Synchronization inputs.....	11
8.4.2. Synchronization outputs.....	12
8.4.3. Serial RS485 interface	13
8.4.4. External temperature sensors	13
8.5. Cable size and length.....	14
9. Communication interfaces.....	14
9.1. Serial RS485 interface	15
9.2. Ethernet interface	15
10. Visual indicators	16
11. Functions of INIT button.....	16
12. Pulse shaping logic	17
12.1. Diagram of internal logic.....	17
12.2. Input filters.....	18
12.3. Input multiplexers	18
12.4. Pulse generators	19
12.5. Output multiplexers	19
12.6. Output protection.....	20
12.7. Free running oscillator.....	21

13. Wiring diagrams.....	21
13.1. Wiring example #1: controller triggers camera	21
13.2. Wiring example #2: camera triggers controller	22
14. Operation.....	23
14.1. Operation with Modbus	23
14.1.1. Comparison of Modbus/RTU, Modbus/TCP and Modbus/UDP	23
14.1.2. Supported function codes	23
14.1.3. Read Holding Registers (0x03).....	24
14.1.4. Write Single Register (0x06).....	24
14.1.5. Write Multiple Registers (0x10)	24
14.2. Register file	24
14.2.1. Register DEVICE_TYPE.....	38
14.2.2. Register BOOT_VERSION	38
14.2.3. Register MCU_VERSION	39
14.2.4. Register FPGA_VERSION	39
14.2.5. Register BOARD_VERSION	39
14.2.6. Register OSC_PERIOD	39
14.2.7. Registers FILTER_SEL[0-1]	39
14.2.8. Registers INPUT_SEL[0-3].....	39
14.2.9. Registers GEN_DELAY_BASE[0-3]	40
14.2.10. Registers GEN_DELAY_COUNT[0-3].....	40
14.2.11. Registers GEN_WIDTH_BASE[0-3]	41
14.2.12. Registers GEN_WIDTH_COUNT[0-3]	41
14.2.13. Registers OUTPUT_SEL_HI[0-1] and OUTPUT_SEL_HI[8-9].....	41
14.2.14. Registers OUTPUT_SEL_LO[0-1] and OUTPUT_SEL_LO[8-9].....	42
14.2.15. Registers PRT_CNT_ON[0-1].....	42
14.2.16. Registers PRT_ENA_ON[0-1].....	43
14.2.17. Registers PRT_CNT_OFF[0-1]	43
14.2.18. Registers PRT_ENA_OFF[0-1]	43
14.2.19. Registers CUR_RANGE[0-1].....	43
14.2.20. Registers CUR_VALUE[0-1]	44
14.2.21. Register RS485_MODBUS_ADDR.....	44
14.2.22. Register RS485_LINE_SPEED	44
14.2.23. Register RS485_LINE_PARITY	44
14.2.24. Registers ETH_MAC_ADDR[0-2].....	45
14.2.25. Registers ETH_HOSTNAME[0-7].....	45
14.2.26. Register ETH_DHCP_ENABLE	45

14.2.27. Register ETH_IP_ADDR_HI	45
14.2.28. Register ETH_IP_ADDR_LO	45
14.2.29. Register ETH_SUBNET_MASK_HI	46
14.2.30. Register ETH_SUBNET_MASK_LO	46
14.2.31. Register ETH_DEF_GATEWAY_HI	46
14.2.32. Register ETH_DEF_GATEWAY_LO	46
14.2.33. Register ETH_PRI_DNS_HI	46
14.2.34. Register ETH_PRI_DNS_LO	46
14.2.35. Register ETH_SEC_DNS_HI	46
14.2.36. Register ETH_SEC_DNS_LO	47
14.2.37. Register ETH_MODBUS_ADDR	47
14.2.38. Register ETH_MODBUS_TCP_PORT	47
14.2.39. Register ETH_MODBUS_UDP_PORT	47
14.2.40. Registers WEB_PASSWORD[0-3]	47
14.2.41. Register CONVERTER_TEMPERATURE	47
14.2.42. Register DRIVER_TEMPERATURE	48
14.2.43. Registers REMOTE_TEMPERATURE[0-1]	48
14.2.44. Register SUPPLY_VOLTAGE	48
14.2.45. Registers MEASURED_CURRENT[0-1]	48
14.2.46. Registers MEASURED_VOLTAGE[0-1]	48
14.2.47. Register ERROR_WORD	49
14.2.48. Registers GEN_HOLD_BASE[0-3]	49
14.2.49. Registers GEN_HOLD_COUNT[0-3]	50
14.2.50. Registers GEN_EDGE_SEL[0-3]	50
14.2.51. Registers POLARITY_SEL[0-1]	50
14.2.52. Registers DRIVE_TIME[0-1]	51
14.2.53. Registers CUR_RED_DELAY[0-1]	51
14.2.54. Registers CUR_RED_VALUE[0-1]	51
14.2.55. Register SOFTWARE_TRIGGER	52
14.2.56. Register CONVERTER_MODE	52
14.2.57. Register CONVERTER_MAXIMUM_VOLTAGE	53
14.2.58. Register CONVERTER_PRESET_VOLTAGE	53
14.2.59. Register DRIVER_MODE	53
14.2.60. Register DIGIPOT_VALUE	54
14.2.61. Register CONVERTER_VOLTAGE	54
14.2.62. Register DRIVER_VOLTAGE	54
14.2.63. Register SUPPLY_CURRENT	54

14.2.64. Register CONVERTER_CURRENT	54
14.2.65. Registers CAL_XXX	55
14.2.66. Registers CAL_UNLOCK_CODE[0-1]	55
14.2.67. Register BOARD_COMMAND	55
14.3. Operation with a web browser	55
14.3.1. Main page and navigation menu	56
14.3.2. Setup synch inputs	57
14.3.3. Setup pulse generators	58
14.3.4. Setup light outputs	60
14.3.5. Setup synch outputs	63
14.3.6. General setup	63
14.3.7. Advanced setup	65
15. Operation with FabImage	66
15.1. Filter selection	67
15.2. Designing a simple program for reading a register	69
15.3. Designing a simple program for writing a register	70
16. Electromagnetic compatibility	72
17. Firmware update procedure	72

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 controller outputs pulses with high energy content. 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 that includes power supply conditioning, intensity control, timing generation and advanced triggering functions.

The controller can be set up using a PC with serial RS485 or Ethernet interfaces. Configurations are saved in non-volatile memory so that the controller will resume operation after a power cycle.

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.

4.2. Operating mode

This strobe controller has two programmable, current-controlled light outputs. The two light outputs can be used in pulsed or continuous mode.

In pulsed mode the light is switched on only when necessary. A digital input is used as a trigger source. When a rising or falling edge on the trigger signal is detected the output is pulsed for the programmed amount of time.

Using this technique, it is possible to obtain excellent steady images of moving objects. The camera can be set for an arbitrary long exposure time and the light turned on for a shorter time, just enough to freeze the motion. This helps to overcome the uncertainty issues usually related with integration start which, to some degree, afflict most commercial cameras.

The delay from the trigger to the output pulse, the width of the output pulse and the intensity of the output pulse are all independently configurable. The pulse delay can range from 0 μ s to 1 s. The pulse width can range from 1 μ s to 1 s.

In continuous mode the light is always switched on, independently from the trigger signal. Using this technique, the maximum current value for each channel has to be limited in order to prevent the overheating of the controller.

The two output drivers in the controller can either operate in two modes: independent mode and shared mode. In independent mode, the two light outputs are completely autonomous and currents and timings are independent.

In the shared mode the two output drivers are interconnected. This results in twice the maximum output current than in the independent mode. In the shared mode, the controller effectively behaves as a single channel controller.

In either independent mode or shared mode there are two current ranges. In independent mode the current ranges are:

- Low current, up to 250 mA (with a resolution of 1 mA)
- High current, up to 20 A (with a resolution of 20 mA)

In shared mode the current ranges are:

- Low current, up to 500 mA (with a resolution of 2 mA)
- High current, up to 40 A (with a resolution of 40 mA)

The controller must be powered with a fixed supply voltage of 24 V DC. There is an internal DC/DC converter that can be programmed to provide any voltage between 5 V and 190 V for supplying the output drivers. This allows a large number of different lights to be efficiently driven.

The DC/DC converter is power limited to around 60 W. Limit maximum continuous current per channel to:

- 2 A at a maximum voltage of 15 V
- 1 A at a maximum voltage of 30 V

In any case, if any of the lights is driven continuously limit the DC/DC converter output voltage to a maximum value of 55 V.

For more information about current and power limitations refer to [chapter 7](#).

5. Getting started

Carefully read the sections on [Safety Notes](#) and [Heat Dissipation](#) and check the product fits your needs. Mount the controller 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 PWR LED lit with a stable green colour and the RUN LED lit with a flashing green colour.

Read the section on [Operation](#). The controller can be configured by using both a serial RS485 interface and an Ethernet interface (see [chapter 9](#)).

6. Mechanical fixing

The controller must be mounted on a suitable thermally conductive surface in order to dissipate the generated heat. 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 linear circuits to produce the constant current outputs. This means that it generates heat which needs to be dissipated. The operating temperature range is 0 °C to 40 °C.

With a suitable heatsink the controller can approximately dissipate the following average powers:

- 20 W at 25 °C (about 10 W per channel)
- 15 W at 40 °C (about 7.5 W per channel)

A simple way to estimate the maximum average power the controller can dissipate is by applying the following formula:

$$DissipablePower [W] = (TempHeatsink [^{\circ}C] - TempAmbient [^{\circ}C]) / ThResistance [^{\circ}C/W]$$

Where:

- *DissipablePower* is the maximum average power the controller can dissipate
- *TempHeatsink* is the maximum temperature of the controller heatsink
- *TempAmbient* is the actual temperature of the ambient where the controller is placed
- *ThResistance* is the thermal resistance between the heatsink and the ambient

For this controller the *ThResistance* parameter is about 3.2 °C/W.

The maximum permissible controller heatsink temperature is 90 °C. If the heatsink temperature rises above 90 °C, the controller switches off all the output channels. The output channels are then reactivated once temperature falls below 80 °C.

If the average power that must be dissipated is greater than the previously stated value, a different and more efficient cooling system is required. Solutions could be the use of a cooling fan (active cooling system) or the use of a bigger heatsink (passive cooling system).

There is an internal DC/DC converter that can be programmed to provide any voltage between 5 V and 190 V for supplying the lights. Take care of the actual converter output voltage when calculating the generated heat.

7.1. Calculating generated heat per channel

For a pulsed output, the average power that is transformed to heat and then must be dissipated can be calculated using the following formula:

$$\text{Heat [W]} = \text{LightCurrent [A]} * (\text{DriverVoltage [V]} - \text{LightVoltage [V]}) * \text{DutyCycle [-]}$$

Where:

- *LightCurrent* is the illuminator operating current
- *LightVoltage* is the illuminator operating voltage
- *DriverVoltage* is the actual DC/DC converter output voltage (from 5 V to 190 V)
- *DutyCycle* is the actual duty cycle

The duty cycle is given by:

$$\text{DutyCycle [-]} = \text{PulseWidth [s]} * \text{TriggerFrequency [Hz]}$$

If the output is driven in continuous mode, the previous equations are still valid but the parameter *DutyCycle* becomes 1.0 because the output is always active. In continuous mode limit the output current to 2 A at a maximum voltage of 15 V and to 1 A at a maximum voltage of 30 V.

The parameters *LightCurrent* and *LightVoltage* are light specific and should be either given in the light documentation or measured experimentally.

7.2. Reducing generated heat

The total heat generated by the controller is simply given by adding the generated heat for each of the two channels, as calculated in the previous section.

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 pulse width or output current, if permitted by the application.

Another strategy to reduce the generated heat would be to connect lights in series instead of parallel, if possible. If you have several lights connected in parallel then changing the arrangement to series will increase the voltage across them but also reduce the overall current.

The last option, feasible with this two channels controller, would be to use two or more controllers and use just a few channels from each. For high power applications this may be the only solution.

8. Connections

See the next sections for information about connections. Power supply and light output connections are made via screw terminals on the right-side panel of the controller. Check all connections carefully before switching on the equipment.

The controller has two 24 V DC power supplies: a dedicated power supply for the power stages and a dedicated power supply for the logic section. This is to increase versatility.

Inside the controller, supply to the logic circuits is derived using a pair of diodes from both of these power supplies. This means that either of the two supplies can power the logic circuits.

Ideally, supply to the power stages could be removed at any time to protect the end user from photobiological and other hazards that can occur during fault conditions. Should supply to the power stages be removed while the system is running, the system designer may consider providing the dedicated logic supply to keep the controller powered and responsive.

For convenience, the two power supplies share a single, common negative terminal.

8.1. Layout of connectors

The drawing in *Figure 1: connectors on the controller* depicts all the controller connections, which are easily accessible on the left-side and right-side panels. As indicated in the drawing, connectors are identified by their unique designators (P1, P2, P3, P4 and P5).

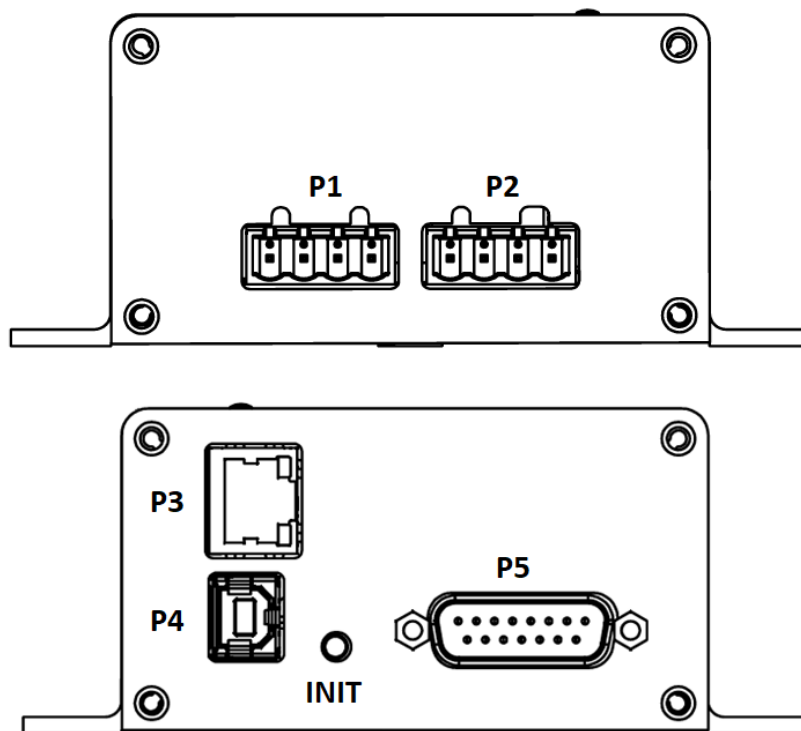


Figure 1: connectors on the controller

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

- Connector P1 is used to supply power
- Connector P2 is used to connect the two lights
- Connector P3 is an Ethernet RJ45 jack
- Connector P4 is a USB port (B type), not active at the moment
- Connector P5 is used for input/output synchronization and for serial RS485 communication

For connectors P1, P2 and P5 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, P2	Phoenix Contact	1757035
P5	MH Connectors	MHDM15SS

Table 1: mating plugs for the controller connectors

8.2. Power and logic supply

The power supply voltage must be 24 V DC. A dedicated and well-regulated switching power supply is required. The external power supply must be capable of supplying the average and peak currents needed for all active light outputs.

Choose a power supply unit that limits its output current by design or use protecting fuses. The fuses 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 these power connections is appropriate for the current to be drawn. The power supply low voltage and mains wiring should be separately routed.

Power supply is delivered to the controller using the screw terminals of connector P1. Connector pinout, ordered from left to right, is listed in *Table 2: pinout of connector P*.

Number	Name	Description	Note
1	EARTH	Protective earth	
2	+V LOG	Power supply. Positive terminal	Used for logic section
3	0V	Power supply. Negative terminal	
4	+V PWR	Power supply. Positive terminal	Used for power stages

Table 2: pinout of connector P1

The controller has two 24 V power terminals to independently supply the logic and power sections inside the unit. They are named +V LOG and +V PWR. These two supplies can be connected together or separately, as required by the application. They share the common negative terminal named 0V. It must be connected to the power supply negative.

Ensure that the polarity of +V LOG, +V PWR and 0V is correct before applying power.

8.3. Light outputs

Light outputs are paired on the 4-way pluggable screw terminal socket named P2. It is possible to use two 2-way connectors in the 4-way socket. The light output connections must not be paralleled or grounded in any way.

The state of each output is shown by a yellow LED indicator on the top panel of the controller.

Make sure you set the correct current rating for a light before using it. See the light datasheet and manual for details on this topic.

Connector pinout, ordered from left to right, is listed in *Table 3: pinout of connector P*. Be careful not to cross-connect the two lights.

Number	Name	Description	Note
1	LD1+	Power channel 1 output. LED anode	
2	LD1-	Power channel 1 output. LED cathode	
3	LD2+	Power channel 2 output. LED anode	
4	LD2-	Power channel 2 output. LED cathode	

Table 3: pinout of connector P2

Please note that LED1- and LED2- are not the same as 0V.

8.4. Input/output synchronization

Connector P5 is used for input and output synchronization and for serial RS485 communication.

There are two independent, galvanically isolated, synchronization inputs. These inputs can be connected directly to the system for voltages up to 24 V. An external series resistor is not necessary. The synchronization inputs may be left unconnected when not used. The state of each synchronization input is shown by a green LED indicator on the top panel of the controller.

There are two independent, galvanically isolated, synchronization outputs. These outputs can be used, for example, to trigger a camera or a slave controller. These outputs can be connected directly to the system for voltages up to 24 V. The state of each synchronization output is shown by a yellow LED indicator on the top panel of the controller.

Connector P5 also provides three signals for a non-electrically isolated serial RS485 interface and four signals for two optional and non-electrically isolated external temperature sensors. The activity of the serial RS485 interface is shown by a dedicated yellow LED on the top panel of the controller.

See the following section for more information about connector P5.

8.4.1. Synchronization inputs

The two synchronization inputs are available on the TR1+, TR1-, TR2+ and TR2- terminals of connector P5. These signals are listed in *Table 4: pinout of connector P5 for synchronization inputs*. Be careful not to cross-connect the two synchronization inputs.

Pin number	Name	Description
1	TR1-	Input 1. Negative terminal
9	TR1+	Input 1. Positive terminal
2	TR2-	Input 2. Negative terminal
10	TR2+	Input 2. Positive terminal

Table 4: pinout of connector P5 for synchronization inputs

The schematic of *Figure 2: interface circuits for input synchronization* depicts the internal input circuits. An internal constant current generator connected in series with each input allows for a broad range of input voltages without any need for a series resistor. These inputs can be directly driven by voltages up to 24 V.

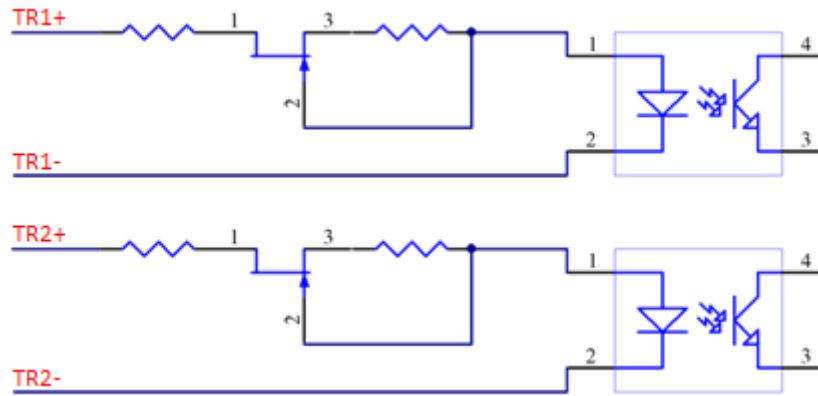


Figure 2: interface circuits for input synchronization

Circuit specifications are summarized in *Table 5: specifications of input synchronization circuits*. Please note the reported values are typical.

Parameter	Value	Unit	Note
U _{in} (low)	0 – 1	V	-
U _{in} (high)	2.5 – 24	V	-
I _{in}	2.6 – 5.7	mA	Internal constant-current generator

Table 5: specifications of input synchronization circuits

8.4.2. Synchronization outputs

The two synchronization outputs are available on the SH1+, SH1-, SH2+ and SH2- terminals of connector P5. These signals are listed in the *Table 6: pinout of connector P5 for synchronization outputs*. Be careful not to cross-connect the two synchronization outputs.

Pin number	Name	Description
3	SH1-	Output 1. Emitter terminal
11	SH1+	Output 1. Collector terminal
4	SH2-	Output 2. Emitter terminal
12	SH2+	Output 2. Collector terminal

Table 6: pinout of connector P5 for synchronization outputs

The schematic of *Figure 3: interface circuits for output synchronization* depicts the internal output circuits. These outputs can be directly connected to voltages up to 24 V.

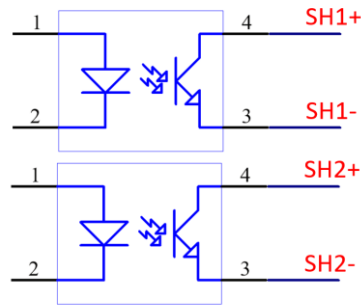


Figure 3: interface circuits for output synchronization

Circuit specifications are summarized in *Table 7: specifications of output synchronization circuits*. Please note the reported values are typical.

Parameter	Value	Unit	Note
I _{out} (typ)	10	mA	-
I _{out} (max)	15	mA	-
U _{out} (max)	24	V	-

Table 7: specifications of output synchronization circuits

8.4.3. Serial RS485 interface

The serial interface is available on the D+, D- and GND terminals of connector P5. These signals are listed in *Table 8: pinout of serial interface in connector P*. Be careful not to cross-connect the serial interface signals.

Pin number	Name	Description	Note
7	D-	RS485 data signal. Negative terminal	-
8	GND	RS485 reference ground	-
15	D+	RS485 data signal. Positive terminal	-

Table 8: pinout of serial interface in connector P5

The interface is NOT electrically isolated. Note that GND is internally connected to 0V.

8.4.4. External temperature sensors

The controller allows for the connection of two external temperature sensors. 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 signals are listed in *Table 9: pinout of external temperature sensors in connector P*. The four terminals can be connected freely to the external thermistors, as the components are not polarized.

Pin number	Name	Description
------------	------	-------------

5	NTC1_A	Temperature sensor 1 terminal A
13	NTC1_B	Temperature sensor 1 terminal B
6	NTC2_A	Temperature sensor 2 terminal A
14	NTC2_B	Temperature sensor 2 terminal B

Table 9: pinout of external temperature sensors in connector P5

These analogue signals are not electrically isolated from the controller electronics. Be careful not to connect them to any other signal. A severe malfunction or even a short circuit may occur.

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 10: 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 and logic supply	1.5	15	5
Light output	0.75	18	3
Synchronization inputs	0.25	24	5
Synchronization outputs	0.25	24	5
Serial RS485 interface	0.25	24	5
External temperature sensors	0.25	24	5

Table 10: cable wire size and length

For improved immunity against external disturbance sources, use a single shielded cable or multiple shielded cables, grounded at the end opposite to the controller, on the synchronization inputs, synchronization outputs, serial RS485 interface and external temperature sensors signals.

For the lights use cables as short as possible and with appropriate wire size. Cable reactance limits performance in pulsed mode, consider to reduce its value by connecting two or more smaller wires in parallel. For long cables it is recommended to raise the voltage of the illumination. This can be realized by selecting lights with LEDs connected in series rather than connected in parallel.

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 14](#) 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 a dedicated yellow LED on the top panel of the controller.

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 11: default parameters for Ethernet communication*.

Parameter	Default Value
Host name	LTDVE2CH-20F
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 11: 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 twelve LEDs on the top panel of the controller and two LEDs embedded in the Ethernet RJ45 jack. Some of them are used to show that power supplies are available, others are pulsed when inputs and outputs are activated, while others are used to indicate activity on the communication interfaces or fault conditions.

The exact meaning of each of the LEDs is listed in *Table 12: 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	485	Yellow	Blinks when there is activity on the serial interface
2	TR1	Green	Pulses when synchronization input 1 is activated
3	TR2	Green	Pulses when synchronization input 2 is activated
4	SH1	Yellow	Pulses when synchronization output 1 is activated
5	SH2	Yellow	Pulses when synchronization output 2 is activated
6	LD1	Yellow	Pulses when light output 1 is activated
7	LD2	Yellow	Pulses when light output 2 is activated
8	SHR	Green	Stable when the light outputs are shared
9	ARM	Yellow	Stable when the DC/DC converter output is enabled
10	PWR	Green	Stable when logic supply is present
11	RUN	Green	Blinks periodically during normal operation
12	ERR	Red	Stable when power supply missing, blinks in error conditions
13	ACT	Yellow	Blinks during Ethernet data transmission
14	LINK	Green	Stable when Ethernet connection established

Table 12: meaning of the LEDs

Either the RUN LED or the ERR LED blinks for 500 ms at power on to identify the source of data used for the settings.

The green RUN LED blinks when the controller powers up using the settings stored in the non-volatile memory (the last configuration saved by the customer). The red ERR LED blinks when the controller reverts to using the factory settings due to user activation of the INIT button (see next section for more information on the INIT button) or a corruption in the stored customer settings.

Please note the logic supply must be present in order for all the LEDs to turn on.

11. Functions of INIT button

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. Pulse shaping logic

Each of the two light channels can be individually configured to output pulses based either on a discrete external trigger signal or an internally-generated trigger signal. A wide variety of internal triggers can be produced by configuring the internal pulse shaping logic.

This logic includes four pulse generators and several multiplexers. The pulse generators allow pulse delay and width control down to 1 μ s resolution. The multiplexers, organized as two routing matrices, allow for the flexible selection of the pulse generators inputs and outputs. The pulse generators can be excluded or bypassed when implementing continuous mode.

An output protection circuit, used to prevent the light from getting overheated and thus damaged, is also included in the logic.

12.1. Diagram of internal logic

The drawing of *Figure 4: diagram of internal logic network* depicts the logic network built in the controller.

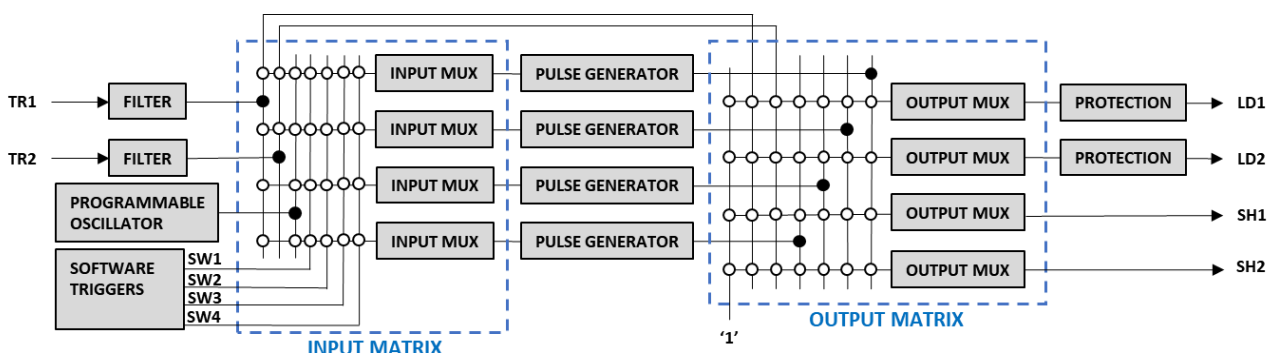


Figure 4: diagram of internal logic network

The two synchronization inputs are shown at the left (TR1 and TR2), while the two light outputs (LD1

and LD2) and the two synchronization outputs (SH1 and SH2) are drawn at the right.

A description of each of the blocks is given in the next sections.

12.2. Input filters

The input filters are used to debounce and remove glitches from the incoming synchronization inputs. Each of the two synchronization inputs has a dedicated, independent filter.

The algorithm implemented in each of the filters processes the relevant synchronization input with a finite state machine. A change in the filter output is performed only when the input signal has remained constant for a defined period of time, called filter time constant. Any pulses shorter than the filter time constant are thus removed and not passed through.

The diagram in *Figure 5: operation of the input filter* shows the filter operation on a random input signal.

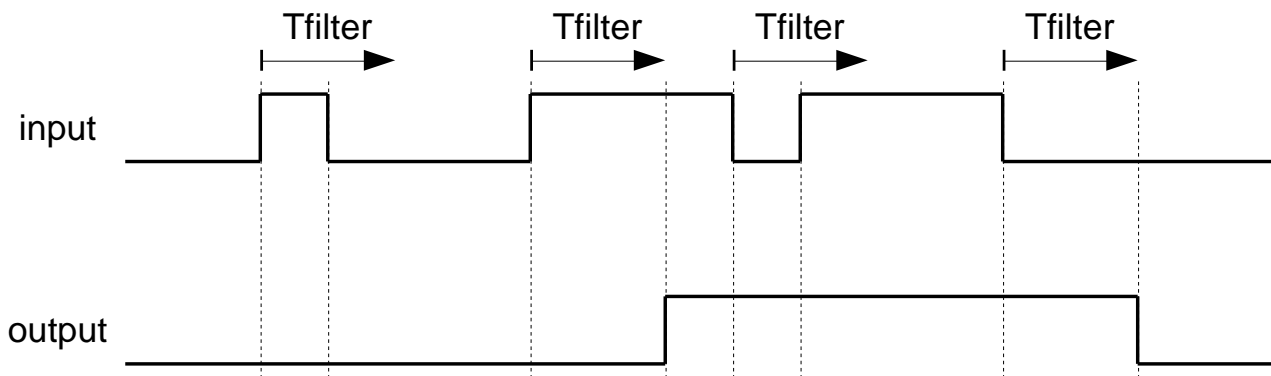


Figure 5: operation of the input filter

As visible, the input signal is filtered by looking for pulses that hold the same state for a time of at least **Tfilter** before the change in state is passed to the output. Please note there is a fixed input to output propagation delay equal to this filter time constant.

Each of the two filters can be set as follows:

- No filtering (pass through)
- Filtering with a 10μs time constant
- Filtering with a 20μs time constant
- Filtering with a 50μs time constant
- Filtering with a 100μs time constant
- Filtering with a 200μs time constant
- Filtering with a 500μs time constant

Setting of the filters can be done using the serial RS485 or Ethernet interfaces.

12.3. Input multiplexers

The input multiplexers are used to route the filtered inputs to the pulse generators. There are four input multiplexers organized in a 7x4 routing matrix.

Each multiplexer can have its output selected from one of the following sources:

- No selection

- Filtered synchronization input 1 (TR1)
- Filtered synchronization input 2 (TR2)
- Free running oscillator
- Software trigger 1 (SW1)
- Software trigger 2 (SW2)
- Software trigger 3 (SW3)
- Software trigger 4 (SW4)

The free running oscillator is an autonomous asynchronous trigger source described in detail in the [chapter 12.7](#). Setting of the input multiplexers can be done using the serial RS485 or Ethernet interfaces.

12.4. Pulse generators

There are four pulse generators. Each of them is characterized by three parameters: pulse delay, pulse width and hold off interval. The pulse delay can range from 0 μs to 1,023,000 μs with variable resolution down to 1 μs . The pulse width can range from 1 μs to 1,023,000 μs with variable resolution down to 1 μs . The hold off interval can range from 0 μs to 1,023,000 μs with variable resolution down to 1 μs .

The diagram in *Figure 6: time diagram of pulse generator* describes the relationship between input and output. As depicted, the rising edge of the input signal triggers the generator, while the falling edge has no special meaning and can happen anywhere in time.

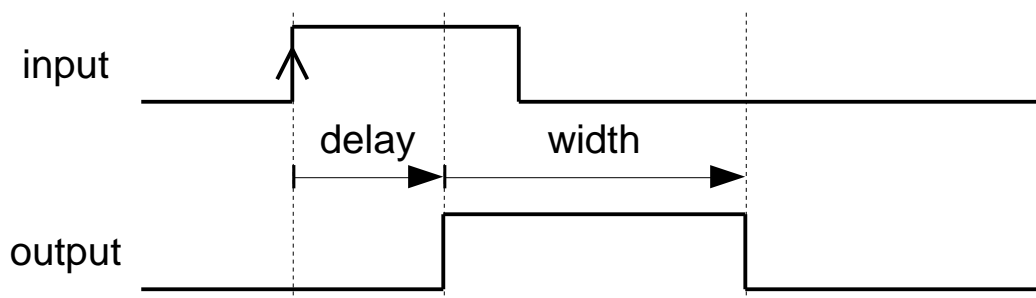


Figure 6: time diagram of pulse generator

Setting of the pulse generators can be done using the serial RS485 or Ethernet interfaces.

12.5. Output multiplexers

The output multiplexers are used to route the inner signals to the output stages. There are four output multiplexers organized in a 7x4 routing matrix.

Each multiplexer can have its output selected from one of the following sources:

- No selection
- Pulse generator 1 output
- Pulse generator 2 output
- Pulse generator 3 output
- Pulse generator 4 output
- Filtered synchronization input 1 (TR1)

- Filtered synchronization input 2 (TR2)
- Continuous

As visible in the internal logic network diagram (see *Figure 4: diagram of internal logic network*), the four pulse generators can be entirely bypassed by selecting one of the two filtered synchronization inputs (TR1 or TR2). Moreover, the outputs can operate continuously by selecting the **Continuous** option.

Setting of the output multiplexers can be done using the serial RS485 or Ethernet interfaces.

12.6. Output protection

The output protection logic is used to prevent the light from getting overheated and thus damaged.

Inside each of the two protection blocks there is an independent state machine comprising a couple of timers. The first timer is used to constrain the turn-on time of the light (T_{on}) to be lesser than or equal to a programmable value **TonMAX**. The second timer is used to constrain the turn-off time of the light (T_{off}) to be greater than or equal to a programmable value **ToffMIN**.

The diagram in *Figure 7: turn-on and turn-off times within limits* shows what happens when both time constraints are satisfied. As visible in the diagram, the output follows the input.

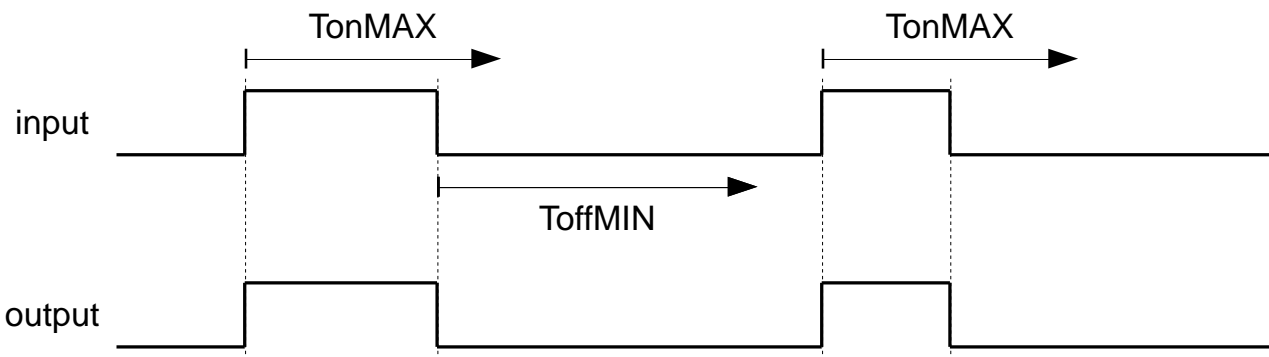


Figure 7: turn-on and turn-off times within limits

The diagram in *Figure 8: protection prevents too long turn-on time* shows what happens when the turn-on time is too long. As visible in the diagram, the light is switched off at **TonMAX**, earlier than the original requirement.

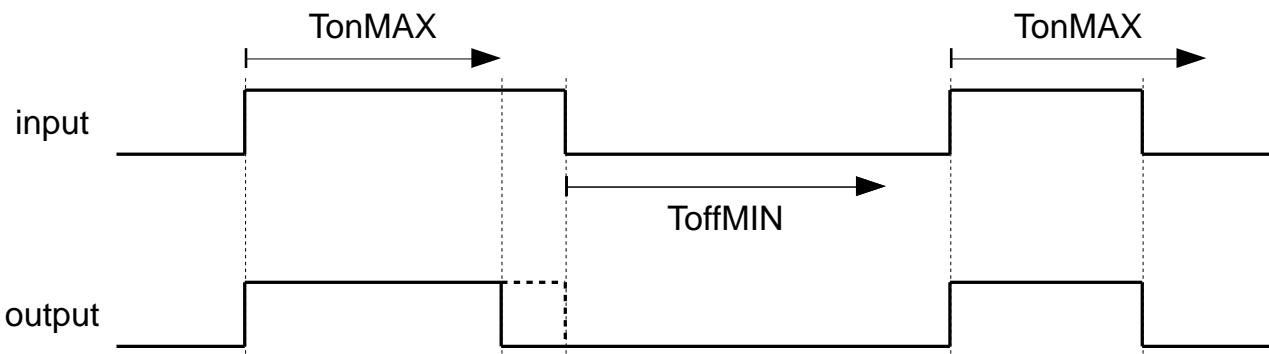


Figure 8: protection prevents too long turn-on time

The diagram in *Figure 9: protection prevents too short turn-off time* shows what happens when turn-off time is too short. As visible in the diagram, the light is switched on at **ToffMIN**, later than the original requirement.

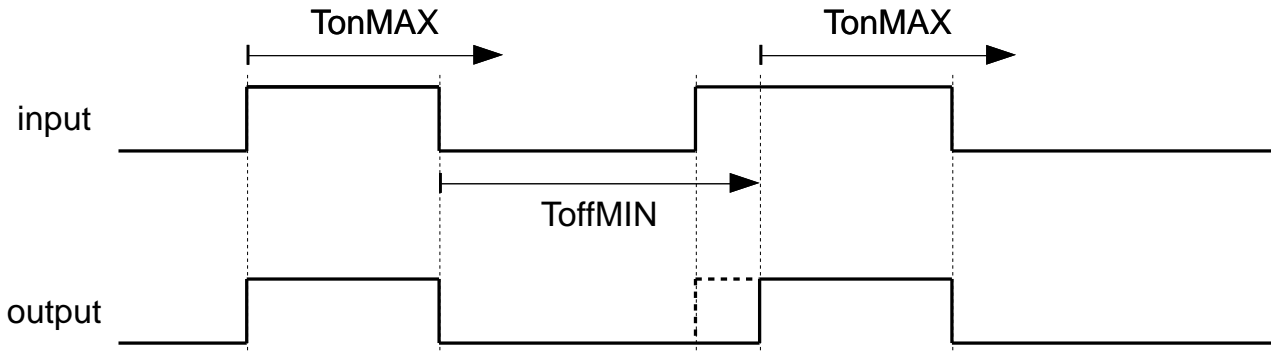


Figure 9: protection prevents too short turn-off time

The two protection blocks are completely independent regarding to **TonMAX** (maximum turn-on time) and **ToffMIN** (minimum turn-off time). All of these time intervals can be programmed individually and may range from 1 ms to 255 ms in steps of 1 ms.

The turn-on protection can be inhibited on select outputs if a continuous operation is required by the application. Similarly, the turn-off protection can be inhibited on select outputs if required by the application.

Setting of the output protection logic can be done using the serial RS485 or Ethernet interfaces.

12.7. Free running oscillator

The free running oscillator is an autonomous asynchronous trigger source with a programmable period from 10 ms up to 1000 ms in steps of 1 ms (corresponding to a frequency of 100 Hz down to 1 Hz). It can be selected as an input to the input multiplexers.

Common usage of the oscillator is to test the lights during machine assembly and deployment.

13. Wiring diagrams

As discussed in the previous sections, the controller is quite flexible and many configurations can be achieved. The following wiring diagrams describe some of the most common.

13.1. Wiring example #1: controller triggers camera

In the schematic diagram of *Figure 10: example schematic #1* the controller is driven by two input triggers, powers a total of two lights and triggers a camera.

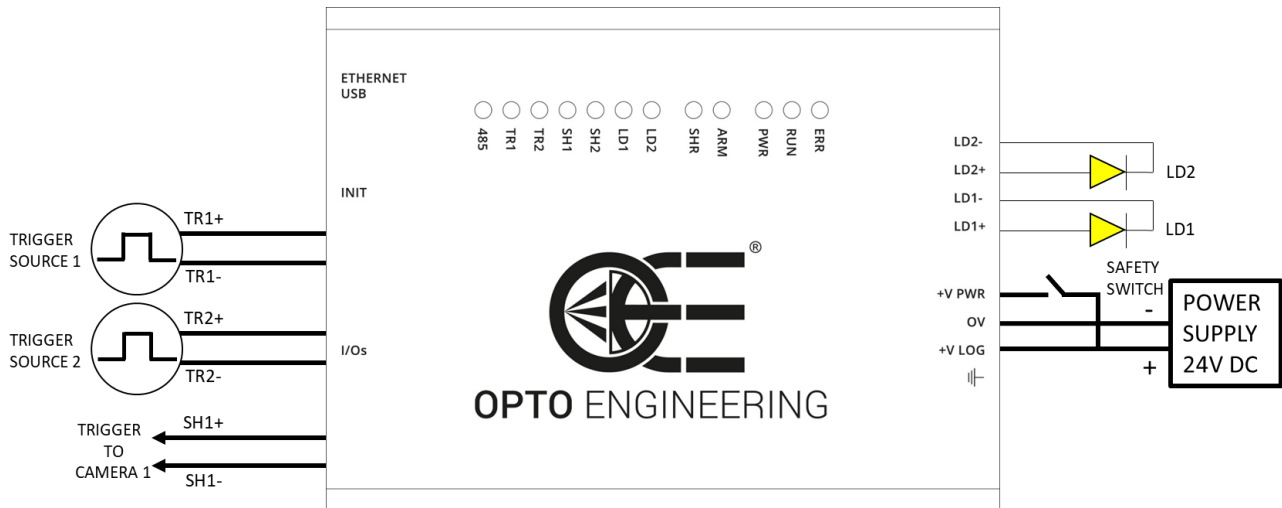


Figure 10: example schematic #1

As shown, the power and logic supplies are derived from a common power supply.

If required by the application, a safety switch may be included in the circuit to cut off supply to the power stages in order to protect the end user from photo-biological hazard. That switch would be appropriately placed on the machine chassis.

The camera is triggered by the controller using one of the two available synchronization outputs. Generally, it is not possible to provide the details of the connections to the cameras because these are often vendor specific. Please see the camera hardware manual for more information.

13.2. Wiring example #2: camera triggers controller

In the schematic diagram of *Figure 11: example schematic #2* the controller is driven by one input trigger coming from a camera, powers a total of two lights and triggers a second camera.

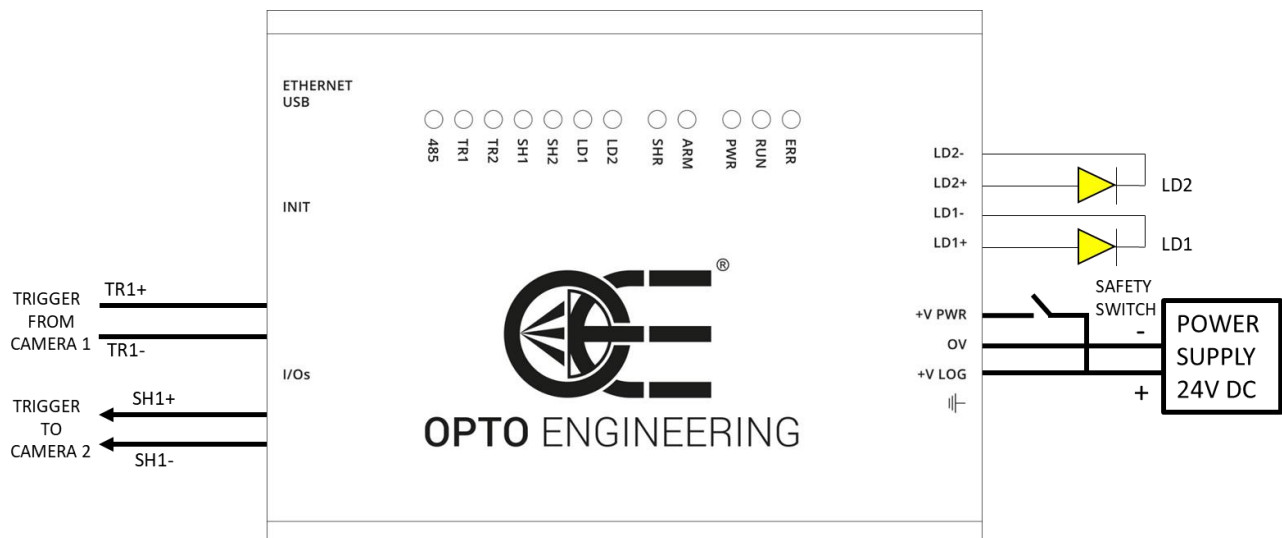


Figure 11: example schematic #2

As shown, the power and logic supplies are derived from a common power supply.

If required by the application, a safety switch may be included in the circuit to cut off supply to the

power stages in order to protect the end user from photo-biological hazard. That switch would be appropriately placed on the machine chassis.

The controller is triggered by camera 1 using one of the two available synchronization inputs. Camera 2 is triggered by the controller using one of the two available synchronization outputs. Generally, it is not possible to provide the details of the connections to the cameras because these are often vendor specific. Please see the camera hardware manual for more information.

14. Operation

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 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 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 Modbus/RTU, Modbus/TCP and Modbus/UDP protocols is given.

14.1. 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>

14.1.1. 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.

14.1.2. 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 13*:

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 13: 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.

14.1.3. 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.

14.1.4. 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.

14.1.5. 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.

14.2. Register file

The list in *Table 14: 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	BOOT_VERSION	R	0-65535	Bootloader firmware version
2	MCU_VERSION	R	0-65535	Microcontroller firmware version
3	FPGA_VERSION	R	0-65535	FPGA firmware version
4	BOARD_VERSION	R	0-65535	Board version
5	OSC_PERIOD	RW	10-1000	Period of the internal oscillator
6	FILTER_SEL0	RW	0-6	Filter setting for input TR1
7	FILTER_SEL1	RW	0-6	Filter setting for input TR2
8	UNUSED	N/A	N/A	
9	UNUSED	N/A	N/A	
10	UNUSED	N/A	N/A	
11	UNUSED	N/A	N/A	
12	UNUSED	N/A	N/A	
13	UNUSED	N/A	N/A	
14	INPUT_SEL0	RW	0-8191	Setting of input multiplexer 1
15	INPUT_SEL1	RW	0-8191	Setting of input multiplexer 2
16	INPUT_SEL2	RW	0-8191	Setting of input multiplexer 3
17	INPUT_SEL3	RW	0-8191	Setting of input multiplexer 4
18	UNUSED	N/A	N/A	
19	UNUSED	N/A	N/A	
20	UNUSED	N/A	N/A	
21	UNUSED	N/A	N/A	
22	UNUSED	N/A	N/A	
23	UNUSED	N/A	N/A	
24	UNUSED	N/A	N/A	
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	GEN_DELAY_BASE0	RW	0-3	Pulse delay time base selector for generator 1
31	GEN_DELAY_COUNT0	RW	0-1023	Pulse delay setting for generator 1
32	GEN_WIDTH_BASE0	RW	0-3	Pulse width time base selector for generator 1
33	GEN_WIDTH_COUNT0	RW	1-1023	Pulse width setting for generator 1
34	GEN_DELAY_BASE1	RW	0-3	Pulse delay time base selector for generator 2
35	GEN_DELAY_COUNT1	RW	0-1023	Pulse delay setting for generator 2
36	GEN_WIDTH_BASE1	RW	0-3	Pulse width time base selector for generator 2
37	GEN_WIDTH_COUNT1	RW	1-1023	Pulse width setting for generator 2
38	GEN_DELAY_BASE2	RW	0-3	Pulse delay time base selector for generator 3
39	GEN_DELAY_COUNT2	RW	0-1023	Pulse delay setting for generator 3
40	GEN_WIDTH_BASE2	RW	0-3	Pulse width time base selector for generator 3
41	GEN_WIDTH_COUNT2	RW	1-1023	Pulse width setting for generator 3
42	GEN_DELAY_BASE3	RW	0-3	Pulse delay time base selector for generator 4
43	GEN_DELAY_COUNT3	RW	0-1023	Pulse delay setting for generator 4
44	GEN_WIDTH_BASE3	RW	0-3	Pulse width time base selector for generator 4
45	GEN_WIDTH_COUNT3	RW	1-1023	Pulse width setting for generator 4
46	UNUSED	N/A	N/A	
47	UNUSED	N/A	N/A	
48	UNUSED	N/A	N/A	
49	UNUSED	N/A	N/A	
50	UNUSED	N/A	N/A	
51	UNUSED	N/A	N/A	
52	UNUSED	N/A	N/A	
53	UNUSED	N/A	N/A	

54	UNUSED	N/A	N/A	
55	UNUSED	N/A	N/A	
56	UNUSED	N/A	N/A	
57	UNUSED	N/A	N/A	
58	UNUSED	N/A	N/A	
59	UNUSED	N/A	N/A	
60	UNUSED	N/A	N/A	
61	UNUSED	N/A	N/A	
62	UNUSED	N/A	N/A	
63	UNUSED	N/A	N/A	
64	UNUSED	N/A	N/A	
65	UNUSED	N/A	N/A	
66	UNUSED	N/A	N/A	
67	UNUSED	N/A	N/A	
68	UNUSED	N/A	N/A	
69	UNUSED	N/A	N/A	
70	UNUSED	N/A	N/A	
71	UNUSED	N/A	N/A	
72	UNUSED	N/A	N/A	
73	UNUSED	N/A	N/A	
74	UNUSED	N/A	N/A	
75	UNUSED	N/A	N/A	
76	UNUSED	N/A	N/A	
77	UNUSED	N/A	N/A	
78	UNUSED	N/A	N/A	
79	UNUSED	N/A	N/A	
80	UNUSED	N/A	N/A	
81	UNUSED	N/A	N/A	
82	UNUSED	N/A	N/A	
83	UNUSED	N/A	N/A	
84	UNUSED	N/A	N/A	
85	UNUSED	N/A	N/A	
86	UNUSED	N/A	N/A	
87	UNUSED	N/A	N/A	

88	UNUSED	N/A	N/A	
89	UNUSED	N/A	N/A	
90	UNUSED	N/A	N/A	
91	UNUSED	N/A	N/A	
92	UNUSED	N/A	N/A	
93	UNUSED	N/A	N/A	
94	OUTPUT_SEL_HI0	RW	0-511	High order byte of setting for LD1 output multiplexer
95	OUTPUT_SEL_LO0	RW	0-65535	Low order byte of setting for LD1 output multiplexer
96	OUTPUT_SEL_HI1	RW	0-511	High order byte of setting for LD2 output multiplexer
97	OUTPUT_SEL_LO1	RW	0-65535	Low order byte of setting for LD2 output multiplexer
98	UNUSED	N/A	N/A	
99	UNUSED	N/A	N/A	
100	UNUSED	N/A	N/A	
101	UNUSED	N/A	N/A	
102	UNUSED	N/A	N/A	
103	UNUSED	N/A	N/A	
104	UNUSED	N/A	N/A	
105	UNUSED	N/A	N/A	
106	UNUSED	N/A	N/A	
107	UNUSED	N/A	N/A	
108	UNUSED	N/A	N/A	
109	UNUSED	N/A	N/A	
110	OUTPUT_SEL_HI8	RW	0-511	High order byte of setting for SH1 output multiplexer
111	OUTPUT_SEL_LO8	RW	0-65535	Low order byte of setting for SH1 output multiplexer
112	OUTPUT_SEL_HI9	RW	0-511	High order byte of setting for SH2 output multiplexer
113	OUTPUT_SEL_LO9	RW	0-65535	Low order byte of setting for SH2 output multiplexer
114	UNUSED	N/A	N/A	
115	UNUSED	N/A	N/A	
116	UNUSED	N/A	N/A	

117	UNUSED	N/A	N/A	
118	UNUSED	N/A	N/A	
119	UNUSED	N/A	N/A	
120	UNUSED	N/A	N/A	
121	UNUSED	N/A	N/A	
122	UNUSED	N/A	N/A	
123	UNUSED	N/A	N/A	
124	UNUSED	N/A	N/A	
125	UNUSED	N/A	N/A	
126	PRT_CNT_ON0	RW	1-255	Maximum turn-on time for light output LD1
127	PRT_ENA_ON0	RW	0-1	Enable limitation of maximum turn-on time for light output LD1
128	PRT_CNT_OFF0	RW	1-255	Minimum turn-off time for light output LD1
129	PRT_ENA_OFF0	RW	0-1	Enable limitation of minimum turn-off time for light output LD1
130	PRT_CNT_ON1	RW	1-255	Maximum turn-on time for light output LD2
131	PRT_ENA_ON1	RW	0-1	Enable limitation of maximum turn-on time for light output LD2
132	PRT_CNT_OFF1	RW	1-255	Minimum turn-off time for light output LD2
133	PRT_ENA_OFF1	RW	0-1	Enable limitation of minimum turn-off time for light output LD2
134	UNUSED	N/A	N/A	
135	UNUSED	N/A	N/A	
136	UNUSED	N/A	N/A	
137	UNUSED	N/A	N/A	
138	UNUSED	N/A	N/A	
139	UNUSED	N/A	N/A	
140	UNUSED	N/A	N/A	
141	UNUSED	N/A	N/A	
142	UNUSED	N/A	N/A	
143	UNUSED	N/A	N/A	
144	UNUSED	N/A	N/A	
145	UNUSED	N/A	N/A	

146	UNUSED	N/A	N/A	
147	UNUSED	N/A	N/A	
148	UNUSED	N/A	N/A	
149	UNUSED	N/A	N/A	
150	UNUSED	N/A	N/A	
151	UNUSED	N/A	N/A	
152	UNUSED	N/A	N/A	
153	UNUSED	N/A	N/A	
154	UNUSED	N/A	N/A	
155	UNUSED	N/A	N/A	
156	UNUSED	N/A	N/A	
157	UNUSED	N/A	N/A	
158	CUR_RANGE0	RW	0-2	Current range for light output LD1
159	CUR_VALUE0	RW	0-40000	Current value for light output LD1
160	CUR_RANGE1	RW	0-2	Current range for light output LD2
161	CUR_VALUE1	RW	0-20000	Current value for light output LD2
162	UNUSED	N/A	N/A	
163	UNUSED	N/A	N/A	
164	UNUSED	N/A	N/A	
165	UNUSED	N/A	N/A	
166	UNUSED	N/A	N/A	
167	UNUSED	N/A	N/A	
168	UNUSED	N/A	N/A	
169	UNUSED	N/A	N/A	
170	UNUSED	N/A	N/A	
171	UNUSED	N/A	N/A	
172	UNUSED	N/A	N/A	
173	UNUSED	N/A	N/A	
174	RS485_MODBUS_ADDR	RW	1-247	Modbus address for serial RS485 interface
175	RS485_LINE_SPEED	RW	0-7	Speed for serial RS485 interface
176	RS485_LINE_PARITY	RW	0-2	Parity for serial RS485 interface

177	ETH_MAC_ADDR0	R	0-65535	Bytes 0 and 1 of Ethernet MAC address
178	ETH_MAC_ADDR1	R	0-65535	Bytes 2 and 3 of Ethernet MAC address
179	ETH_MAC_ADDR2	R	0-65535	Bytes 4 and 5 of Ethernet MAC address
180	ETH_HOSTNAME0	RW	0-65535	Bytes 0 and 1 of Ethernet host name
181	ETH_HOSTNAME1	RW	0-65535	Bytes 2 and 3 of Ethernet host name
182	ETH_HOSTNAME2	RW	0-65535	Bytes 4 and 5 of Ethernet host name
183	ETH_HOSTNAME3	RW	0-65535	Bytes 6 and 7 of Ethernet host name
184	ETH_HOSTNAME4	RW	0-65535	Bytes 8 and 9 of Ethernet host name
185	ETH_HOSTNAME5	RW	0-65535	Bytes 10 and 11 of Ethernet host name
186	ETH_HOSTNAME6	RW	0-65535	Bytes 12 and 13 of Ethernet host name
187	ETH_HOSTNAME7	RW	0-65535	Bytes 14 and 15 of Ethernet host name
188	ETH_DHCP_ENABLE	RW	0-1	Ethernet DHCP enable/disable flag
189	ETH_IP_ADDR_HI	RW	0-65535	High order word of Ethernet IP address
190	ETH_IP_ADDR_LO	RW	0-65535	Low order word of Ethernet IP address
191	ETH_SUBNET_MASK_HI	RW	0-65535	High order word of Ethernet subnet mask
192	ETH_SUBNET_MASK_LO	RW	0-65535	Low order word of Ethernet subnet mask
193	ETH_DEF_GATEWAY_HI	RW	0-65535	High order word of Ethernet default gateway
194	ETH_DEF_GATEWAY_LO	RW	0-65535	Low order word of Ethernet default gateway
195	ETH_PRI_DNS_HI	RW	0-65535	High order word of Ethernet primary DNS address
196	ETH_PRI_DNS_LO	RW	0-65535	Low order word of Ethernet primary DNS address
197	ETH_SEC_DNS_HI	RW	0-65535	High order word of Ethernet secondary DNS address

198	ETH_SEC_DNS_LO	RW	0-65535	Low order word of Ethernet secondary DNS address
199	ETH_MODBUS_ADDR	RW	1-247	Modbus address for Ethernet interface
200	ETH_MODBUS_TCP_PORT	RW	1-65535	Ethernet port for Modbus/TCP
201	ETH_MODBUS_UDP_PORT	RW	1-65535	Ethernet port for Modbus/UDP
202	WEB_PASSWORD0	RW	0-65535	Bytes 0 and 1 of web password
203	WEB_PASSWORD1	RW	0-65535	Bytes 2 and 3 of web password
204	WEB_PASSWORD2	RW	0-65535	Bytes 4 and 5 of web password
205	WEB_PASSWORD3	RW	0-65535	Bytes 6 and 7 of web password
206	CONVERTER_TEMPERATURE	R	-200 to +1000	Temperature of the internal DC/DC converter
207	DRIVER_TEMPERATURE	R	-200 to +1000	Temperature of the drivers for light outputs LD1 and LD2
208	REMOTE_TEMPERATURE0	R	-200 to +1000	Temperature measured by external thermal sensor 1
209	REMOTE_TEMPERATURE1	R	-200 to +1000	Temperature measured by external thermal sensor 2
210	UNUSED	N/A	N/A	
211	SUPPLY_VOLTAGE	R	0-240	Measured supply voltage
212	MEASURED_CURRENT0	R	0-20000	Measured current for light output LD1
213	MEASURED_CURRENT1	R	0-20000	Measured current for light output LD2
214	UNUSED	N/A	N/A	
215	UNUSED	N/A	N/A	
216	UNUSED	N/A	N/A	
217	UNUSED	N/A	N/A	
218	UNUSED	N/A	N/A	
219	UNUSED	N/A	N/A	
220	MEASURED_VOLTAGE0	R	0-2000	Measured voltage for light output LD1
221	MEASURED_VOLTAGE1	R	0-2000	Measured voltage for light output LD2

222	UNUSED	N/A	N/A	
223	UNUSED	N/A	N/A	
224	UNUSED	N/A	N/A	
225	UNUSED	N/A	N/A	
226	UNUSED	N/A	N/A	
227	UNUSED	N/A	N/A	
228	ERROR_WORD	R	0-65535	Composite error word
229	GEN_HOLD_BASE0	RW	0-3	Hold off time base selector for generator 1
230	GEN_HOLD_COUNT0	RW	0-1023	Hold off setting for generator 1
231	GEN_HOLD_BASE1	RW	0-3	Hold off time base selector for generator 2
232	GEN_HOLD_COUNT1	RW	0-1023	Hold off setting for generator 2
233	GEN_HOLD_BASE2	RW	0-3	Hold off time base selector for generator 3
234	GEN_HOLD_COUNT2	RW	0-1023	Hold off setting for generator 3
235	GEN_HOLD_BASE3	RW	0-3	Hold off time base selector for generator 4
236	GEN_HOLD_COUNT3	RW	0-1023	Hold off setting for generator 4
237	UNUSED	N/A	N/A	
238	UNUSED	N/A	N/A	
239	UNUSED	N/A	N/A	
240	UNUSED	N/A	N/A	
241	UNUSED	N/A	N/A	
242	UNUSED	N/A	N/A	
243	UNUSED	N/A	N/A	
244	UNUSED	N/A	N/A	
245	UNUSED	N/A	N/A	
246	UNUSED	N/A	N/A	
247	UNUSED	N/A	N/A	
248	UNUSED	N/A	N/A	
249	UNUSED	N/A	N/A	
250	UNUSED	N/A	N/A	

251	UNUSED	N/A	N/A	
252	UNUSED	N/A	N/A	
253	UNUSED	N/A	N/A	
254	UNUSED	N/A	N/A	
255	UNUSED	N/A	N/A	
256	UNUSED	N/A	N/A	
257	UNUSED	N/A	N/A	
258	UNUSED	N/A	N/A	
259	UNUSED	N/A	N/A	
260	UNUSED	N/A	N/A	
261	GEN_EDGE_SEL0	RW	0-2	Edge selector for generator 1
262	GEN_EDGE_SEL1	RW	0-2	Edge selector for generator 2
263	GEN_EDGE_SEL2	RW	0-2	Edge selector for generator 3
264	GEN_EDGE_SEL3	RW	0-2	Edge selector for generator 4
265	UNUSED	N/A	N/A	
266	UNUSED	N/A	N/A	
267	UNUSED	N/A	N/A	
268	UNUSED	N/A	N/A	
269	UNUSED	N/A	N/A	
270	UNUSED	N/A	N/A	
271	UNUSED	N/A	N/A	
272	UNUSED	N/A	N/A	
273	UNUSED	N/A	N/A	
274	UNUSED	N/A	N/A	
275	UNUSED	N/A	N/A	
276	UNUSED	N/A	N/A	
277	POLARITY_SEL0	N/A	N/A	Polarity selector for input TR1
278	POLARITY_SEL1	N/A	N/A	Polarity selector for input TR2
279	RESERVED	N/A	N/A	Reserved for future use
280	RESERVED	N/A	N/A	Reserved for future use
281	RESERVED	N/A	N/A	Reserved for future use
282	RESERVED	N/A	N/A	Reserved for future use
283	RESERVED	N/A	N/A	Reserved for future use
284	RESERVED	N/A	N/A	Reserved for future use

285	DRIVE_TIME0	R	0-60000	Measured driving time for light output LD1
286	DRIVE_TIME1	R	0-60000	Measured driving timer for light output LD2
287	UNUSED	N/A	N/A	
288	UNUSED	N/A	N/A	
289	UNUSED	N/A	N/A	
290	UNUSED	N/A	N/A	
291	UNUSED	N/A	N/A	
292	UNUSED	N/A	N/A	
293	CUR_RED_DELAY0	RW	0-60000	Current reduction delay for light output LD1
294	CUR_RED_VALUE0	RW	0-100	Current reduction percentage for light output LD1
295	CUR_RED_DELAY1	RW	0-60000	Current reduction delay for light output LD2
296	CUR_RED_VALUE1	RW	0-100	Current reduction percentage for light output LD2
297	UNUSED	N/A	N/A	
298	UNUSED	N/A	N/A	
299	UNUSED	N/A	N/A	
300	UNUSED	N/A	N/A	
301	UNUSED	N/A	N/A	
302	UNUSED	N/A	N/A	
303	UNUSED	N/A	N/A	
304	UNUSED	N/A	N/A	
305	UNUSED	N/A	N/A	
306	UNUSED	N/A	N/A	
307	UNUSED	N/A	N/A	
308	UNUSED	N/A	N/A	
309	SOFTWARE_TRIGGER	RW	0-15	Software trigger
310	CONVERTER_MODE	RW	0-1	Operating mode of DC/DC converter
311	CONVERTER_MAXIMUM_VOLTAGE	RW	0-200	Maximum output voltage of DC/DC converter
312	CONVERTER_PRESET_VOLTAGE	R	0-200	Preset output voltage of DC/DC converter
313	DRIVER_MODE	RW	0-1	Operating mode of LD1 and LD2

				light output drivers
314	DIGIPOT_VALUE	R	0-255	Digital control setting of DC/DC converter
315	CONVERTER_VOLTAGE	R	0-2000	Measured DC/DC converter output voltage
316	DRIVER_VOLTAGE	R	0-2000	Measured LD1 and LD2 light output drivers voltage
317	SUPPLY_CURRENT	R	0-20000	Measured supply current
318	CONVERTER_CURRENT	R	0-20000	Measured DC/DC converter output current
319	CAL_GEN_CUR_LOW_MUL0	R	0-65535	Calibration constant
320	CAL_GEN_CUR_LOW_ADD0	R	0-65535	Calibration constant
321	CAL_GEN_CUR_LOW_MUL1	R	0-65535	Calibration constant
322	CAL_GEN_CUR_LOW_ADD1	R	0-65535	Calibration constant
323	CAL_GEN_CUR_HIGH_MUL0	R	0-65535	Calibration constant
324	CAL_GEN_CUR_HIGH_ADD0	R	0-65535	Calibration constant
325	CAL_GEN_CUR_HIGH_MUL1	R	0-65535	Calibration constant
326	CAL_GEN_CUR_HIGH_ADD1	R	0-65535	Calibration constant
327	CAL_MES_CUR_LOW_MUL0	R	0-65535	Calibration constant
328	CAL_MES_CUR_LOW_ADD0	R	0-65535	Calibration constant
329	CAL_MES_CUR_LOW_MUL1	R	0-65535	Calibration constant
330	CAL_MES_CUR_LOW_ADD1	R	0-65535	Calibration constant
331	CAL_MES_CUR_HIGH_MUL0	R	0-65535	Calibration constant
332	CAL_MES_CUR_HIGH_ADD0	R	0-65535	Calibration constant
333	CAL_MES_CUR_HIGH_MUL1	R	0-65535	Calibration constant
334	CAL_MES_CUR_HIGH_ADD1	R	0-	Calibration constant

			65535	
335	CAL_MES_VLT_LOW_MUL0	R	0-65535	Calibration constant
336	CAL_MES_VLT_LOW_ADD0	R	0-65535	Calibration constant
337	CAL_MES_VLT_LOW_MUL1	R	0-65535	Calibration constant
338	CAL_MES_VLT_LOW_ADD1	R	0-65535	Calibration constant
339	CAL_MES_VLT_HIGH_MUL0	R	0-65535	Calibration constant
340	CAL_MES_VLT_HIGH_ADD0	R	0-65535	Calibration constant
341	CAL_MES_VLT_HIGH_MUL1	R	0-65535	Calibration constant
342	CAL_MES_VLT_HIGH_ADD1	R	0-65535	Calibration constant
343	CAL_SET_CNV_POT_MUL	R	0-65535	Calibration constant
344	CAL_SET_CNV_POT_ADD	R	0-65535	Calibration constant
345	CAL_MES_TMP_CNV_MUL	R	0-65535	Calibration constant
346	CAL_MES_TMP_CNV_ADD	R	0-65535	Calibration constant
347	CAL_MES_TMP_DRV_MUL	R	0-65535	Calibration constant
348	CAL_MES_TMP_DRV_ADD	R	0-65535	Calibration constant
349	CAL_MES_TMP_REM0_MUL	R	0-65535	Calibration constant
350	CAL_MES_TMP_REM0_ADD	R	0-65535	Calibration constant
352	CAL_MES_TMP_REM1_MUL	R	0-65535	Calibration constant
352	CAL_MES_TMP_REM1_ADD	R	0-65535	Calibration constant
353	CAL_MES_VLT_SUP_MUL	R	0-65535	Calibration constant
354	CAL_MES_VLT_SUP_ADD	R	0-65535	Calibration constant
355	CAL_MES_VLT_CNV_MUL	R	0-	Calibration constant

			65535	
356	CAL_MES_VLT_CNV_ADD	R	0-65535	Calibration constant
357	CAL_MES_VLT_DRV_MUL	R	0-65535	Calibration constant
358	CAL_MES_VLT_DRV_ADD	R	0-65535	Calibration constant
359	CAL_MES_CUR_SUP_MUL	R	0-65535	Calibration constant
360	CAL_MES_CUR_SUP_ADD	R	0-65535	Calibration constant
361	CAL_MES_CUR_DRV_MUL	R	0-65535	Calibration constant
362	CAL_MES_CUR_DRV_ADD	R	0-65535	Calibration constant
363	CAL_UNLOCK_CODE0	R	0-65535	Unlock code 0 for the calibration constants
364	CAL_UNLOCK_CODE1	R	0-65535	Unlock code 1 for the calibration constants
365-510	RESERVED	N/A	N/A	Reserved for future use
511	BOARD_COMMAND	RW	0-3	Board command

Table 14: 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.

14.2.1. Register **DEVICE_TYPE**

This register contains the device type. This information is encoded as a 16-bit unsigned number. For the standard LTDVE2CH-20F the device type is 0x000A.

14.2.2. Register **BOOT_VERSION**

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

14.2.3. Register MCU_VERSION

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

14.2.4. Register FPGA_VERSION

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

14.2.5. Register BOARD_VERSION

This register contains the board version. This information is encoded as a 16-bit unsigned number.

14.2.6. Register OSC_PERIOD

Bits [9:0] of this register contain the period of the internal oscillator. The information is expressed in ms. Allowed values are in the range from 10 (corresponding to 100 Hz) up to 1000 (corresponding to 1 Hz). Default value is 200 (corresponding to 5 Hz). Avoid operation with non-allowed values.

Bit field [15:10] of this register is unused. When writing these bits, they must be set to zero.

14.2.7. Registers FILTER_SEL[0-1]

Each bit field [2:0] of these two registers selects the time constant for filtering the relevant input signal.

- **FILTER_SEL0**: filter setting for input TR1
- **FILTER_SEL1**: filter setting for input TR2

Allowed values are in the range from 0 to 6 and are listed below. Avoid operation with non-listed values.

- When 0x0 filter is disabled (pass through) (default value)
- When 0x1 filter is enabled with a 10 μ s time constant
- When 0x2 filter is enabled with a 20 μ s time constant
- When 0x3 filter is enabled with a 50 μ s time constant
- When 0x4 filter is enabled with a 100 μ s time constant
- When 0x5 filter is enabled with a 200 μ s time constant
- When 0x6 filter is enabled with a 500 μ s time constant

Bit fields [15:3] of these registers are unused. When writing these bits, they must be set to zero.

14.2.8. Registers INPUT_SEL[0-3]

Each bit field [11:0] of these four registers is the selector of the relevant input multiplexer. Each input multiplexer feeds a dedicated pulse generator.

- **INPUT_SEL0**: setting of input multiplexer 0
- **INPUT_SEL1**: setting of input multiplexer 1
- **INPUT_SEL2**: setting of input multiplexer 2
- **INPUT_SEL3**: setting of input multiplexer 3

Allowed values are listed below. Avoid operation with non-listed values.

- When 0x0000 the input multiplexer is disabled (default value)
- When 0x0001 the filtered TR1 input is selected
- When 0x0002 the filtered TR2 input is selected
- When 0x0100 the free running oscillator is selected
- When 0x0200 the software trigger SW1 is selected
- When 0x0400 the software trigger SW2 is selected
- When 0x0800 the software trigger SW3 is selected
- When 0x1000 the software trigger SW4 is selected

Bit fields [15:13] and [7:2] of these registers are unused. When writing these bits, they must be set to zero.

14.2.9. Registers **GEN_DELAY_BASE[0-3]**

Each bit field [1:0] of these four registers holds the time base selector for the generation of the pulse delay in the relevant pulse generator.

- **GEN_DELAY_BASE0**: time base selector for generation of pulse delay in generator 1
- **GEN_DELAY_BASE1**: time base selector for generation of pulse delay in generator 2
- **GEN_DELAY_BASE2**: time base selector for generation of pulse delay in generator 3
- **GEN_DELAY_BASE3**: time base selector for generation of pulse delay in generator 4

Allowed values are in the range from 0 to 3 and are listed below. Avoid operation with non-listed values.

- When 0x0 a time base of 1 μ s is selected (default value)
- When 0x1 a time base of 10 μ s is selected
- When 0x2 a time base of 100 μ s is selected
- When 0x3 a time base of 1000 μ s is selected

Bit fields [15:2] of these registers are unused. When writing these bits, they must be set to zero.

14.2.10. Registers **GEN_DELAY_COUNT[0-3]**

Each bit field [9:0] of these four registers holds the actual count for the generation of the pulse delay in the relevant pulse generator.

- **GEN_DELAY_COUNT0**: pulse delay setting for generator 1
- **GEN_DELAY_COUNT1**: pulse delay setting for generator 2
- **GEN_DELAY_COUNT2**: pulse delay setting for generator 3
- **GEN_DELAY_COUNT3**: pulse delay setting for generator 4

Allowed values are in the range from 0 (default value) to 1023 (maximum value). Avoid operation with non-allowed values.

According to the time base selected in register **GEN_DELAY_BASE[x]** and the count set in **GEN_DELAY_COUNT[x]**, the pulse delay may be calculated using the following formula:

$$Delay[x] [\mu s] = value(GEN_DELAY_BASE[x]) * value(GEN_DELAY_COUNT[x])$$

The pulse delay may range from 0 μ s to 1,023,000 μ s with variable absolute resolution.

Bit fields [15:10] of these registers are unused. When writing these bits, they must be set to zero.

14.2.11. Registers **GEN_WIDTH_BASE[0-3]**

Each bit field [1:0] of these four registers holds the time base selector for the generation of the pulse width in the relevant pulse generator.

- **GEN_WIDTH_BASE0**: time base selector for generation of pulse width in generator 1
- **GEN_WIDTH_BASE1**: time base selector for generation of pulse width in generator 2
- **GEN_WIDTH_BASE2**: time base selector for generation of pulse width in generator 3
- **GEN_WIDTH_BASE3**: time base selector for generation of pulse width in generator 4

Allowed values are in the range from 0 to 3 and are listed below. Avoid operation with non-listed values.

- When 0x0 a time base of 1 μ s is selected (default value)
- When 0x1 a time base of 10 μ s is selected
- When 0x2 a time base of 100 μ s is selected
- When 0x3 a time base of 1000 μ s is selected

Bit fields [15:2] of these registers are unused. When writing these bits, they must be set to zero.

14.2.12. Registers **GEN_WIDTH_COUNT[0-3]**

Each bit field [9:0] of these four registers holds the actual count for the generation of the pulse width in the relevant pulse generator.

- **GEN_WIDTH_COUNT0**: pulse width setting for generator 1
- **GEN_WIDTH_COUNT1**: pulse width setting for generator 2
- **GEN_WIDTH_COUNT2**: pulse width setting for generator 3
- **GEN_WIDTH_COUNT3**: pulse width setting for generator 4

Allowed values are in the range from 1 (default value) to 1023 (maximum value). Avoid operation with non-allowed values.

According to the time base selected in register **GEN_WIDTH_BASE[x]** and the count set in **GEN_WIDTH_COUNT[x]**, the pulse width may be calculated using the following formula:

$$Width[x] [\mu s] = value(GEN_WIDTH_BASE[x]) * value(GEN_WIDTH_COUNT[x])$$

The pulse width may range from 1 μ s to 1,023,000 μ s with variable absolute resolution.

Bit fields [15:10] of these registers are unused. When writing these bits, they must be set to zero.

14.2.13. Registers **OUTPUT_SEL_HI[0-1]** and **OUTPUT_SEL_HI[8-9]**

The output multiplexers are used to route the internal signals to the light outputs and synchronization outputs. Each output multiplexer has an independent selector.

The selector of a specific output multiplexer is a 32 bits binary number, split on a pair of contiguous Modbus registers named **OUTPUT_SEL_HI[x]** and **OUTPUT_SEL_LO[x]**.

The **OUTPUT_SEL_HI[x]** registers contain the upper sixteen bits of the selectors, while the **OUTPUT_SEL_LO[x]** registers contain the remaining lower sixteen bits of the selectors.

- **OUT_SEL_HI0**: upper sixteen bits of output multiplexer 1 selector (light LD1)
- **OUT_SEL_HI1**: upper sixteen bits of output multiplexer 2 selector (light LD2)

- **OUT_SEL_HI8**: upper sixteen bits of output multiplexer 9 selector (sync. output SH1)
- **OUT_SEL_HI9**: upper sixteen bits of output multiplexer 10 selector (sync. output SH2)

Allowed values for the selectors of the output multiplexers are listed below. Avoid operation with non-listed values.

- When 0x0000:0000 the output multiplexer is disabled (default value)
- When 0x0000:0001 pulse generator 1 output is selected
- When 0x0000:0002 pulse generator 2 output is selected
- When 0x0000:0004 pulse generator 3 output is selected
- When 0x0000:0008 pulse generator 4 output is selected
- When 0x0001:0000 filtered input TR1 is selected
- When 0x0002:0000 filtered input TR2 is selected
- When 0x0100:0000 the output is always active (continuous)

Bit fields [15:9] and [7:2] of these registers are unused. When writing these bits, they must be set to zero.

14.2.14. Registers **OUTPUT_SEL_LO[0-1]** and **OUTPUT_SEL_LO[8-9]**

The output multiplexers are used to route the internal signals to the light outputs and synchronization outputs. Each output multiplexer has an independent selector.

The selector of a specific output multiplexer is a 32 bits binary number, split on a pair of contiguous Modbus registers named **OUTPUT_SEL_HI[x]** and **OUTPUT_SEL_LO[x]**.

The **OUTPUT_SEL_HI[x]** registers contain the upper sixteen bits of the selectors, while the **OUTPUT_SEL_LO[x]** registers contain the remaining lower sixteen bits of the selectors.

- **OUT_SEL_LO0**: lower sixteen bits of output multiplexer 1 selector (light LD1)
- **OUT_SEL_LO1**: lower sixteen bits of output multiplexer 2 selector (light LD2)
- **OUT_SEL_LO8**: lower sixteen bits of output multiplexer 9 selector (sync. output SH1)
- **OUT_SEL_LO9**: lower sixteen bits of output multiplexer 10 selector (sync. output SH2)

Allowed values for the selectors of the output multiplexers are listed in [section 14.2.13](#), along with a description of the **OUTPUT_SEL_HI[x]** registers. Avoid operation with non-listed values.

Bit fields [15:4] of these registers are unused. When writing these bits, they must be set to zero.

14.2.15. Registers **PRT_CNT_ON[0-1]**

Each bit field [9:0] of these two registers holds the maximum turn-on time for light outputs LD1 and LD2.

- **PRT_CNT_ON0**: maximum turn-on time for light output LD1
- **PRT_CNT_ON1**: maximum turn-on time for light output LD2

Allowed values are in the range from 1 (default value) to 1023 (maximum value) and are expressed in ms, so the maximum turn-on time can range from 1 ms to 1023 ms with a resolution of 1 ms. Avoid operation with non-allowed values.

Bit fields [15:10] of these registers are unused. When writing these bits, they must be set to zero.

14.2.16. Registers PRT_ENA_ON[0-1]

Bit 0 of these two registers holds the enable flag for the limitation of turn-on time for light outputs LD1 and LD2.

- **PRT_ENA_ON0**: enable flag for the limitation of turn-on time for light output LD1
- **PRT_ENA_ON1**: enable flag for the limitation of turn-on time for light output LD2

When set to zero the limitation of turn-on time is disabled (default value), when set to one the limitation of turn-on time is enabled.

Bit fields [15:1] of these registers are unused. When writing these bits, they must be set to zero.

14.2.17. Registers PRT_CNT_OFF[0-1]

Each bit field [9:0] of these two registers holds the minimum turn-off time for light outputs LD1 and LD2.

- **PRT_CNT_OFF0**: minimum turn-off time for light output LD1
- **PRT_CNT_OFF1**: minimum turn-off time for light output LD2

Allowed values are in the range from 1 (default value) to 1023 (maximum value) and are expressed in ms, so the minimum turn-off time can range from 1 ms to 1023 ms with a resolution of 1 ms. Avoid operation with non-allowed values.

Bit fields [15:10] of these registers are unused. When writing these bits, they must be set to zero.

14.2.18. Registers PRT_ENA_OFF[0-1]

Bit 0 of these two registers holds the enable flag for the limitation of turn-off time for light outputs LD1 and LD2.

- **PRT_ENA_OFF0**: enable flag for the limitation of turn-off time for light output LD1
- **PRT_ENA_OFF1**: enable flag for the limitation of turn-off time for light output LD2

When set to zero the limitation of turn-off time is disabled (default value), when set to one the limitation of turn-off time is enabled.

Bit fields [15:1] of these registers are unused. When writing these bits, they must be set to zero.

14.2.19. Registers CUR_RANGE[0-1]

Each bit field [1:0] of these two registers selects the current range for the relevant light output.

- **CUR_RANGE0**: current range setting for light output LD1
- **CUR_RANGE1**: current range setting for light output LD2

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

- When 0x0 the automatic current range mode is selected
- When 0x1 the low current range (zero up to 250 mA if light outputs are independent, or zero up to 500 mA if light outputs are shared) is selected
- When 0x2 the high current range (zero up to 20000 mA if light outputs are independent, or zero up to 40000 mA if light outputs are shared) is selected

Bit fields [15:2] of these registers are unused. When writing these bits, they must be set to zero.

14.2.20. Registers CUR_VALUE[0-1]

Each of these two registers defines the current value for the relevant light output.

- **CUR_VALUE0**: current value setting for light output LD1
- **CUR_VALUE1**: current value setting for light output LD2

Allowed values for register **CUR_VALUE0** are in the range from zero to 20000 if light outputs are independent, or zero up to 40000 if light outputs are shared, and are expressed in mA. Allowed values for register **CUR_VALUE1** are in the range zero up to 20000 and are expressed in mA. Avoid operation with non-allowed values.

The effective resolution of the generated current depends on the range the required current falls in. Please see [section 4.2](#) for more information.

14.2.21. 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).

14.2.22. Register RS485_LINE_SPEED

Each 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 0x0 the selected baud rate is 1200 bits per second
- When 0x1 the selected baud rate is 2800 bits per second
- When 0x2 the selected baud rate is 4800 bits per second
- When 0x3 the selected baud rate is 9600 bits per second (default value)
- When 0x4 the selected baud rate is 19200 bits per second
- When 0x5 the selected baud rate is 38400 bits per second
- When 0x6 the selected baud rate is 57600 bits per second
- When 0x7 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 a maximum of 9600 bits per second. Setting a higher rate in this register will cause the serial interface to operate at 9600 bits per second.

14.2.23. Register RS485_LINE_PARITY

Each 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 0x0 there is no parity bit
- When 0x1 the parity bit is even (default value)

- When 0x2 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).

14.2.24. 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).

14.2.25. 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).

14.2.26. Register ETH_DHCP_ENABLE

Bit 0 of this register 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.

14.2.27. 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).

14.2.28. 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).

14.2.29. 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).

14.2.30. 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).

14.2.31. 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).

14.2.32. 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).

14.2.33. 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).

14.2.34. 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).

14.2.35. 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).

14.2.36. 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).

14.2.37. 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).

14.2.38. 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).

14.2.39. 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).

14.2.40. 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).

14.2.41. Register CONVERTER_TEMPERATURE

This register contains the measured temperature of the electronic components that implement the DC/DC converter. The temperature is expressed as a signed sixteen bits integer, with a resolution of 0.1 °C. Given a register value, the corresponding actual temperature can be calculated as follows:

$$\text{ConverterTemperature [}^{\circ}\text{C]} = \text{value}(\text{CONVERTER_TEMPERATURE}) * 0.1$$

This register is continuously updated. This register is read only.

14.2.42. Register DRIVER_TEMPERATURE

This register contains the measured temperature of the electronic components that implement the output drivers. The temperature is expressed as a signed sixteen bits integer, with a resolution of 0.1 °C. Given a register value, the corresponding actual temperature can be calculated as follows:

$$DriverTemperature [^{\circ}C] = value(DRIVER_TEMPERATURE) * 0.1$$

This register is continuously updated. This register is read only.

14.2.43. Registers REMOTE_TEMPERATURE[0-1]

The controller allows for the connection of two external temperature sensors (see [section 8.4.4](#) for more information).

These registers contain the measured temperatures, expressed as signed sixteen bits integers, with a resolution of 0.1 °C. Given the register value, the corresponding actual temperature for the remote sensor can be calculated as follows:

$$RemoteTemperature[x] [^{\circ}C] = value(REMOTE_TEMPERATURE[x]) * 0.1$$

In case an external temperature sensor is not connected, the value in the corresponding register is not defined. These registers are continuously updated. These registers are read only.

14.2.44. Register SUPPLY_VOLTAGE

The controller is capable of measuring the actual incoming supply voltage on the +V PWR wire.

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 for the power stages can be calculated as follows:

$$SupplyVoltage [V] = value(SUPPLY_VOLTAGE) * 0.1$$

The measured value drops to zero if the supply on the +V PWR wire is removed. The register is continuously updated and the value is approximate. This register is read only.

14.2.45. Registers MEASURED_CURRENT[0-1]

The controller is capable of measuring the actual output currents for the two light outputs.

- **MEASURED_CURRENT0**: measured current for light output LD1
- **MEASURED_CURRENT1**: measured current for light output LD2

These registers contain the measured output currents, expressed as unsigned sixteen bits integers, with a resolution of 1 mA. Given a register value, the corresponding actual current for a light output can be calculated as follows:

$$MeasuredCurrent[x] [A] = value(MEASURED_CURRENT[x]) * 0.001$$

These registers are updated as long as the outputs are activated. The registers stop being updated when the outputs turn off. These registers are read only.

14.2.46. Registers MEASURED_VOLTAGE[0-1]

The controller is capable of measuring the actual output voltages for the two light outputs.

- **MEASURED_VOLTAGE0**: measured voltage for light output LD1
- **MEASURED_VOLTAGE1**: measured voltage for light output LD2

These registers contain the measured output voltages, expressed as unsigned sixteen bits integers, with a resolution of 0.1 V. Given a register value, the corresponding actual voltage for a light output

can be calculated as follows:

$$\text{MeasuredVoltage}[x] [V] = \text{value}(\text{MEASURED_VOLTAGE}[x]) * 0.1$$

These registers are updated as long as the outputs are activated. The registers stop being updated when the outputs turn off. These registers are read only.

14.2.47. 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: set if supply on the +V PWR wire is missing, reset if supply is present
- Bit 1: set if excessive temperature on the DC/DC converter, reset if safe temperature
- Bit 2: set if excessive temperature on the output drivers, reset if safe temperature
- Bit 3: set if bad reading or excessive temperature on remote thermal sensor 1, reset if good reading and safe temperature
- Bit 4: set if bad reading or excessive temperature on remote thermal sensor 2, reset if good reading and safe temperature
- Bit 5: set if LD1 output driver has been disabled, reset if LD1 output driver is active
- Bit 6: set if LD2 output driver has been disabled, reset if LD2 output driver is active

When the thermal sensor on the DC/DC converter reports an excessive temperature, the relevant error bit (1) is set and the output drivers are shut down. This happens when the measured temperature goes above about 90 °C. The output drivers are then reactivated when the temperature goes below about 80 °C.

When the thermal sensor on the output drivers reports an excessive temperature, the relevant error bit (2) is set and the output drivers are shut down. This happens when the measured temperature goes above about 90 °C. The output drivers are then reactivated when the temperature goes below about 80 °C.

When a hardware problem is detected on any of the two output channels, the relevant error bit (5 or 6) is set and the relevant output driver is permanently disabled. The error condition may be reset by cycling the power supply off and on or by issuing a reboot command to the controller (see [chapter 14.2.52](#) for more information)

When the supply on the +V PWR wire is missing, the ERR LED is lit with a stable red colour. When an over temperature condition or a hardware problem is detected the ERR LED is lit with a blinking red colour.

Synchronization outputs SH1 and SH2 are never disabled.

The upper bits of this register are unused.

14.2.48. Registers **GEN_HOLD_BASE[0-3]**

Each bit field [1:0] of these four registers holds the time base selector for the generation of the hold off interval in the relevant pulse generator.

- **GEN_HOLD_BASE0**: time base selector for generation of hold off interval in generator 1
- **GEN_HOLD_BASE1**: time base selector for generation of hold off interval in generator 2
- **GEN_HOLD_BASE2**: time base selector for generation of hold off interval in generator 3
- **GEN_HOLD_BASE3**: time base selector for generation of hold off interval in generator 4

Allowed values are in the range from 0 to 3 and are listed below. Avoid operation with non-listed

values.

- When 0x0 a time base of 1 μ s is selected (default value)
- When 0x1 a time base of 10 μ s is selected
- When 0x2 a time base of 100 μ s is selected
- When 0x3 a time base of 1000 μ s is selected

Bit fields [15:2] of these registers are unused. When writing these bits, they must be set to zero.

14.2.49. Registers GEN_HOLD_COUNT[0-3]

Each bit field [9:0] of these four registers holds the actual count for the generation of the hold off interval in the relevant pulse generator.

- **GEN_HOLD_COUNT0**: hold off interval setting for generator 1
- **GEN_HOLD_COUNT1**: hold off interval setting for generator 2
- **GEN_HOLD_COUNT2**: hold off interval setting for generator 3
- **GEN_HOLD_COUNT3**: hold off interval setting for generator 4

Allowed values are in the range from 0 (default value) to 1023 (maximum value). Avoid operation with non-allowed values.

According to the time base selected in register **GEN_HOLD_BASE[x]** and the count set in **GEN_HOLD_COUNT[x]**, the hold off interval may be calculated using the following formula:

$$\text{HoldOff}[x] [\mu\text{s}] = \text{value}(\text{GEN_HOLD_BASE}[x]) * \text{value}(\text{GEN_HOLD_COUNT}[x])$$

The hold off interval may range from 0 μ s to 1,023,000 μ s with variable absolute resolution.

Bit fields [15:10] of these registers are unused. When writing these bits, they must be set to zero.

14.2.50. Registers GEN_EDGE_SEL[0-3]

Each bit field [1:0] of these four registers holds the trigger edge selector for the relative pulse generator.

- **GEN_EDGE_SEL0**: trigger edge selector for generator 1
- **GEN_EDGE_SEL1**: trigger edge selector for generator 2
- **GEN_EDGE_SEL2**: trigger edge selector for generator 3
- **GEN_EDGE_SEL3**: trigger edge selector for generator 4

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

- When 0x0 the generator is triggered by a rising edge
- When 0x1 the generator is triggered by a falling edge
- When 0x2 the generator is triggered by both a rising and a falling edge

Bit fields [15:2] of these registers are unused. When writing these bits, they must be set to zero.

14.2.51. Registers POLARITY_SEL[0-1]

Bit 0 of these two registers holds the polarity selector for the relative trigger input.

- **POLARITY_SEL0**: polarity selector for trigger input TR1
- **POLARITY_SEL1**: polarity selector for trigger input TR2

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

- When 0x0 polarity is normal (default value)
- When 0x1 polarity is inverted

Bit fields [15:1] of these registers are unused. When writing these bits, they must be set to zero.

14.2.52. Registers **DRIVE_TIME[0-1]**

The controller is capable of measuring the actual driving time for the two light outputs.

- **DRIVE_TIME0**: driving time for light output LD1
- **DRIVE_TIME1**: driving time for light output LD2

These registers contain the measured driving times, expressed as unsigned sixteen bits integers, with a resolution of 1 ms. Given a register value, the corresponding actual driving time for a light output can be calculated as follows:

$$\text{DrivingTime}[x] [\text{ms}] = \text{value}(\text{DRIVING_TIME}[x]) * 1.0$$

These registers are updated as long as the outputs are activated. The registers stop being updated when the outputs turn off. Maximum measured value is 65535 ms, for longer driving times 65535 ms is always returned. The values are computed in software, therefore they are approximate and subject to fluctuation. These registers are read only.

14.2.53. Registers **CUR_RED_DELAY[0-1]**

The controller includes a current reduction feature that allows the automatic decrease of the current in the light outputs when they are continuously driven for a time longer than a specified, programmed amount.

Each of these two registers defines the delay time for the activation of the current reduction feature for the relevant light output.

- **CUR_RED_DELAY0**: current reduction delay setting for light output LD1
- **CUR_RED_DELAY1**: current reduction delay setting for light output LD2

Allowed values are in the range from zero to 60000 and are expressed in ms, so the current reduction delay can range from zero up to 60000 ms. A value of zero (factory default) disables the feature.

14.2.54. Registers **CUR_RED_VALUE[0-1]**

The controller includes a current reduction feature that allows the automatic decrease of the current in the light outputs when they are continuously driven for a time longer than a specified, programmed amount.

Each of these two registers defines the reduced current, in terms of percentage of the programmed regular current, for the relevant light output.

- **CUR_RED_VALUE0**: reduced current setting for light output LD1
- **CUR_RED_VALUE1**: reduced current setting for light output LD2

Allowed values are in the range from zero to 100 and are expressed in units of percentage, therefore the reduced current can vary from zero up to the programmed regular current. A value of zero (factory default) selects a reduced current of zero, effectively turning off the light output after the current reduction delay has expired.

14.2.55. Register SOFTWARE_TRIGGER

Bit field [3:0] of this register contains four bits that can be used to issue up to four independent and direct software triggers to the controller internal timing logic. The meaning of the bits is as follows:

- Bit 0 issues software trigger SW1 when set to 1
- Bit 1 issues software trigger SW2 when set to 1
- Bit 2 issues software trigger SW3 when set to 1
- Bit 3 issues software trigger SW4 when set to 1

Multiple software triggers can be issued at once by setting the relevant bits in the same Modbus write of this register. The controller architecture guarantees that the software triggers are propagated in parallel with the same latency. This register is self-clearing: any bits set to 1 are automatically reset back to zero when the software trigger has been processed.

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

14.2.56. Register CONVERTER_MODE

The internal DC/DC converter can either operate in two modes: fixed voltage mode and autodetect voltage mode. The actual operating mode is selected by this register.

In the fixed voltage mode, the output voltage of the DC/DC converter is simply set to the value programmed in the **CONVERTER_MAXIMUM_VOLTAGE** register regardless of the actual voltages developed on the two light outputs when they are driven at the programmed currents. The **CONVERTER_PRESET_VOLTAGE** register is overwritten with the same voltage value.

In the autodetect voltage mode, the output voltage of the DC/DC converter is gradually increased from zero to the value programmed in the **CONVERTER_MAXIMUM_VOLTAGE** register. At each step the lights are flashed for 100 μ s while the actual currents are measured and evaluated. This process is stopped as soon as the programmed current is produced in each of the lights. At the end of the process, the two converter controlling registers **CONVERTER_MAXIMUM_VOLTAGE** and **CONVERTER_PRESET_VOLTAGE** are overwritten with the final voltage value reached on the DC/DC converter.

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

- When 0x0 the DC/DC converter is in the fixed voltage mode (default)
- When 0x1 the DC/DC converted is in the autodetect voltage mode

The factory value for this register is zero, therefore by default the DC/DC converter operates in the fixed voltage mode.

Notes:

- The automatic voltage sensing process is actually triggered by either writing register **BOARD_COMMAND** with value 4 (see section [14.2.66. Register BOARD_COMMAND](#)) or by pressing the **Apply** button in the **Converter settings** section in the **Light outputs** web page (see section [14.3.1. Main page and navigation menu](#))
- The automatic voltage sensing process works by flashing the lights for 100 μ s. This switch-on time is intentionally short in order not to risk damage to the lights. If a longer turn-on time is required in the final application, the output voltage of the DC/DC converter may need to be increased of some amount. Should this be the case, refer to the light current and voltage measurements collected in the **MEASURED_CURRENT[x]** and **MEASURED_VOLTAGE[x]** registers or the actual data shown in the **Main** web page

14.2.57. Register **CONVERTER_MAXIMUM_VOLTAGE**

The internal DC/DC converter can either operate in two modes: fixed voltage mode and autodetect voltage mode.

In the fixed voltage mode, the output voltage of the DC/DC converter is simply set to the value programmed in the **CONVERTER_MAXIMUM_VOLTAGE** register regardless of the actual voltages developed on the two light outputs when they are driven at the programmed currents. The **CONVERTER_PRESET_VOLTAGE** register is overwritten with the same voltage value.

In the autodetect voltage mode, the output voltage of the DC/DC converter is gradually increased from zero to the value programmed in the **CONVERTER_MAXIMUM_VOLTAGE** register. At each step the lights are flashed for 100 μ s while the actual currents are measured and evaluated. This process is stopped as soon as the programmed current is produced in each of the lights. At the end of the process, the two converter controlling registers **CONVERTER_MAXIMUM_VOLTAGE** and **CONVERTER_PRESET_VOLTAGE** are overwritten with the final voltage value reached on the DC/DC converter.

Allowed values for this register are in the range from 0 to 200 and are expressed in volts, so the theoretical maximum DC/DC converter output voltage would range from zero up to 200 V. However, the actual minimum is around 5 V and the actual maximum is about 190 V.

The factory value for this register is 30, therefore by default the maximum voltage produced by the DC/DC converter is 30 V.

Avoid operation with non-allowed values.

14.2.58. Register **CONVERTER_PRESET_VOLTAGE**

The internal DC/DC converter can either operate in two modes: fixed voltage mode and autodetect voltage mode.

In the fixed voltage mode, the output voltage of the DC/DC converter is simply set to the value programmed in the **CONVERTER_MAXIMUM_VOLTAGE** register regardless of the actual voltages developed on the two light outputs when they are driven at the programmed currents. The **CONVERTER_PRESET_VOLTAGE** register is overwritten with the same voltage value.

In the autodetect voltage mode, the output voltage of the DC/DC converter is gradually increased from zero to the value programmed in the **CONVERTER_MAXIMUM_VOLTAGE** register. At each step the lights are flashed for 100 μ s while the actual currents are measured and evaluated. This process is stopped as soon as the programmed current is produced in each of the lights. At the end of the process, the two converter controlling registers **CONVERTER_MAXIMUM_VOLTAGE** and **CONVERTER_PRESET_VOLTAGE** are overwritten with the final voltage value reached on the DC/DC converter.

This register holds the actual DC/DC converter output voltage setting. Values in this register are expressed in volts. This register is read only.

14.2.59. Register **DRIVER_MODE**

The two output drivers in the controller, supplying light outputs LD1 and LD2, can either operate in two modes: independent mode and shared mode. The actual operating mode is selected by this register.

In the independent mode the two light outputs are completely autonomous. In this operating mode the current and timing settings are programmed for each light output using dedicated Modbus registers and dedicated controls on the web interface.

In the shared mode the two output drivers are interconnected. This results in twice the maximum output current than in the independent mode. In the shared mode the controller effectively behaves as a single channel controller. When operating in this mode the light must be connected to output

terminals LD1+ and LD1-, while output terminals LD2+ and LD2- must be left unconnected. In this operating mode the current and timings settings are programmed by Modbus registers and controls on the web interface related to LD1.

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

- When 0x0 the output drivers operate in independent mode (default)
- When 0x1 the output drivers operate in shared mode

The factory value for this register is zero, therefore by default the output drivers operate in independent mode.

14.2.60. Register DIGIPOT_VALUE

This register holds the digital control setting for the DC/DC converter. This register is read only.

14.2.61. Register CONVERTER_VOLTAGE

The controller is capable of measuring the actual voltage produced by the DC/DC converter.

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

$$\text{ConverterVoltage [V]} = \text{value}(\text{CONVERTER_VOLTAGE}) * 0.1$$

The measured value drops to zero if the DC/DC converter is defective. The register is continuously updated and the value is approximate. This register is read only.

14.2.62. Register DRIVER_VOLTAGE

The controller is capable of measuring the actual voltage supplying the drivers of the light outputs.

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

$$\text{DriverVoltage [V]} = \text{value}(\text{DRIVER_VOLTAGE}) * 0.1$$

The measured value drops to zero if the DC/DC converter is defective or the internal fuse protecting the output drivers has blown. The register is continuously updated and the value is approximate. This register is read only.

14.2.63. Register SUPPLY_CURRENT

The controller is capable of measuring the actual current drawn by the DC/DC converter.

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 current drawn by the DC/DC converter can be calculated as follows:

$$\text{SupplyCurrent [mA]} = \text{value}(\text{SUPPLY_CURRENT}) * 1.0$$

This register is continuously updated and the value is approximate. This register is read only.

14.2.64. Register CONVERTER_CURRENT

The controller is capable of measuring the actual current output from the DC/DC converter.

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 current output from the DC/DC converter can be calculated as follows:

$$\text{ConverterCurrent [mA]} = \text{value}(\text{CONVERTER_CURRENT}) * 1.0$$

This register is continuously updated and the value is approximate. This register is read only.

14.2.65. Registers CAL_XXX

These registers contain several calibration constants for current generation, current measurement, voltage measurement and temperature measurement.

Factory values can be overridden but the saving of the new values in non-volatile memory is not allowed unless valid unlock codes are entered in registers **CAL_UNLOCK_CODE0** and **CAL_UNLOCK_CODE1**.

14.2.66. Registers CAL_UNLOCK_CODE[0-1]

These registers contain unlock codes to allow saving in non-volatile memory of the various calibration constants used for current generation, current measurement, voltage measurement and temperature measurement.

14.2.67. 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
- When 4 the voltage autodetect is started if the DC/DC converter operates in autodetect voltage mode (register **CONVERTER_MODE** set to 1). This setting has no effect if the DC/DC converter operates in fixed voltage mode (register **CONVERTER_MODE** set to 0)

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.

14.3. 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 14.1](#) and [chapter 14.2](#).

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.


14.3.1. 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 12: the Main web page*).



Main page

Version information

 OPTO ENGINEERING	Device type:	LTDVE2CH-20F
	Bootloader firmware version:	1.10
	MCU firmware version:	1.10
	FPGA firmware version:	1.10
	Board version:	1.00

Current state

Converter temperature:	24.9 °C	Driver temperature:	24.3 °C
Remote temperature 1:	25.6 °C	Remote temperature 2:	25.4 °C
Supply voltage:	23.7 V	Supply current:	0.18 A
Converter output voltage:	29.8 V	Converter output current:	0.05 A
Driver voltage:	29.8 V		
Measured current LD1:	0.00 A	Measured voltage LD1:	0.0 V
Measured current LD2:	0.00 A	Measured voltage LD2:	0.0 V

Refresh

Figure 12: 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**
- **Setup synch inputs**
- **Setup pulse generators**
- **Setup light outputs**
- **Setup synch outputs**
- **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 MCU bootloader firmware
 - **MCU firmware version:** version of the MCU application firmware
 - **FPGA firmware version:** version of the FPGA firmware
 - **Board version:** version of the hardware board
- **Current state:**
 - **Converter temperature:** temperature of the DC step-up converter
 - **Driver temperature:** average temperature of the internal drivers
 - **Remote temperature 1:** temperature measured by remote sensor 1, if present
 - **Remote temperature 2:** temperature measured by remote sensor 2, if present
 - **Supply voltage:** actual supply voltage of the controller
 - **Supply current:** actual current drawn by the DC/DC converter
 - **Converter output voltage:** actual voltage output from the DC/DC converter
 - **Converter output current:** actual current output from the DC/DC converter
 - **Driver voltage:** actual voltage supplying the output drivers for the lights
 - **Measured current LD1:** measured current on light output LD1
 - **Measured voltage LD1:** measured voltage on light output LD1
 - **Measured current LD2:** measured current on light output LD2
 - **Measured voltage LD2:** measured voltage on light output LD2

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

Light currents and voltages are measured continuously when the outputs are active. Light currents and voltages stop being updated when the outputs turn off. If a measurement is not available, the indication N/A, standing for Not Available, is displayed in the web page.

14.3.2. Setup synch inputs

This page allows to inspect and change all the settings related to the two synchronization inputs. The **Setup synch inputs** page is shown in the image of *Figure 13: the Setup synch inputs web page*.

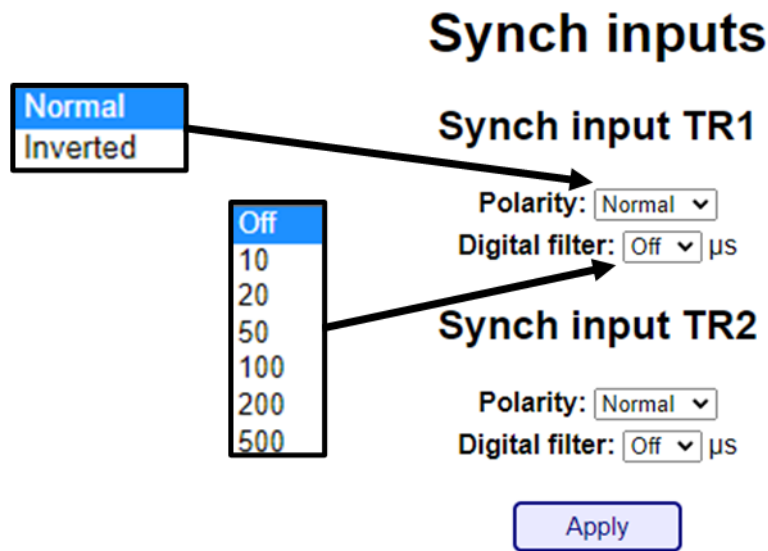


Figure 13: the **Setup synch inputs** web page

Each synchronization input has a programmable polarity control. The available selections are:

- Normal: polarity is active high (default value)
- Inverted: polarity is active low

Each synchronization input has a digital filter that can be set independently. Each of the two filters can be enabled or disabled. If enabled, it can be set to one of six predefined time constants. The available selections are:


- Off: the filter is disabled (pass through) (default value)
- 10 µs: the filter is enabled with a 10 µs time constant
- 20 µs: the filter is enabled with a 20 µs time constant
- 50 µs: the filter is enabled with a 50 µs time constant
- 100 µs: the filter is enabled with a 100 µs time constant
- 200 µs: the filter is enabled with a 200 µs time constant
- 500 µs: the filter is enabled with a 500 µs time constant

14.3.3. Setup pulse generators

This page allows to inspect and change all the settings related to the pulse generators. The **Setup pulse generators** page is shown in the image of *Figure 14: the Setup pulse generators GN1-GN4 web page*.



Pulse generators

Pulse generator GN1

Source: 
 Edge: 
 Pulse delay: μs
 Pulse width: μs
 Holdoff time: μs


- Off
- Input TR1
- Input TR2
- Software SW1
- Software SW2
- Software SW3
- Software SW4
- Oscillator

Pulse generator GN2



Source: 
 Edge: 
 Pulse delay: μs
 Pulse width: μs
 Holdoff time: μs

- Rising
- Falling
- Both

Pulse generator GN3

Source: 
 Edge: 
 Pulse delay: μs
 Pulse width: μs
 Holdoff time: μs

Pulse generator GN4

Source: 
 Edge: 
 Pulse delay: μs
 Pulse width: μs
 Holdoff time: μs

Apply

Figure 14: the **Setup pulse generators GN1-GN4** web page

Each pulse generator can be set independently. Each pulse generator is triggered by a selectable source and can have a different pulse delay, pulse width and hold off interval.

- **Source** selects the trigger for the generator. It can be chosen between **Off** (default), **Input TR1**, **Input TR2**, **Oscillator**, **Software SW1**, **Software SW2**, **Software SW3** and **Software SW4**. When set to **Off** the generator is disabled. When set to **Oscillator** the internal free running oscillator is used as the trigger
- **Edge** selects the triggering edge for the generator. It can be chosen between **Rising** (default), **Falling** and **Both**. With the latter selection the generator is triggered by both the

rising and falling edges

- **Pulse delay** is the delay of the generated pulse with respect to the triggering source. The pulse delay may range from 0 μs to 1,023,000 μs with variable resolution
- **Pulse width** is the time duration of the generated pulse. The pulse width may range from 1 μs to 1,023,000 μs with variable resolution
- **Holdoff time** is the minimum time, after a pulse has been generated, that the relevant generator waits before possibly producing a new pulse. The hold off interval may range from 0 μs to 1,023,000 μs with variable resolution

Settings **Pulse delay**, **Pulse width** and **Holdoff time** are expressed in μs .

14.3.4. Setup light outputs

This page allows to inspect and change all the settings related to the DC/DC converter and to the light outputs LD1 and LD2. The **Setup light outputs** page is shown in the image of *Figure 15: the Setup light outputs web page*.

Light outputs

Converter settings

Converter mode: Fixed voltage
 Converter maximum voltage: V
 Converter preset voltage: V

Common driver settings

Driver mode: Independent outputs

Light output LD1

Source: Off
 Current range: Automatic
 Ton check: Disabled
 Toff check: Disabled
 Current reduction delay: ms
 Current value: A
 Ton max: ms
 Toff min: ms
 Final current value: %

Light output LD2

Source: Off
 Current range: Automatic
 Ton check: Disabled
 Toff check: Disabled
 Current reduction delay: ms
 Current value: A
 Ton max: ms
 Toff min: ms
 Final current value: %

Figure 15: the **Setup light outputs** web page

The fields under **Converter settings** collect all the settings related to the DC/DC converter. Basically, the DC/DC converter output voltage can be manually set to a specified value or can be automatically detected by the controller itself.

- **Converter mode** selects the operating mode for the DC/DC converter. It can be chosen between **Fixed voltage** (default) and **Autodetect voltage**. When set to **Autodetect voltage** the controller detects the lowest converter output voltage sufficient to drive the programmed current into the two lights. By selecting the lowest converter output voltage, the heat generated by the controller is minimized
- **Converter maximum voltage** is the maximum voltage the DC/DC converter is ever allowed to produce. This register defines the actual output voltage when the DC/DC converter operates in fixed voltage mode. Conversely, when the DC/DC converter is set to

automatically detect the output voltage, this register will be overwritten with the final output voltage value reached on the DC/DC converter

- **Converter preset voltage** is the actual output voltage setting for the DC/DC converter

The two output drivers in the controller can either operate in two modes: independent mode and shared mode. In the independent mode, the two light outputs are completely autonomous and current and timing settings are programmed independently. In the shared mode the two output drivers are interconnected. This results in twice the maximum output current than in the independent mode. In the shared mode, the controller effectively behaves as a single channel controller whose current and timing settings are controlled by setting the first light output. The fields under **Common driver settings** collect all the settings related to operation in independent or shared mode.

- **Driver mode** selects the operating mode for the controller outputs. It can be chosen between **Independent outputs** (default) and **Shared outputs**

When operating in independent mode, each light output can be set separately. Each light output can have a specific activation source, a specific output current and independent maximum turn-on and minimum turn-off parameters. Moreover, the protection against excessive turn-on time and protection against insufficient turn-off time can be independently enabled or disabled for each light output.

- **Source** selects the activation source for the light output. It can be chosen between **Off** (default), **Generator GN1**, **Generator GN2**, **Generator GN3**, **Generator GN4**, **Input TR1**, **Input TR2** and **Continuous**. When set to **Off** the light output is disabled. When set to **Continuous** the light output is always active
- **Current range** selects the current range for the light output. It can be chosen between **Automatic** range selection (default), **Low** range (from zero up to 250 mA in independent mode, or from zero up to 500 mA in shared mode) and **High** range (from zero up to 20 A in independent mode, or from zero up to 40 A in shared mode)
- **Current value** is the required output current expressed in A. According with the current range selected, the output current may range from zero up to 20 A in independent mode or from zero up to 40 A in shared mode with variable resolution. Upon entering the required current, the controller will adjust it to the closest achievable value
- **Ton check** enables or disables (default) the protection against excessive turn-on time
- **Ton max** is the maximum turn-on time for the connected light, expressed in ms. The maximum turn-on time may range from 1 ms to 255 ms with a resolution of 1 ms
- **Toff check** enables or disables (default) the protection against insufficient turn-off time
- **Toff min** is the minimum turn-off time for the connected light, expressed in ms. The minimum turn-off time may range from 1 ms to 255 ms with a resolution of 1 ms

Each light output driver includes a current reduction circuit that allows the automatic decrease of the current in the light in case it is continuously driven for a time longer than a specified, programmed amount. Settings **Current reduction delay** and **Final current value** control this feature.

- **Current reduction delay** is the delay time for the activation of the current reduction feature. Allowed values are in the range from zero to 60000 and are expressed in ms, so the current reduction delay can range from zero up to 60000 ms. A value of zero (factory default) disables the feature
- **Final current value** is the reduced current, in terms of percentage of the programmed regular current, for the light output. Allowed values are in the range from zero to 100 and are expressed in units of percentage, therefore the reduced current can vary from zero up to the programmed regular current. A value of zero (factory default) selects a reduced current of zero, effectively turning off the light output after the current reduction delay has expired

Settings **Ton max**, **Toff min** and **Current reduction delay** are expressed in ms.

14.3.5. Setup synch outputs

This page allows to inspect and change all the settings related to the two synchronization outputs. The **Setup synch outputs** page is shown in the image of *Figure 16: the Setup synch outputs web page*.

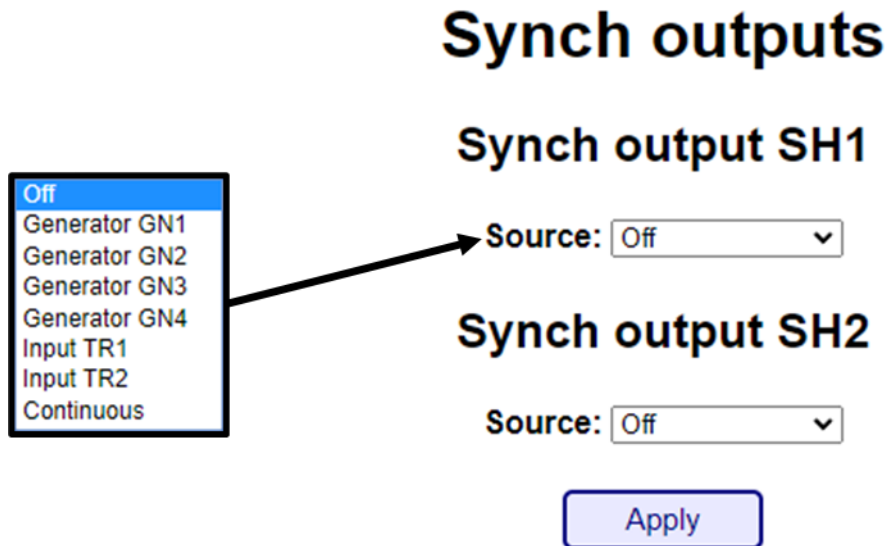


Figure 16: the **Setup synch outputs** web page

Each synchronization output can be set independently. Each one can have a different activation source chosen between **Off** (default), **Generator GN1**, **Generator GN2**, **Generator GN3**, **Generator GN4**, **Input TR1**, **Input TR2** and **Continuous**. When set to **Off** the synchronization output is disabled. When set to **Continuous** the synchronization output is always active.

14.3.6. General setup

This page allows to inspect and change the password for the controller, the Ethernet interface parameters, the serial RS485 settings and the internal oscillator parameters. The **General setup** page is shown in the image of *Figure 17: the General setup web page*.

General setup

Enter unlock password

Current password:

Ethernet interface

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

RS485 interface

Modbus address:
 Line speed: 1200
2400
4800
9600
19200
38400
57600
115200
 Line parity: None
Even
Odd

Internal oscillator

Oscillator period: ms

Figure 17: 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 [chapter 14.3.7](#)).

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 a maximum of 9600 bits per second. Selecting a higher rate will cause the serial interface to operate at 9600 bits per second.

The fields under Internal oscillator collect all the settings related to the internal oscillator.

- **Oscillator period** is the oscillator period in ms. Allowed values are in the range from 10 (corresponding to 100 Hz) up to 1000 (corresponding to 1 Hz)

14.3.7. 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 18: 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 18: 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 14.2](#) for a comprehensive list of the Modbus registers available.

15. 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.

15.1. 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.

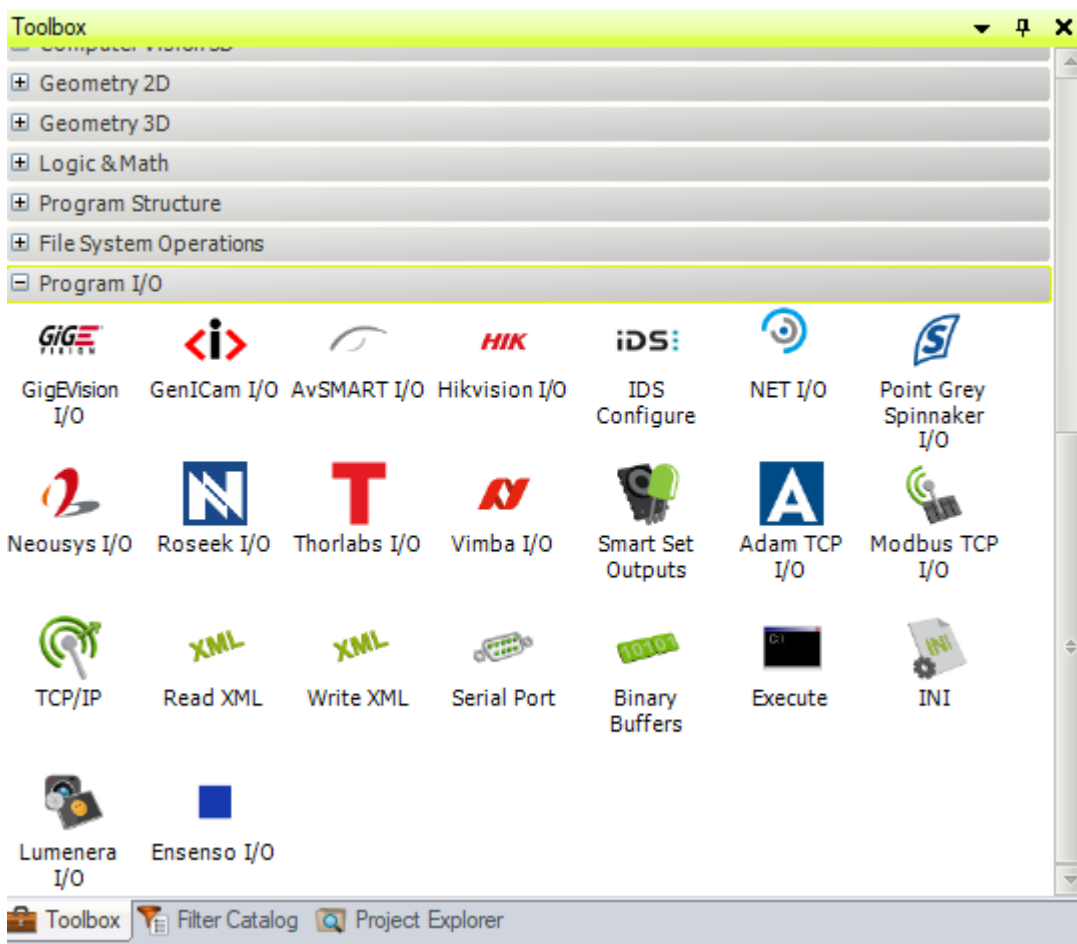


Figure 19. The **Program I/O** group of filters in FabImage

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.

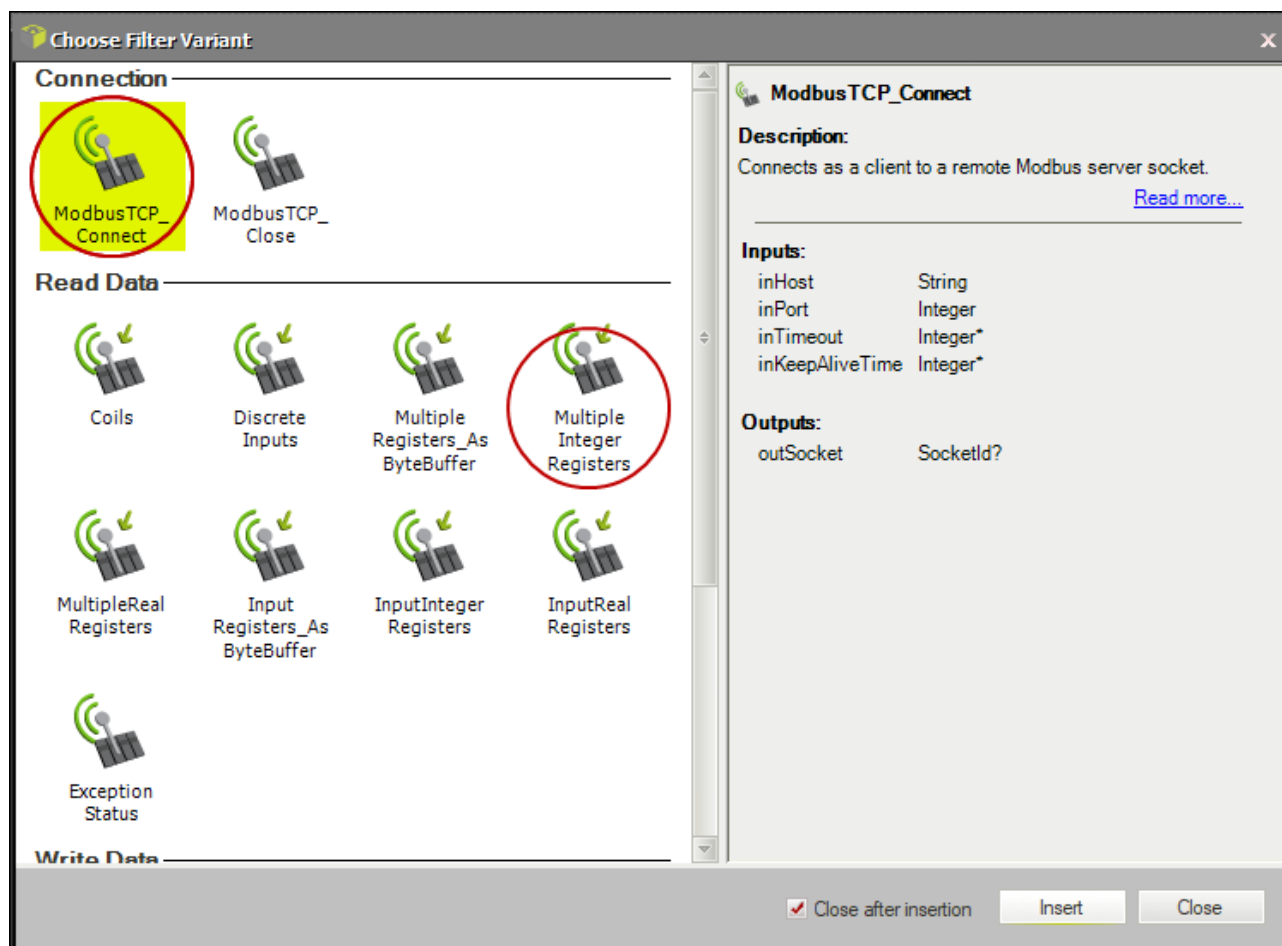


Figure 20. Selection of the Modbus TCP filter variant (top part)

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

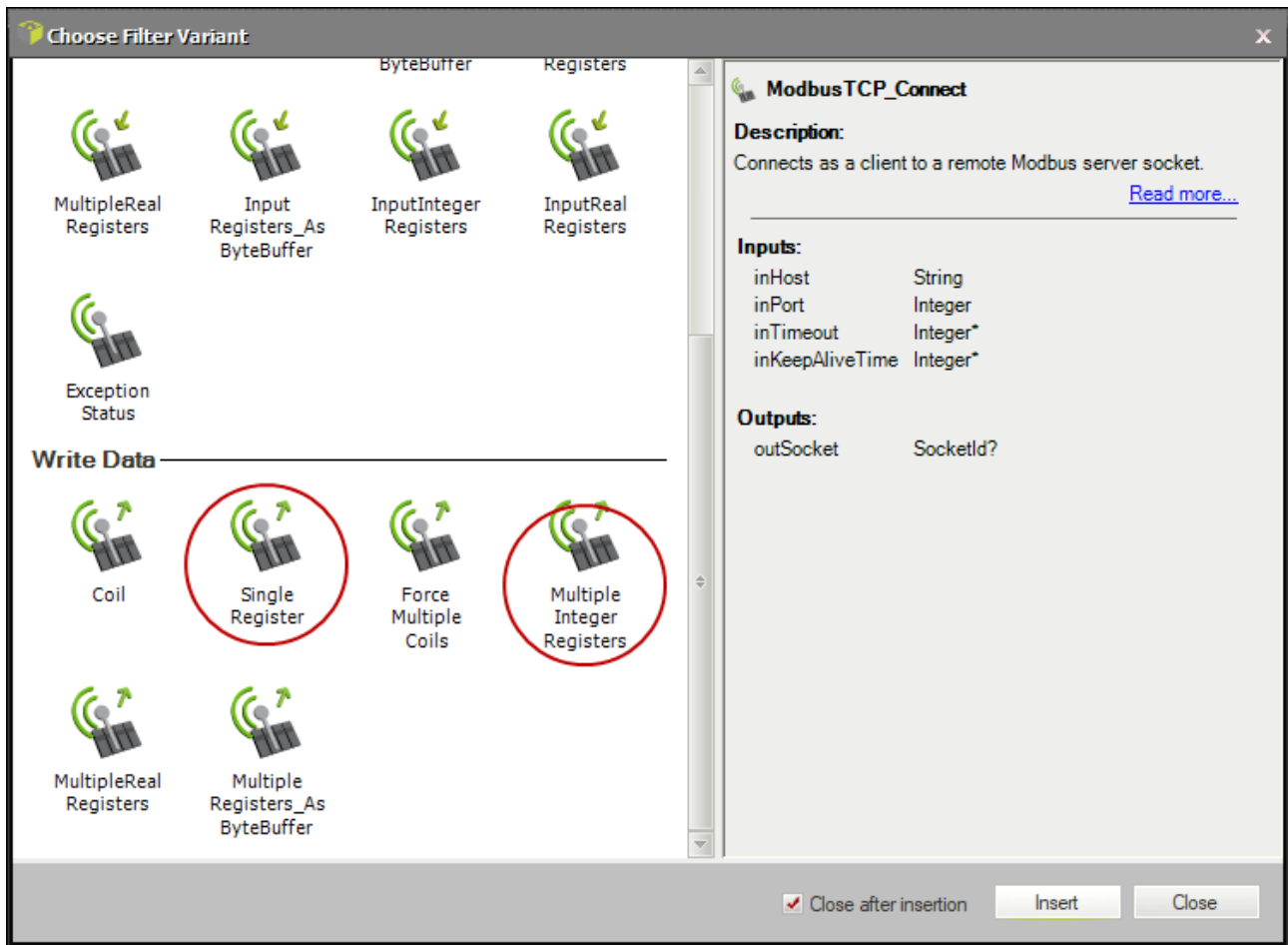


Figure 21. Selection of the Modbus TCP filter variant (bottom part)

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.

15.2. Designing a simple program for reading a register

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

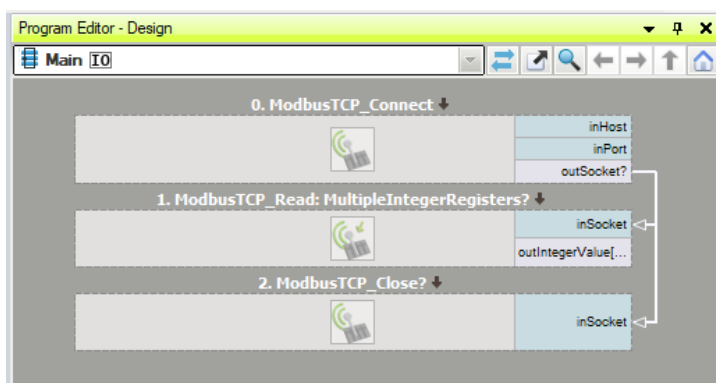


Figure 22. Example of data-flow code for reading a register

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.

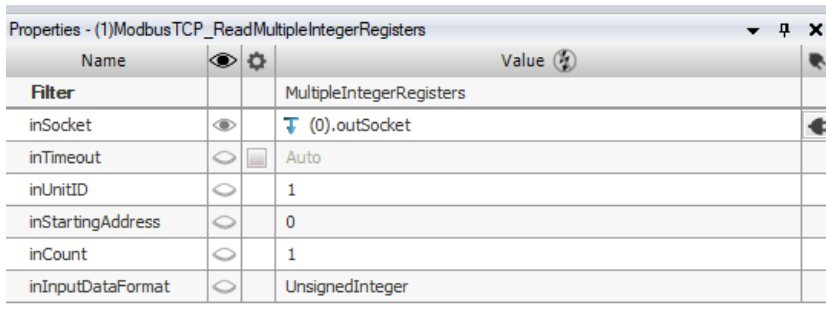
Configure the **ModbusTCP_Connect** filter as shown in the following figure.



Name	Value
inHost	192.168.0.32
inPort	502
inTimeout	Auto
inKeepAliveTime	Auto

Figure 23. Configuration of **ModbusTCP_Connect** filter

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



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

Figure 24. 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.

15.3. Designing a simple program for writing a register

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

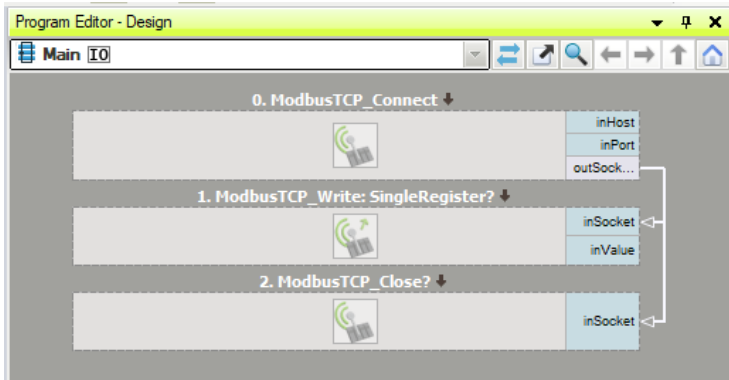


Figure 25. Example of data-flow code for writing a register

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.

Properties - (0)ModbusTCP_Connect			
Name		Value	
inHost		192.168.0.32	
inPort		502	
inTimeout		Auto	
inKeepAliveTime		Auto	

Figure 26. Configuration of **ModbusTCP_Connect** filter

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

Properties - (1)ModbusTCP_WriteSingleRegister			
Name		Value	
Filter		SingleRegister	
inSocket		(0).outSocket	
inTimeout		Auto	
inUnitID		1	
inRegisterAddress		0	
inValue		0	

Figure 27. 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.

16. 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).

In case of long cables electromagnetic interference may be counteracted by inserting ferrite cores on the power supply and light cables. In case, suggested parts are Würth Elektronik 74271221, to be applied with up to 3 turns of the cables.

17. Firmware update procedure

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

The first step is to connect the controller to be updated to the PC on which the LTDVE 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 the male D-shell connector P5. Data signal D+ is pin 15, data signal D- is pin 7 and the reference ground GND is pin 8.

The image in *Figure 28: connection of RS485-USB adapter to the controller* shows the RS485-USB adapter connected to the controller using a temporary female D-shell connector.

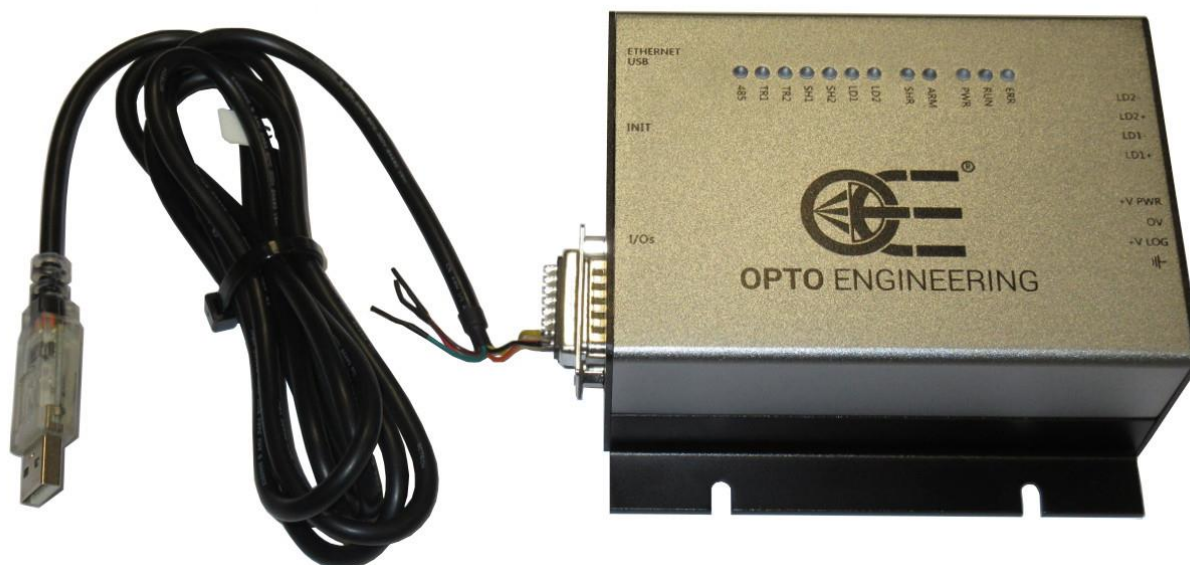


Figure 28: connection of RS485-USB adapter to the controller

The image in *Figure 29: main window of LTDVE firmware updater application* presents the main window of the PC application.

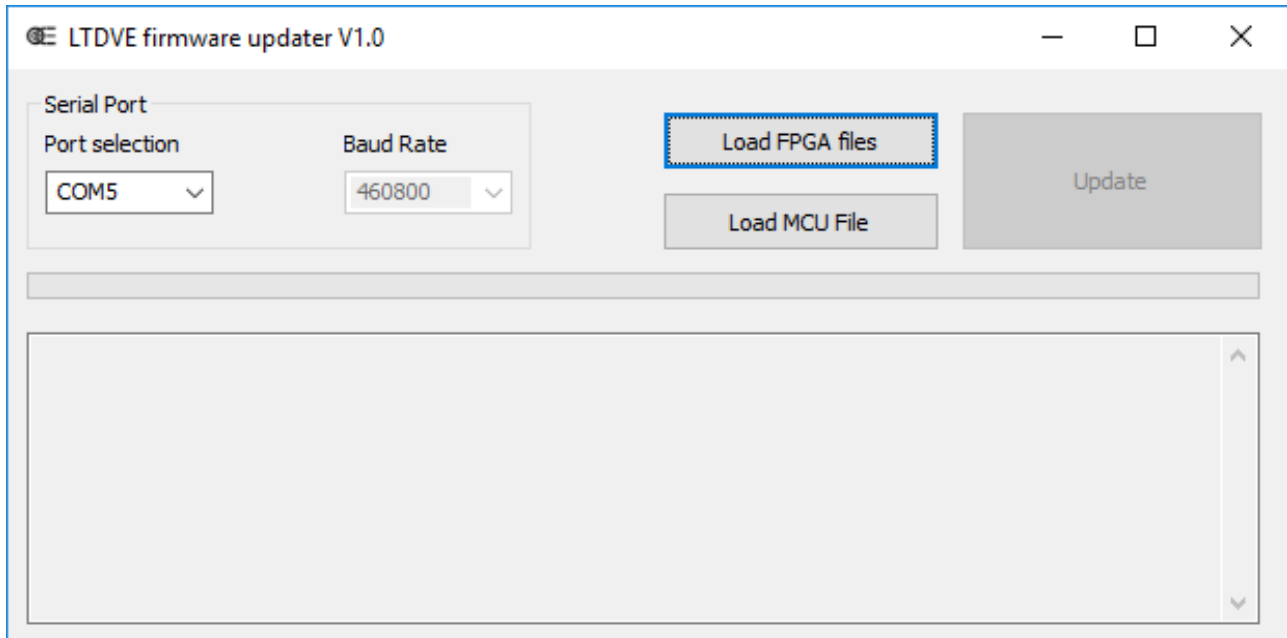


Figure 29: main window of **LTDVE firmware updater** application

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 are two programmable components, a MCU (Micro Controller Unit) and a FPGA (Field Programmable Gate Array), that must be updated at the same time. Failure to comply with this requirement can lead to controller malfunction. The two small buttons at the window right side are used to locate the relevant programming files in the PC file system.

To update the FPGA firmware it is necessary to specify a couple of files, named the FPGA Algo file and the FPGA Data file. After pressing the **Load FPGA files** button the modal dialog of *Figure 30: dialog used to locate the FPGA Algo firmware file* will be displayed, asking to specify the FPGA Algo file.

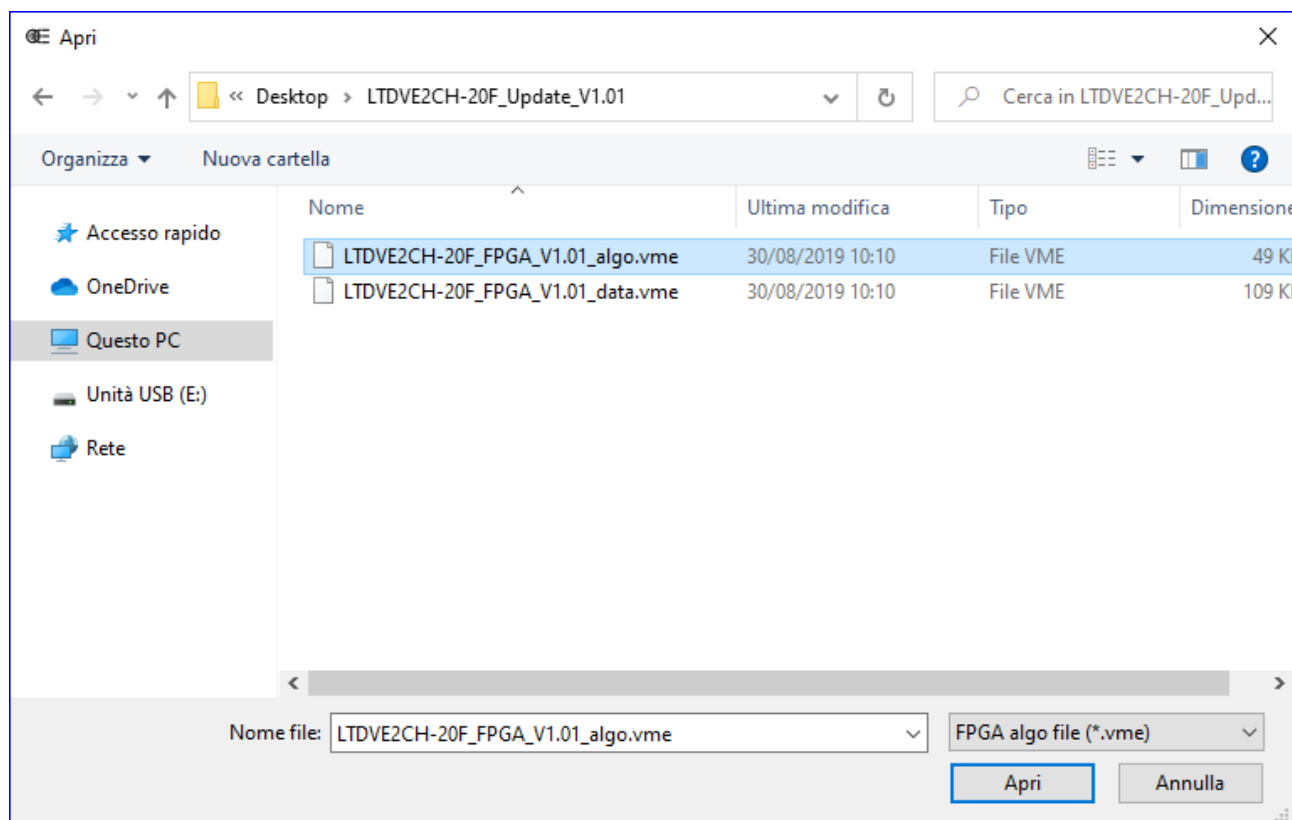


Figure 30: dialog used to locate the FPGA Algo firmware file

After the FPGA Algo file has been specified, the relative modal dialog is closed. The new modal dialog of *Figure 31: dialog used to locate the FPGA Data firmware file* will then be displayed, asking to specify the FPGA Data file.

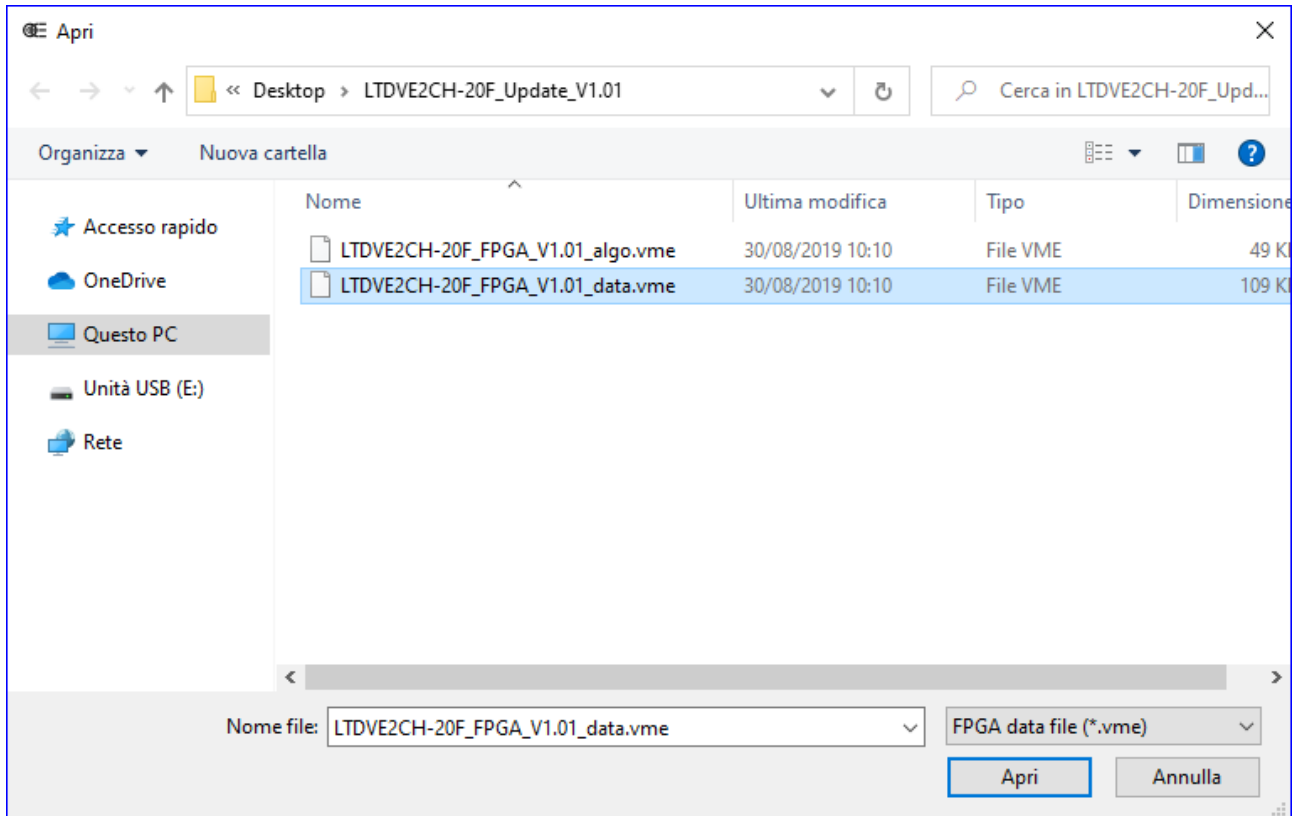


Figure 31: dialog used to locate the FPGA Data firmware file

After the FPGA Data file has been specified, the relative modal dialog is closed. The couple of FPGA firmware files 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 32: main window after the FPGA firmware files have been loaded).

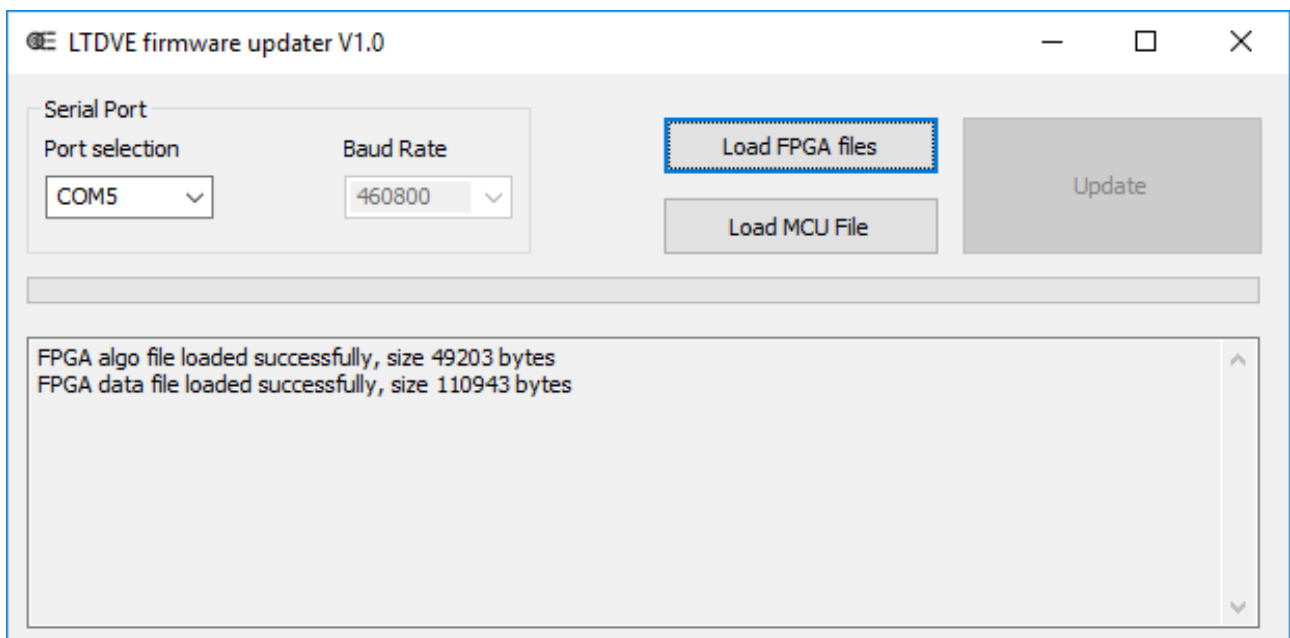


Figure 32: main window after the FPGA firmware files have been loaded

To update the MCU firmware it is necessary to specify a single file, named the MCU file. After pressing the **Load MCU file** button the modal dialog of *Figure 33: dialog used to locate the MCU firmware file* will be displayed, asking to specify the MCU file.

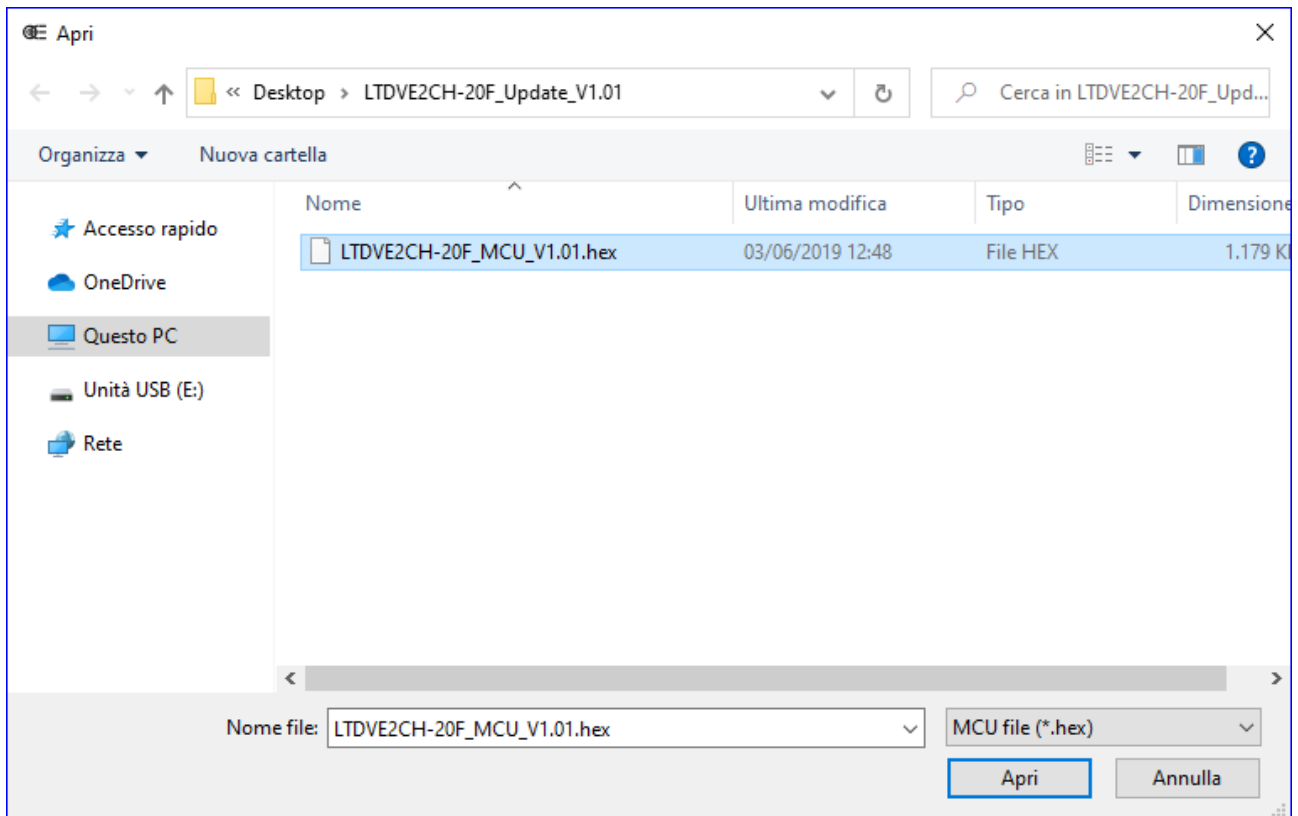


Figure 33: dialog used to locate the MCU firmware file

After the MCU file has been specified, the relative modal dialog is closed. The single 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 34: main window after the MCU firmware file has been loaded*).

The **Update** button at the right side of the main window becomes active after all the three files have been loaded from the PC disk to the PC RAM.

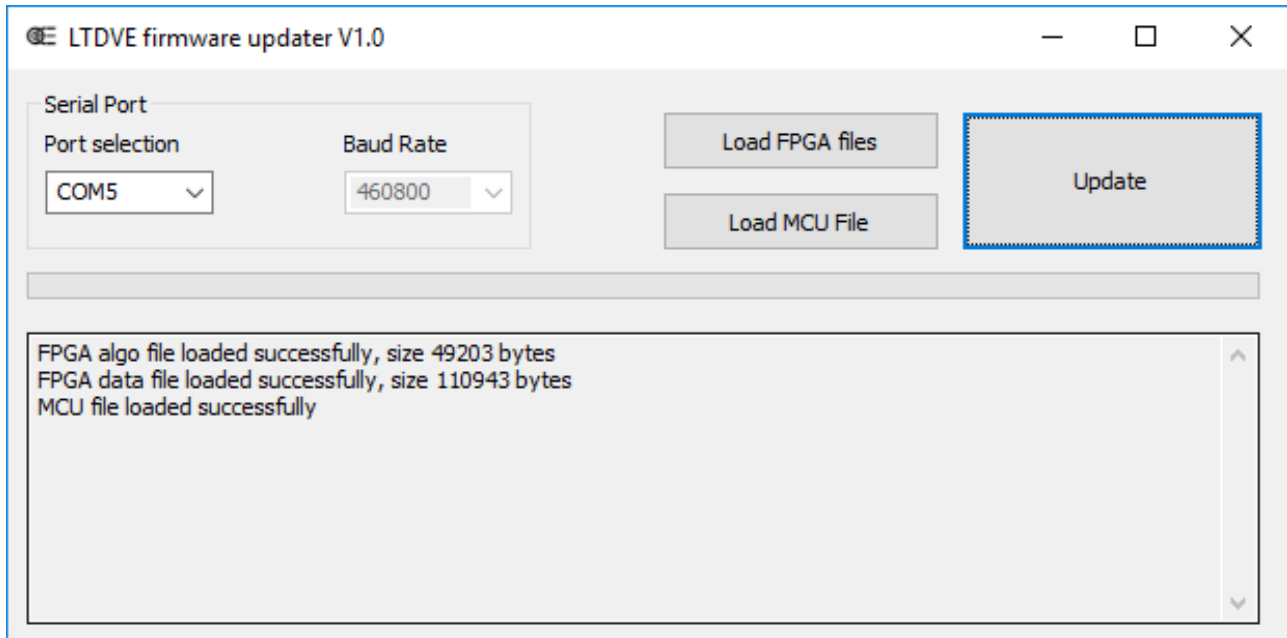


Figure 34: main window after the MCU firmware file has been loaded

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 first data transferred to the controller is the MCU firmware (see image in *Figure 35: MCU firmware update sequence*).

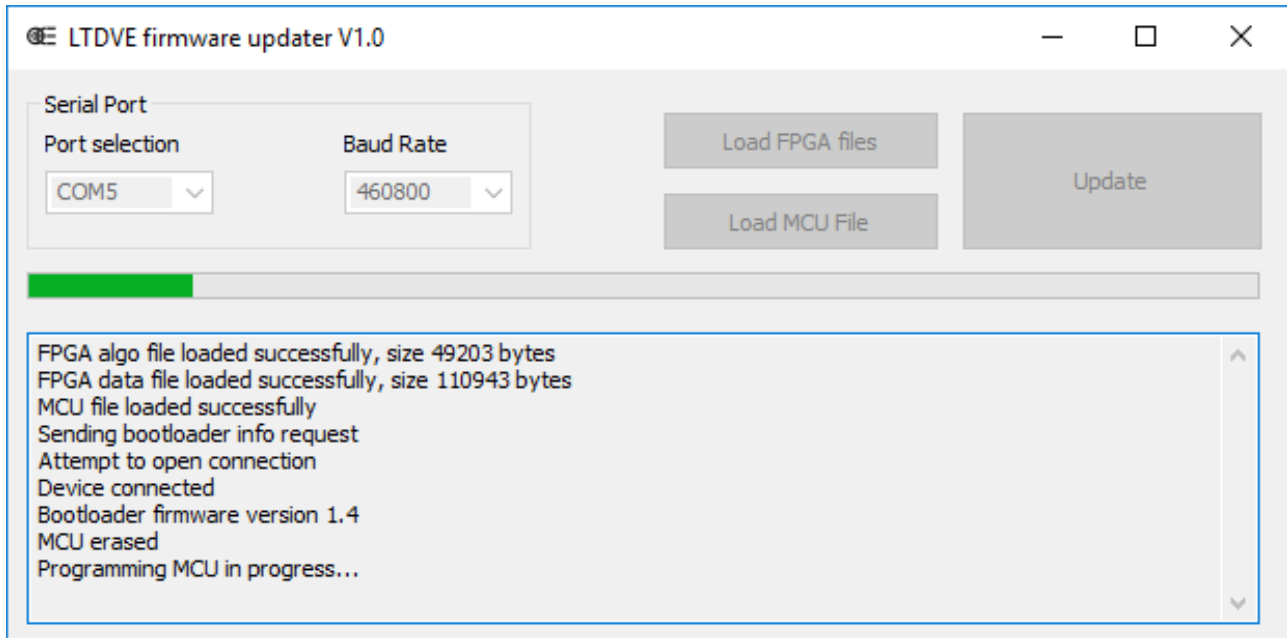


Figure 35: MCU firmware update sequence

Next, the FPGA Data firmware is transferred (see image in *Figure 36: FPGA Data firmware update sequence*).

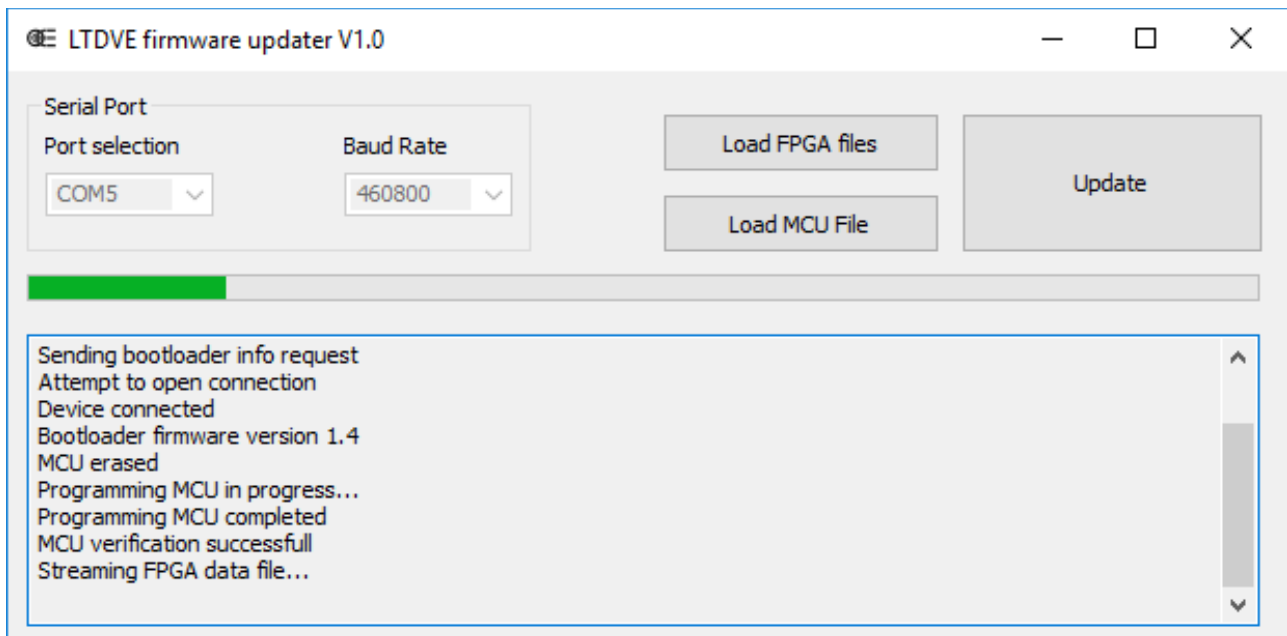


Figure 36: FPGA Data firmware update sequence

Last, the FPGA Algo firmware is transferred (see image in *Figure 37: FPGA Algo firmware update sequence*).

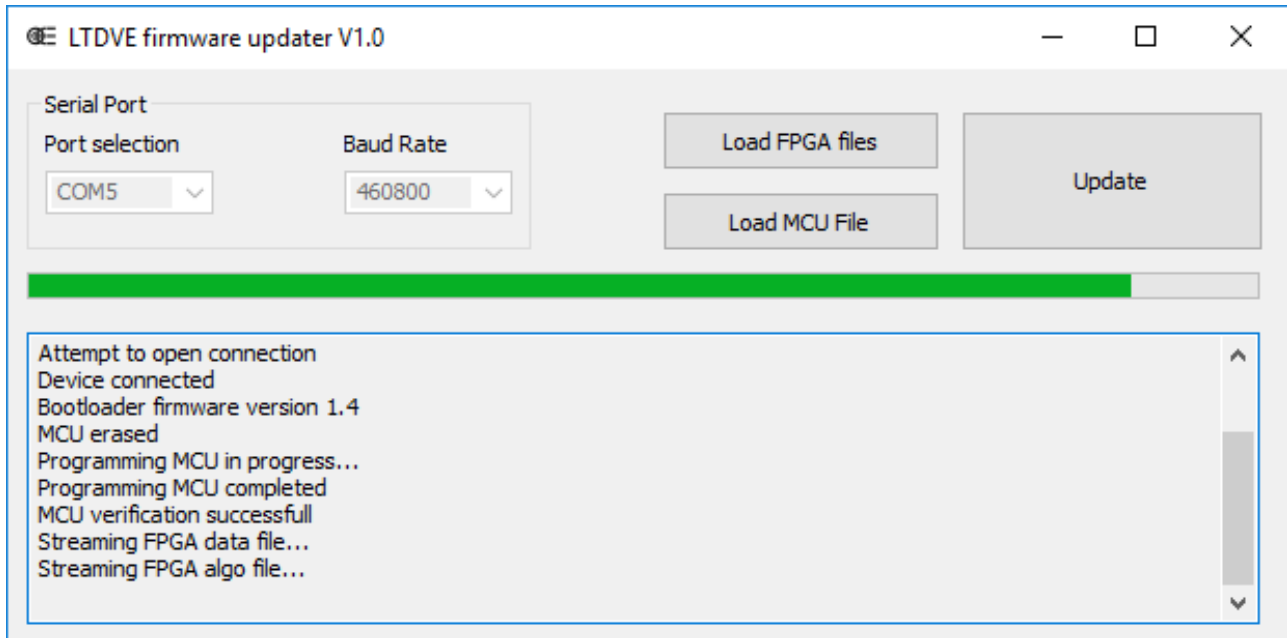


Figure 37: FPGA Algo firmware update sequence

The FPGA is programmed just after both the FPGA Data and the FPGA Algo files have been transferred (see image in Figure 38: main window after FPGA programming).

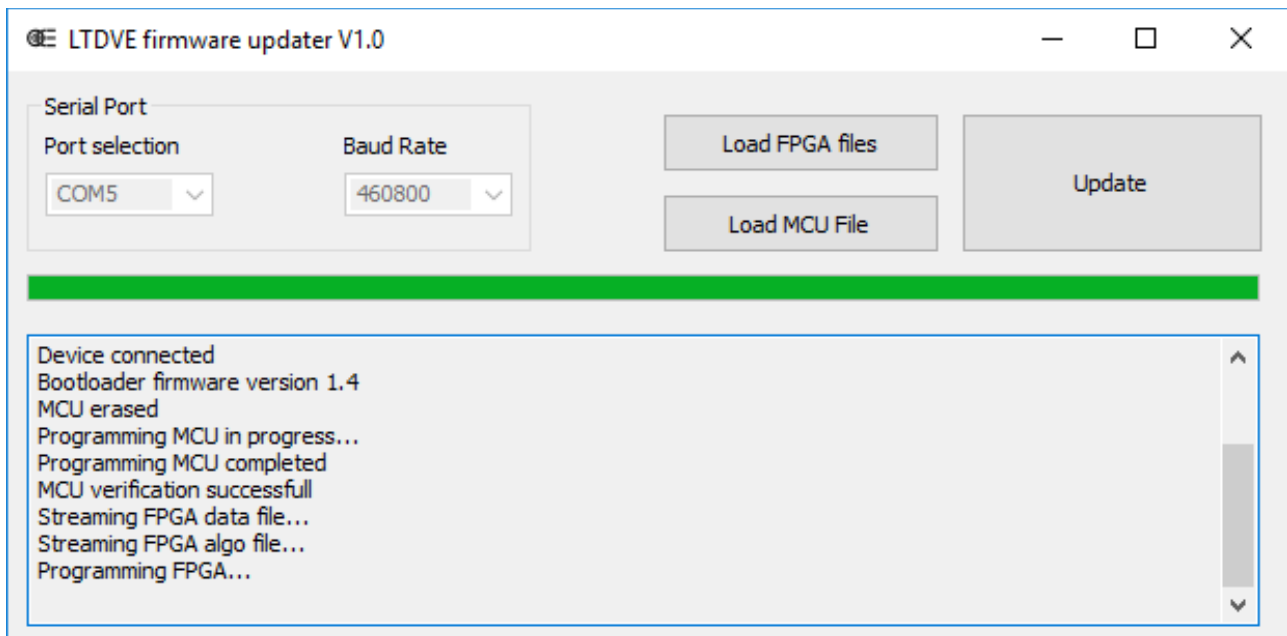


Figure 38: main window after FPGA programming

The information in Figure 39: main window after successful firmware update is displayed when the update process is successfully finished.

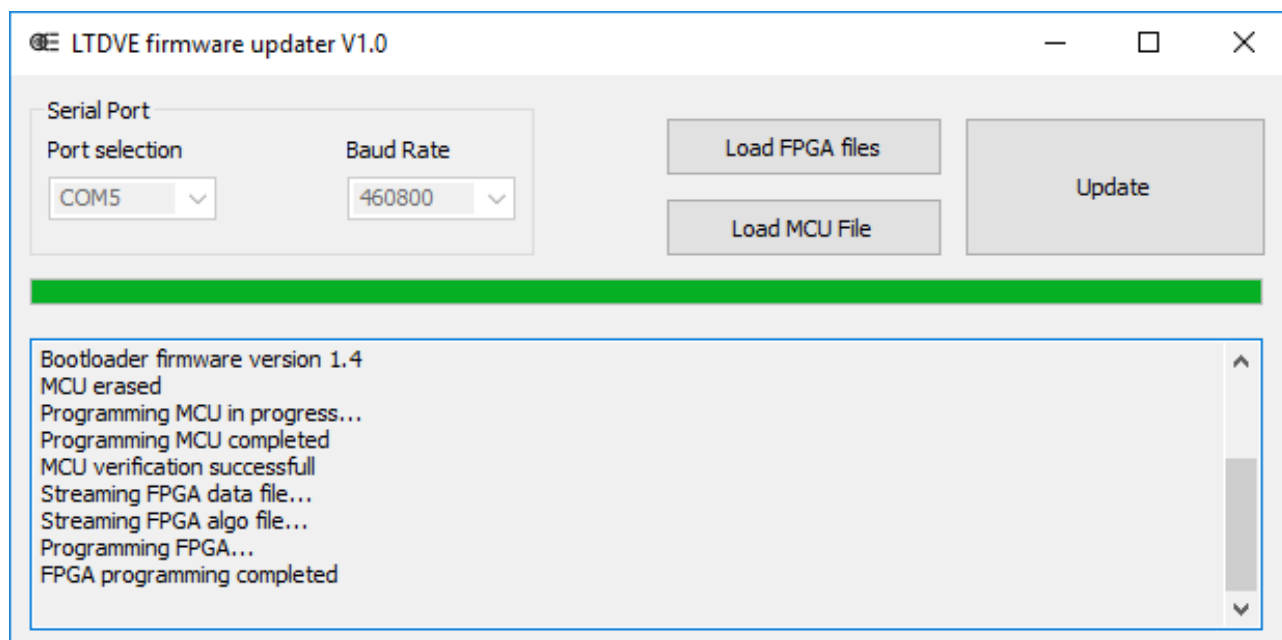


Figure 39: main window after successful firmware update

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



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