OPTO ENGINEERING

INSTRUCTIONS MANUAL

LTDVE4CH-20

Strobe controller 4 CH – Firmware version 1.10

ACCESSORIES

INDEX

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 four independent, programmable, current-controlled light outputs. The four 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 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 µs to 1 s. The pulse width can range from 1 µs 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.

There are three current ranges. They are:

- Low current, up to 200 mA (with resolution of 1 mA)
- Mid current, up to 4 A (with resolution of 4 mA)
- High current, up to 20 A (with resolution of 20 mA)

The controller must be powered with a fixed supply voltage between 24 V and 48 V DC. This allows a large number of different lights to be efficiently driven.

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 using a DIN rail as described in the section on Mechanical fixing.

Connect the controller as in the section on Connections. When the controller powers up it should show the 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 DIN rail. Allow free flow of air around the unit. The controller has an IP rating of 20 and should be installed so that moisture and dirt cannot enter it.

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

7. Heat dissipation

The controller integrates several 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.

The controller can approximately dissipate the following average powers:

- 30 W at 25 °C (about 7.5 W per channel)
- 25 W at 40 °C (about 6.3 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 [°C] – TempAmbient [°C]) / ThResistance [°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 1.91 °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).

The controller must be powered with a fixed supply voltages between 24 V and 48 V DC. Take care of the actual supply 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:

*Heat [W] = LightCurrent [A] * (SupplyVoltage [V] – LightVoltage [V]) * DutyCycle [·]*

Where:

- *LightCurrent* is the illuminator operating current

- *LightVoltage* is the illuminator operating voltage
- *SupplyVoltage* is the actual supply voltage (from 24 V to 48 V)
- *DutyCycle* is the actual duty cycle

The duty cycle is given by:

*DutyCycle [·] = PulseWidth [s] * 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.

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 four 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 four 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 bottom-side panel of the controller. Check all connections carefully before switching on the equipment.

The controller has two 24 V to 48 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 front panel* depicts all the controller connections, which are easily accessible on the bottom side panel. As indicated in the drawing, connectors are identified by their unique designators (P1, P2, P3, P4, P5 and P6).

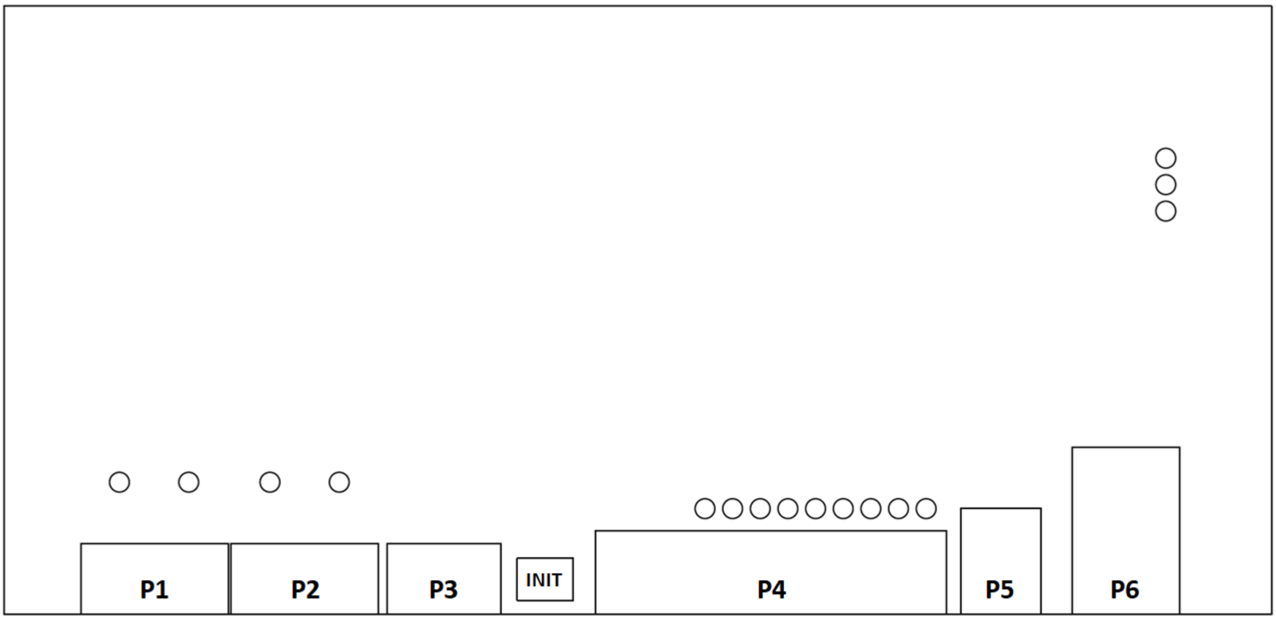


Figure 1: connectors on the controller front panel

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

- Connectors P1 and P2 are used to connect the four lights
- Connector P3 is used to supply power
- Connector P4 is used for input/output synchronization and for serial RS485 communication
- Connector P5 is a USB port (B type), not active at the moment
- Connector P6 is an Ethernet RJ45 jack

For connectors P1, P2, P3 and P4 a mating plug is provided in the controller package. For convenience the relevant manufacturer part numbers are listed in *Table 1: mating plugs for the controller connectors*. Even if equivalent mating plugs may be available, these are the recommended components.

Connector designator	**Manufacturer**	**Mating plug part number**
P1, P2	Phoenix Contact	1757035
P3	Phoenix Contact	1757022
P4	MH Connectors	MHDM25SS

Table 1: mating plugs for the controller connectors

8.2. Power and logic supply

The power supply voltage must be between 24 V and 48 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 P3. Connector pinout, ordered from left to right, is listed in *Table 2: pinout of connector P3*.

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

Table 2: pinout of connector P3

The controller has two 24 V to 48 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 two 4-way pluggable screw terminal sockets named P1 and P2. It is possible to use two 2-way connectors in a 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 next to the connector.

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.

8.3.1. Light outputs 1 and 2

Light outputs 1 and 2 are available on the LD1+, LD1-, LD2+ and LD2- screw terminals of connector P1. Connector pinout, ordered from left to right, is listed in *Table 3: pinout of connector P1*. 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 P1

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

8.3.2. Light outputs 3 and 4

Light outputs 3 and 4 are available on the LD3+, LD3-, LD4+ and LD4- screw terminals of connector P2. Connector pinout, ordered from left to right, is listed in *Table 4: pinout of connector P2*. Be careful not to cross-connect the two lights.

Number	Name	Description	Note
1	LD3+	Power channel 3 output. LED anode	
2	LD3-	Power channel 3 output. LED cathode	
3	LD4+	Power channel 4 output. LED anode	
4	LD4-	Power channel 4 output. LED cathode	

Table 4: pinout of connector P2

Please note that LED3- and LED4- are not the same as 0V.

8.4. Input/output synchronization

Connector P4 is used for input and output synchronization and for serial RS485 communication.

There are four 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 next to the connector.

There are four 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 60 V. The state of each synchronization output is shown by a yellow LED indicator next to the connector.

Connector P4 also provides three signals for an electrically isolated serial RS485 interface and two signals for an optional and non-electrically isolated external temperature sensor. The activity of the serial RS485 interface is shown by a dedicated yellow LED next to connector P4.

See the following section for more information about connector P4.

8.4.1. Synchronization inputs

The four synchronization inputs are available on the TR1+, TR1-, TR2+, TR2-, TR3+, TR3-, TR4+ and TR4- terminals of connector P4. These signals are listed in *Table 5: pinout of connector P4 for synchronization inputs*. Be careful not to cross-connect the four synchronization inputs.

Pin number	Name	Description
1	TR1-	Input 1. Negative terminal
14	TR1+	Input 1. Positive terminal
2	TR2-	Input 2. Negative terminal
15	TR2+	Input 2. Positive terminal
3	TR3-	Input 3. Negative terminal
16	TR3+	Input 3. Positive terminal
4	TR4-	Input 4. Negative terminal
17	TR4+	Input 4. Positive terminal

Table 5: pinout of connector P4 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.

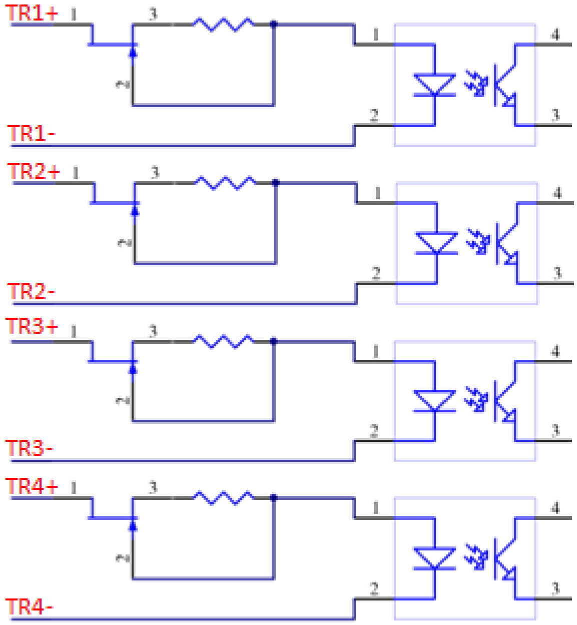


Figure 2: interface circuits for input synchronization

Circuit specifications are summarized in *Table 6: specifications of input synchronization circuits*. Please note the reported values are typical.

Parameter	Value	Unit	Note
Uin (low)	0 – 1	V	-
Uin (high)	3.3 – 24	V	-
Iin	5 – 9	mA	Internal constant-current generator

Table 6: specifications of input synchronization circuits

8.4.2. Synchronization outputs

The four synchronization outputs are available on the SH1+, SH1-, SH2+, SH2-, SH3+, SH3-, SH4+ and SH4- terminals of connector P4. These signals are listed in the *Table 7: pinout of connector P4 for synchronization outputs*. Be careful not to cross-connect the four synchronization outputs.

Pin number	Name	Description
5	SH1-	Output 1. Emitter terminal
18	SH1+	Output 1. Collector terminal
6	SH2-	Output 2. Emitter terminal
19	SH2+	Output 2. Collector terminal

7	SH3-	Output 3. Emitter terminal
20	SH3+	Output 3. Collector terminal
8	SH4-	Output 4. Emitter terminal
21	SH4+	Output 4. Collector terminal

Table 7: pinout of connector P4 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 60 V.

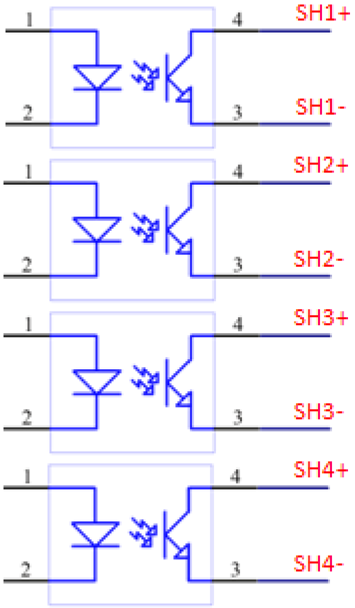


Figure 3: interface circuits for output synchronization

Circuit specifications are summarized in *Table 8: specifications of output synchronization circuits*. Please note the reported values are typical.

Parameter	**Value**	**Unit**	**Note**
Iout (typ)	10	mA	-
Iout (max)	15	mA	-
Uout (max)	30	V	-

Table 8: 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 P4. These signals are listed in *Table 9: pinout of serial interface in connector P4*. Be careful not to cross-connect the serial interface signals.

Pin number	Name	Description	Note
12	D-	RS485 data signal. Negative terminal	-
13	GND	RS485 reference ground	-
25	D+	RS485 data signal. Positive terminal	-

Table 9: pinout of serial interface in connector P4

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

8.4.4. External temperature sensor

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

The signals are listed in *Table 10: pinout of external temperature sensor in connector P4*. The two terminals can be connected freely to the external thermistor, as the component is not polarized.

Pin number	Name	Description
11	NTC_A	Temperature sensor terminal A
24	NTC_B	Temperature sensor terminal B

Table 10: pinout of external temperature sensor in connector P4

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 11: 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	5
Synchronization inputs	0.25	24	5
Synchronization outputs	0.25	24	5
Serial RS485 interface	0.25	24	5
External temperature sensor	0.25	24	5

Table 11: 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 sensor 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 next to connector P4.

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 12: default parameters for Ethernet communication*.

Parameter	Default Value

Host name	LTDVE4CH-20
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 12: 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 sixteen 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 output 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 13: 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	PWR	Green	Stable when logic supply is present
2	RUN	Green	Blinks periodically during normal operation
3	ERR	Red	Stable when power supply missing, blinks in error conditions
4	LD1	Yellow	Pulses when light output 1 is activated
5	LD2	Yellow	Pulses when light output 2 is activated
6	LD3	Yellow	Pulses when light output 3 is activated
7	LD4	Yellow	Pulses when light output 4 is activated
8	TR1	Green	Pulses when synchronization input 1 is activated
9	TR2	Green	Pulses when synchronization input 2 is activated
10	TR3	Green	Pulses when synchronization input 3 is activated
11	TR4	Green	Pulses when synchronization input 4 is activated
12	SH1	Yellow	Pulses when synchronization output 1 is activated
13	SH2	Yellow	Pulses when synchronization output 2 is activated
14	SH3	Yellow	Pulses when synchronization output 3 is activated
15	SH4	Yellow	Pulses when synchronization output 4 is activated

16	485	Yellow	Blinks when there is activity on the serial interface
17	ACT	Yellow	Blinks during Ethernet data transmission
18	LINK	Green	Stable when Ethernet connection established

Table 13: 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 four 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 eight pulse generators and several multiplexers. The pulse generators allow pulse delay and width control down to 1 μs 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.

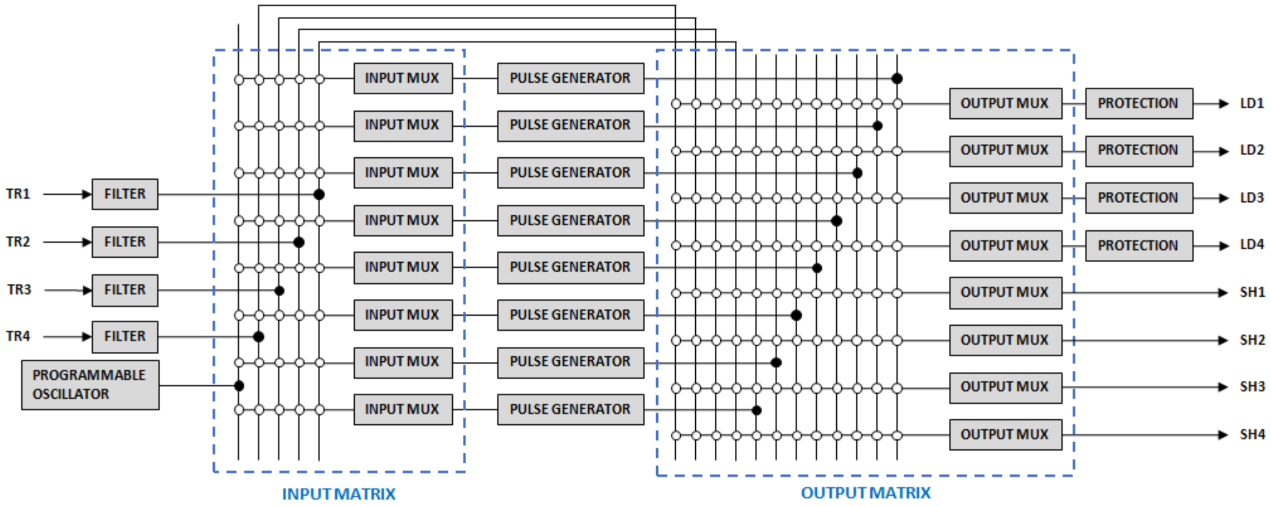


Figure 4: diagram of internal logic network

The four synchronization inputs are shown at the left (TR1, TR2, TR3 and TR4), while the four light outputs (LD1, LD2, LD3 and LD4) and the four synchronization outputs (SH1, SH2, SH3 and SH4) 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 four 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.

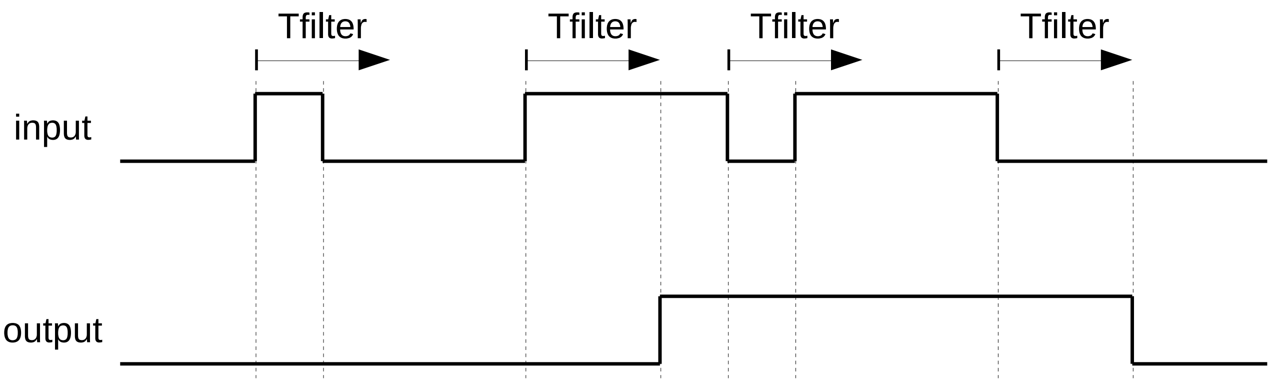


Figure 5: operation of the input filter

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 four 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 eight input multiplexers organized in a 5x8 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)
- Filtered synchronization input 3 (TR3)
- Filtered synchronization input 4 (TR4)
- Free running oscillator

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 generator

There are eight pulse generators. Each of them is characterized by two parameters: pulse delay and pulse width. 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 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.

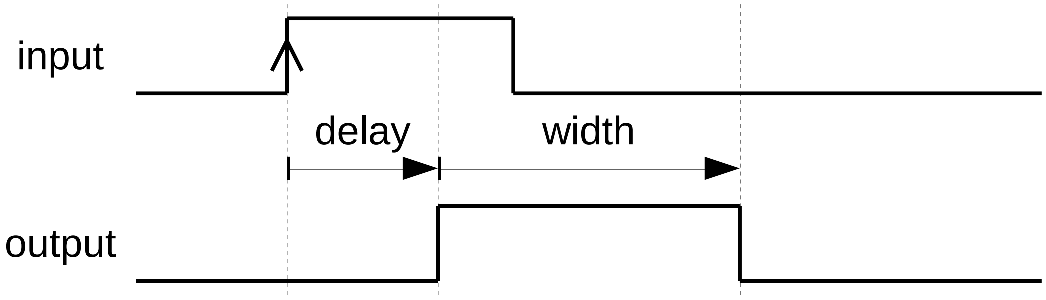


Figure 6: time diagram of pulse generator

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 eight output multiplexers organized in a 12x8 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
- Pulse generator 5 output
- Pulse generator 6 output
- Pulse generator 7 output
- Pulse generator 8 output
- Filtered synchronization input 1 (TR1)
- Filtered synchronization input 2 (TR2)
- Filtered synchronization input 3 (TR3)
- Filtered synchronization input 4 (TR4)
- Continuous

As visible in the internal logic network diagram (see *Figure 4: diagram of internal logic network*), the eight pulse generators can be entirely bypassed by selecting one of the four filtered synchronization inputs (TR1, TR2, TR3 or TR4). 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 four 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 (Ton) 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 (Toff) 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.

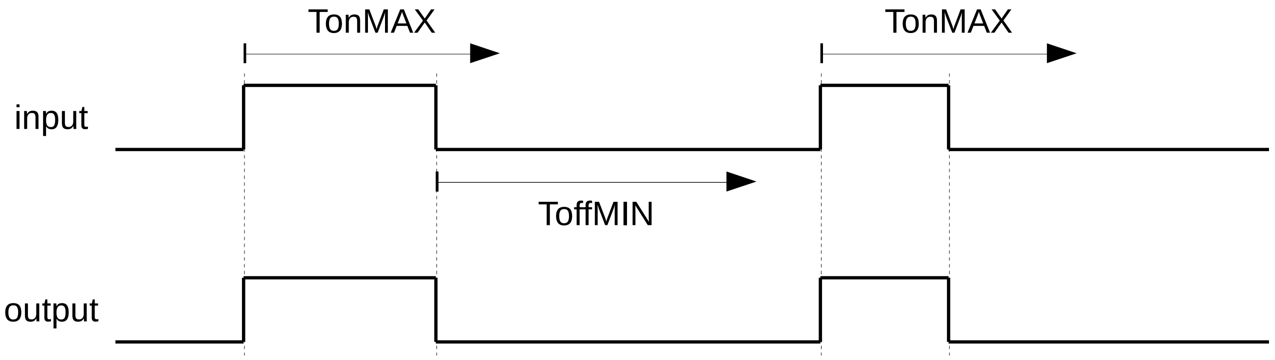


Figure 7: turn-on and turn-off times within limits

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.

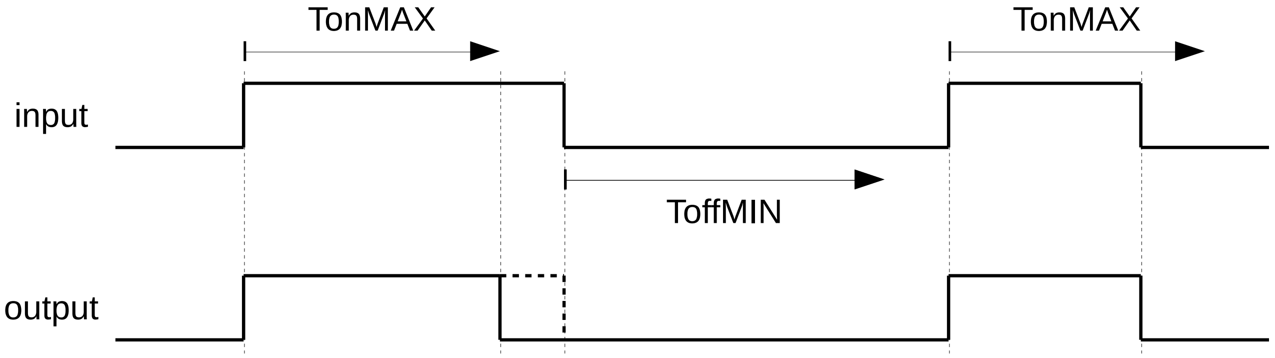


Figure 8: protection prevents too long turn-on time

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.

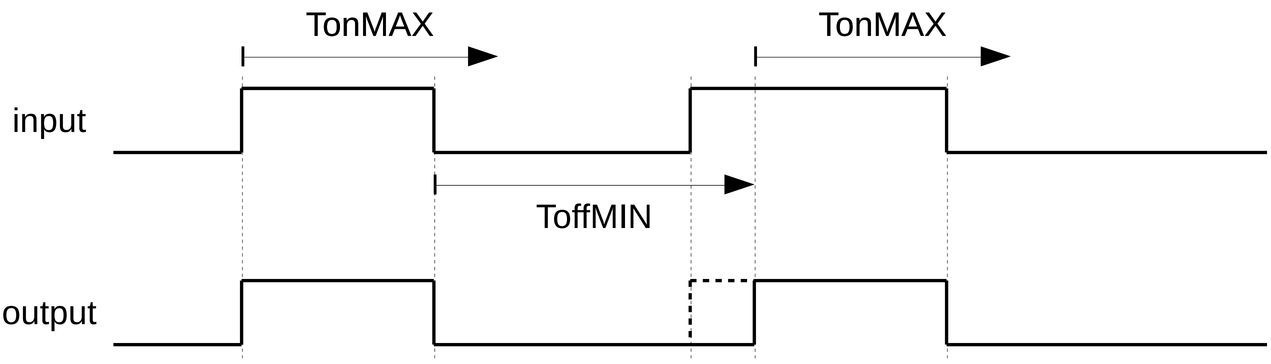


Figure 9: protection prevents too short turn-off time

The four 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 three lights and triggers a camera.

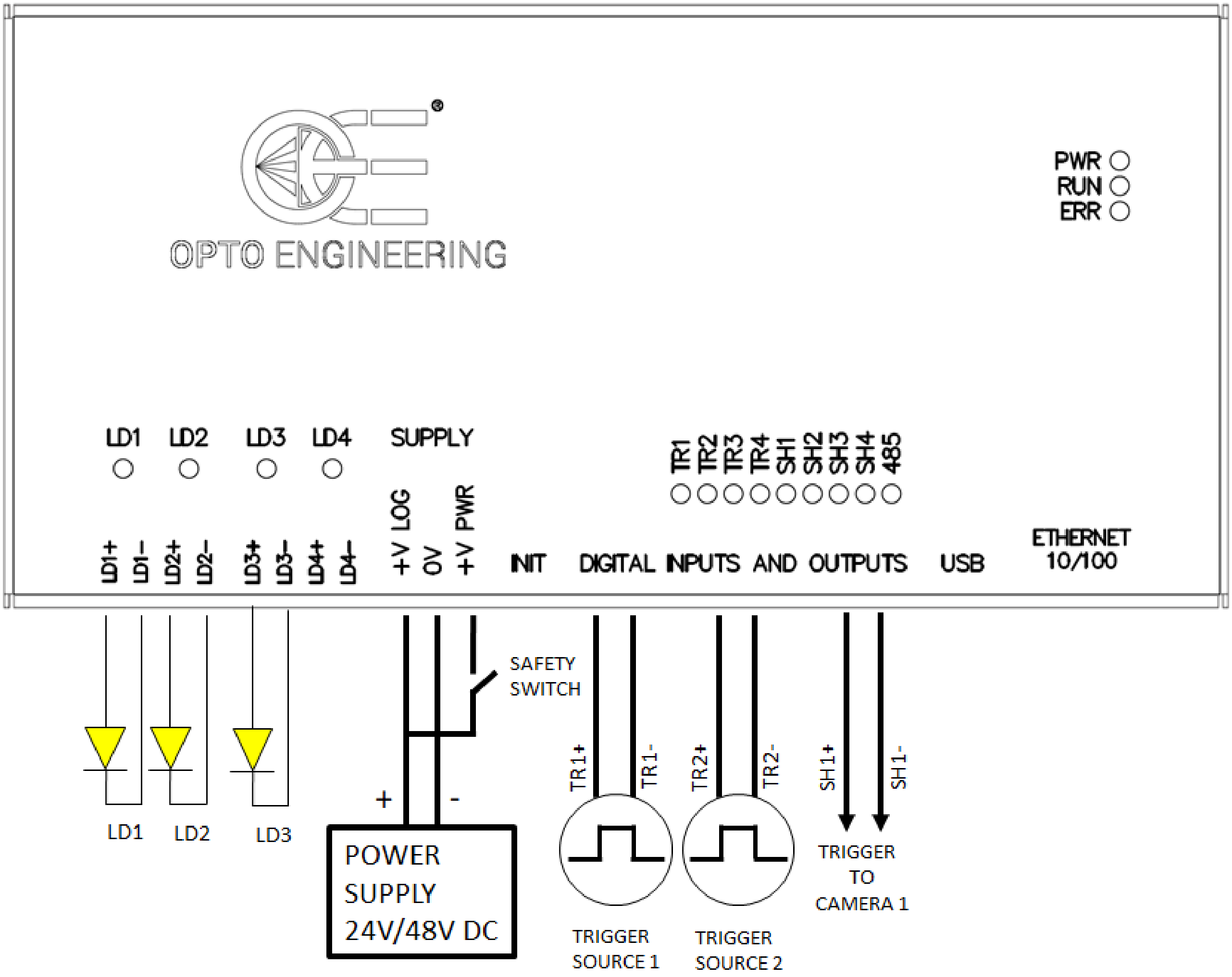


Figure 10: example schematic #1

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 four 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.

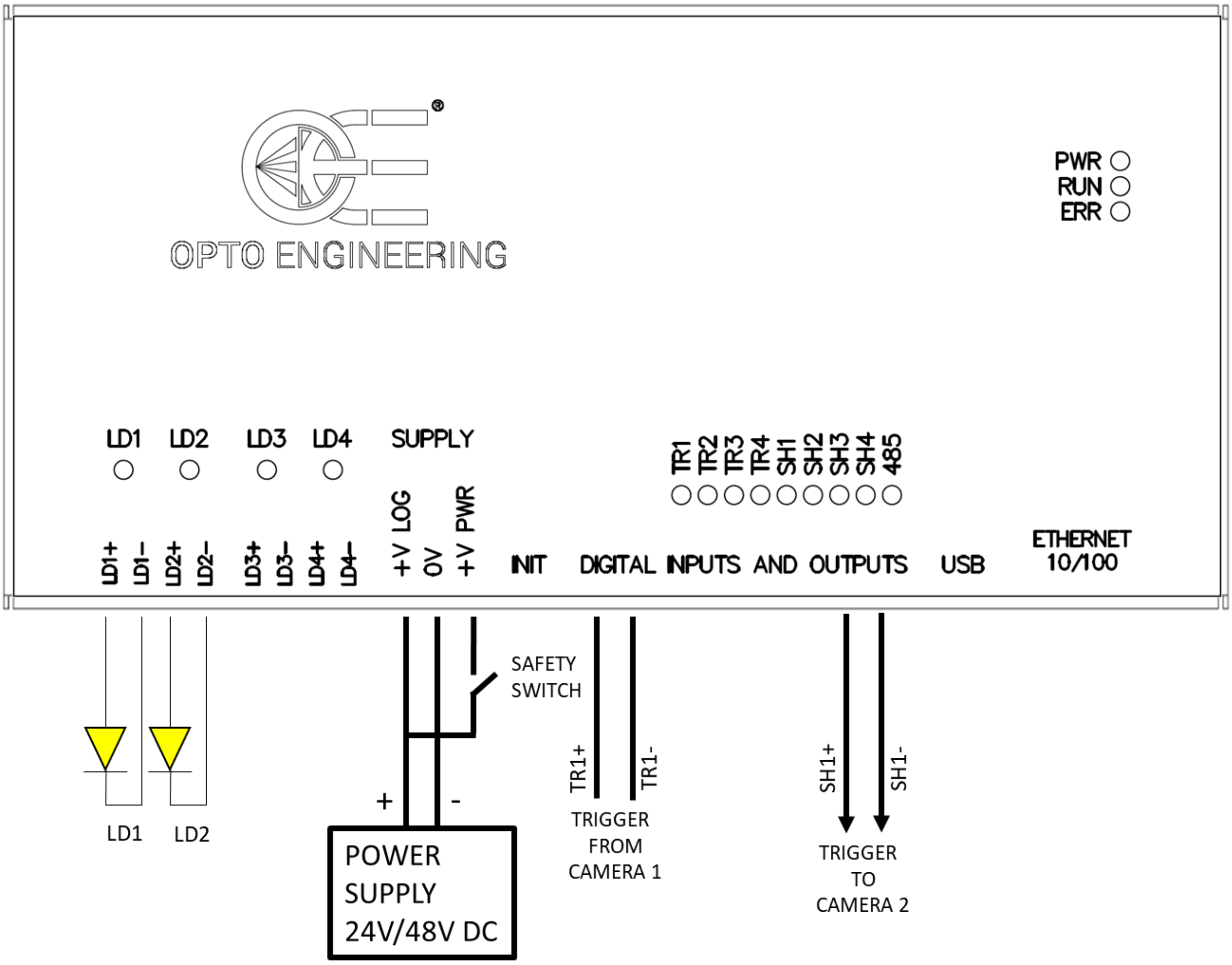


Figure 11: example schematic #2

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 four available synchronization inputs. Camera 2 is triggered by the controller using one of the four 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 14: 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 14: 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 15: 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	FILTER_SEL2	RW	0-6	Filter setting for input TR3

9	FILTER_SEL3	RW	0-6	Filter setting for input TR4
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	0-511	Setting of input multiplexer 1
15	INPUT_SEL1	RW	0-511	Setting of input multiplexer 2
16	INPUT_SEL2	RW	0-511	Setting of input multiplexer 3
17	INPUT_SEL3	RW	0-511	Setting of input multiplexer 4
18	INPUT_SEL4	RW	0-511	Setting of input multiplexer 5
19	INPUT_SEL5	RW	0-511	Setting of input multiplexer 6
20	INPUT_SEL6	RW	0-511	Setting of input multiplexer 7
21	INPUT_SEL7	RW	0-511	Setting of input multiplexer 8
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	GEN_DELAY_BASE4	RW	0-3	Pulse delay time base selector for generator 5
47	GEN_DELAY_COUNT4	RW	0-1023	Pulse delay setting for generator 5
48	GEN_WIDTH_BASE4	RW	0-3	Pulse width time base selector for generator 5
49	GEN_WIDTH_COUNT4	RW	1-1023	Pulse width setting for generator 5
50	GEN_DELAY_BASE5	RW	0-3	Pulse delay time base selector for generator 6
51	GEN_DELAY_COUNT5	RW	0-1023	Pulse delay setting for generator 6
52	GEN_WIDTH_BASE5	RW	0-3	Pulse width time base selector for generator 6
53	GEN_WIDTH_COUNT5	RW	1-1023	Pulse width setting for generator 6
54	GEN_DELAY_BASE6	RW	0-3	Pulse delay time base selector for generator 7
55	GEN_DELAY_COUNT6	RW	0-1023	Pulse delay setting for generator 7
56	GEN_WIDTH_BASE6	RW	0-3	Pulse width time base selector for generator 7
57	GEN_WIDTH_COUNT6	RW	1-1023	Pulse width setting for generator 7
58	GEN_DELAY_BASE7	RW	0-3	Pulse delay time base selector for generator 8

59	GEN_DELAY_COUNT7	RW	0-1023	Pulse delay setting for generator 8
60	GEN_WIDTH_BASE7	RW	0-3	Pulse width time base selector for generator 8
61	GEN_WIDTH_COUNT7	RW	1-1023	Pulse width setting for generator 8
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	OUTPUT_SEL_HI2	RW	0-511	High order byte of setting for LD3 output multiplexer
99	OUTPUT_SEL_LO2	RW	0-65535	Low order byte of setting for LD3 output multiplexer
100	OUTPUT_SEL_HI3	RW	0-511	High order byte of setting for LD4 output multiplexer
101	OUTPUT_SEL_LO3	RW	0-65535	Low order byte of setting for LD4 output multiplexer
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	OUTPUT_SEL_HI10	RW	0-511	High order byte of setting for SH3 output multiplexer
115	OUTPUT_SEL_LO10	RW	0-65535	Low order byte of setting for SH3 output multiplexer

116	OUTPUT_SEL_HI11	RW	0-511	High order byte of setting for SH4 output multiplexer
117	OUTPUT_SEL_LO11	RW	0-65535	Low order byte of setting for SH4 output multiplexer
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	PRT_CNT_ON2	RW	1-255	Maximum turn-on time for light output LD3
135	PRT_ENA_ON2	RW	0-1	Enable limitation of maximum turn-on time for light output LD3
136	PRT_CNT_OFF2	RW	1-255	Minimum turn-off time for light output LD3
137	PRT_ENA_OFF2	RW	0-1	Enable limitation of minimum turn-off time for light output LD3
138	PRT_CNT_ON3	RW	1-255	Maximum turn-on time for light output LD4
139	PRT_ENA_ON3	RW	0-1	Enable limitation of maximum turn-on time for light output LD4

140	PRT_CNT_OFF3	RW	1-255	Minimum turn-off time for light output LD4
141	PRT_ENA_OFF3	RW	0-1	Enable limitation of minimum turn-off time for light output LD4
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	0-3	Current range for light output LD1
159	CUR_VALUE0	RW	0-20000	Current value for light output LD1
160	CUR_RANGE1	RW	0-3	Current range for light output LD2
161	CUR_VALUE1	RW	0-20000	Current value for light output LD2
162	CUR_RANGE2	RW	0-3	Current range for light output LD3
163	CUR_VALUE2	RW	0-20000	Current value for light output LD3
164	CUR_RANGE3	RW	0-3	Current range for light output LD4
165	CUR_VALUE3	RW	0-20000	Current value for light output LD4
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-	High order word of Ethernet default

			65535	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	BOARD_TEMPERATURE0	R	-200 to +1000	Board temperature next to LD1 and LD2 output drivers
207	BOARD_TEMPERATURE1	R	-200 to +1000	Board temperature next to LD3 and LD4 output drivers
208	UNUSED	N/A	N/A	
209	UNUSED	N/A	N/A	
210	REMOTE_TEMPERATURE	R	-200 to +1000	Temperature measured by external thermal sensor
211	SUPPLY_VOLTAGE	R	0-480	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	MEASURED_CURRENT2	R	0-20000	Measured current for light output LD3
215	MEASURED_CURRENT3	R	0-20000	Measured current for light output LD4

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	0-480	Measured voltage for light output LD1
221	MEASURED_VOLTAGE1	R	0-480	Measured voltage for light output LD2
222	MEASURED_VOLTAGE2	R	0-480	Measured voltage for light output LD3
223	MEASURED_VOLTAGE3	R	0-480	Measured voltage for light output LD4
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	CAL_CUR_ADD0	R	0-65535	Calibration constant
230	CAL_CUR_MUL0	R	0-65535	Calibration constant
231	CAL_CUR_ADD1	R	0-65535	Calibration constant
232	CAL_CUR_MUL1	R	0-65535	Calibration constant
233	CAL_CUR_ADD2	R	0-65535	Calibration constant
234	CAL_CUR_MUL2	R	0-65535	Calibration constant
235	CAL_CUR_ADD3	R	0-65535	Calibration constant
236	CAL_CUR_MUL3	R	0-65535	Calibration constant
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	CAL_VLT_ADD0	R	0-65535	Calibration constant
246	CAL_VLT_MUL0	R	0-65535	Calibration constant
247	CAL_VLT_ADD1	R	0-65535	Calibration constant
248	CAL_VLT_MUL1	R	0-65535	Calibration constant
249	CAL_VLT_ADD2	R	0-65535	Calibration constant
250	CAL_VLT_MUL2	R	0-65535	Calibration constant
251	CAL_VLT_ADD3	R	0-65535	Calibration constant
252	CAL_VLT_MUL3	R	0-65535	Calibration constant
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	CAL_UNLOCK_CODE	R	0-65535	Unlock code for the calibration constants
262-510	RESERVED	N/A	N/A	Reserved for future use
511	BOARD_COMMAND	RW	0-3	Board command

Table 15: 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 reserved (type N/A). Registers marked as 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 LTDVE4CH-20 the device type is 0x0007.

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[0-3]

Each bit field [2:0] of these four 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
- **FILTER_SEL2**: filter setting for input TR3
- **FILTER_SEL3**: filter setting for input TR4

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 µs time constant
- When 0x2 filter is enabled with a 20 µs time constant
- When 0x3 filter is enabled with a 50 µs time constant
- When 0x4 filter is enabled with a 100 µs time constant
- When 0x5 filter is enabled with a 200 µs time constant
- When 0x6 filter is enabled with a 500 µs 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[0-7]

Each bit field [4:0] 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
- **INPUT_SEL4**: setting of input multiplexer 4
- **INPUT_SEL5**: setting of input multiplexer 5
- **INPUT_SEL6**: setting of input multiplexer 6
- **INPUT_SEL7**: setting of input multiplexer 7

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 0x0004 the filtered TR3 input is selected
- When 0x0008 the filtered TR4 input is selected
- When 0x0100 the free running oscillator is selected

Bit fields [15:9] and [7:4] of these registers are unused. When writing these bits, they must be set to zero.

14.2.9. Registers GEN_DELAY_BASE[0-7]

Each bit field [1:0] of these eight 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
- **GEN_DELAY_BASE4**: time base selector for generation of pulse delay in generator 5
- **GEN_DELAY_BASE5**: time base selector for generation of pulse delay in generator 6
- **GEN_DELAY_BASE6**: time base selector for generation of pulse delay in generator 7
- **GEN_DELAY_BASE7**: time base selector for generation of pulse delay in generator 8

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 μs is selected (default value)
- When 0x1 a time base of 10 μs is selected
- When 0x2 a time base of 100 μs is selected
- When 0x3 a time base of 1000 μs 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[0-7]

Each bit field [9:0] of these eight 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
- **GEN_DELAY_COUNT4**: pulse delay setting for generator 5
- **GEN_DELAY_COUNT5**: pulse delay setting for generator 6
- **GEN_DELAY_COUNT6**: pulse delay setting for generator 7
- **GEN_DELAY_COUNT7**: pulse delay setting for generator 8

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] [µs] = value(**GEN_DELAY_BASE**[x]) * value(**GEN_DELAY_COUNT**[x])*

The pulse delay may range from 0 µs to 1,023,000 µs 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[0-7]

Each bit field [1:0] of these eight 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
- **GEN_WIDTH_BASE4**: time base selector for generation of pulse width in generator 5
- **GEN_WIDTH_BASE5**: time base selector for generation of pulse width in generator 6
- **GEN_WIDTH_BASE6**: time base selector for generation of pulse width in generator 7
- **GEN_WIDTH_BASE7**: time base selector for generation of pulse width in generator 8

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 µs is selected (default value)
- When 0x1 a time base of 10 µs is selected
- When 0x2 a time base of 100 µs is selected
- When 0x3 a time base of 1000 µs 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[0-7]

Each bit field [9:0] of these eight 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
- **GEN_WIDTH_COUNT4**: pulse width setting for generator 5
- **GEN_WIDTH_COUNT5**: pulse width setting for generator 6
- **GEN_WIDTH_COUNT6**: pulse width setting for generator 7
- **GEN_WIDTH_COUNT7**: pulse width setting for generator 8

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] [µs] = value(**GEN_WIDTH_BASE**[x]) * value(**GEN_WIDTH_COUNT**[x])*

The pulse width may range from 1 µs to 1,023,000 µs 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[0-3] and OUTPUT_SEL_HI[8-11]

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_HI2**: upper sixteen bits of output multiplexer 3 selector (light LD3)
- **OUT_SEL_HI3**: upper sixteen bits of output multiplexer 4 selector (light LD4)
- **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)
- **OUT_SEL_HI10**: upper sixteen bits of output multiplexer 11 selector (sync. output SH3)
- **OUT_SEL_HI11**: upper sixteen bits of output multiplexer 12 selector (sync. output SH4)

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 0x0000:0010 pulse generator 5 output is selected
- When 0x0000:0020 pulse generator 6 output is selected
- When 0x0000:0040 pulse generator 7 output is selected
- When 0x0000:0080 pulse generator 8 output is selected
- When 0x0001:0000 filtered input TR1 is selected
- When 0x0002:0000 filtered input TR2 is selected
- When 0x0004:0000 filtered input TR3 is selected
- When 0x0008:0000 filtered input TR4 is selected
- When 0x0100:0000 the output is always active (continuous)

Bit fields [15:9] and [7:4] of these registers are unused. When writing these bits, they must be set to zero.

14.2.14. Registers OUTPUT_SEL_LO[0-3] and OUTPUT_SEL_LO[8-11]

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_LO2**: lower sixteen bits of output multiplexer 3 selector (light LD3)
- **OUT_SEL_LO3**: lower sixteen bits of output multiplexer 4 selector (light LD4)
- **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)
- **OUT_SEL_LO10**: lower sixteen bits of output multiplexer 11 selector (sync. output SH3)
- **OUT_SEL_LO11**: lower sixteen bits of output multiplexer 12 selector (sync. output SH4)

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.

14.2.15. Registers PRT_CNT_ON[0-3]

Each bit field [7:0] of these four registers holds the maximum turn-on time for light outputs LD1, LD2, LD3 and LD4.

- **PRT_CNT_ON0**: maximum turn-on time for light output LD1
- **PRT_CNT_ON1**: maximum turn-on time for light output LD2
- **PRT_CNT_ON2**: maximum turn-on time for light output LD3
- **PRT_CNT_ON3**: maximum turn-on time for light output LD4

Allowed values are in the range from 1 (default value) to 255 (maximum value) and are expressed in ms, so the maximum turn-on time can range from 1 ms to 255 ms with a resolution of 1 ms. Avoid operation with non-allowed values.

Bit fields [15:8] of these registers are unused. When writing these bits, they must be set to zero.

14.2.16. Registers PRT_ENA_ON[0-3]

Bit 0 of these four registers holds the enable flag for the limitation of turn-on time for light outputs LD1, LD2, LD3 and LD4.

- **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
- **PRT_ENA_ON2**: enable flag for the limitation of turn-on time for light output LD3
- **PRT_ENA_ON3**: enable flag for the limitation of turn-on time for light output LD4

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[0-3]

Each bit field [7:0] of these four registers holds the minimum turn-off time for light outputs LD1, LD2, LD3 and LD4.

- **PRT_CNT_OFF0**: minimum turn-off time for light output LD1
- **PRT_CNT_OFF1**: minimum turn-off time for light output LD2
- **PRT_CNT_OFF2**: minimum turn-off time for light output LD3
- **PRT_CNT_OFF3**: minimum turn-off time for light output LD4

Allowed values are in the range from 1 (default value) to 255 (maximum value) and are expressed in ms, so the minimum turn-off time can range from 1 ms to 255 ms with a resolution of 1 ms. Avoid operation with non-allowed values.

Bit fields [15:8] of these registers are unused. When writing these bits, they must be set to zero.

14.2.18. Registers PRT_ENA_OFF[0-3]

Bit 0 of these four registers holds the enable flag for the limitation of turn-off time for light outputs LD1, LD2, LD3 and LD4.

- **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
- **PRT_ENA_OFF2**: enable flag for the limitation of turn-off time for light output LD3
- **PRT_ENA_OFF3**: enable flag for the limitation of turn-off time for light output LD4

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[0-3]

Each bit field [1:0] of these four 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
- **CUR_RANGE2**: current range setting for light output LD3
- **CUR_RANGE3**: current range setting for light output LD4

Allowed values are in the range from 0 to 3 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 200 mA) is selected
- When 0x2 the mid current range (zero up to 4000 mA) is selected
- When 0x3 the high current range (zero up to 20000 mA) is selected

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

14.2.20. Registers CUR_VALUE[0-3]

Each of these four 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
- **CUR_VALUE2**: current value setting for light output LD3
- **CUR_VALUE3**: current value setting for light output LD4

Allowed values for these registers are in the range from zero 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 current firmware supports only 9600 bits per second. Value in this register is therefore forced to the default 3.

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).

Note the current firmware supports only even parity. Value in this register is therefore forced to the default 1.

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 BOARD_TEMPERATURE[0-1]

The controller integrates two temperature sensors. There is a temperature sensor for every couple of power stages.

- **BOARD_TEMPERATURE0**: average temperature of LD1 and LD2 power stages
- **BOARD_TEMPERATURE1**: average temperature of LD3 and LD4 power stages

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

*BoardTemperature[x] [°C] = value(**BOARD_TEMPERATURE**[x]) * 0.1*

The average heatsink temperature may be estimated by averaging the temperatures measured by the two temperature sensors. These registers are continuously updated.

14.2.42. Register REMOTE_TEMPERATURE

The controller allows for the connection of one external temperature sensor (see section 8.4.4 for more information).

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

*RemoteTemperature [°C] = value(**REMOTE_TEMPERATURE**) * 0.1*

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

14.2.43. 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. This register is read only.

14.2.44. Registers MEASURED_CURRENT[0-3]

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

- **MEASURED_CURRENT0**: measured current for light output LD1
- **MEASURED_CURRENT1**: measured current for light output LD2
- **MEASURED_CURRENT2**: measured current for light output LD3
- **MEASURED_CURRENT3**: measured current for light output LD4

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.

14.2.45. Registers MEASURED_VOLTAGE[0-3]

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

- **MEASURED_VOLTAGE0**: measured voltage for light output LD1
- **MEASURED_VOLTAGE1**: measured voltage for light output LD2
- **MEASURED_VOLTAGE2**: measured voltage for light output LD3
- **MEASURED_VOLTAGE3**: measured voltage for light output LD4

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:

*MeasuredVoltage[x] [V] = value(**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.

14.2.46. 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 LD1 and LD2 output drivers, reset if safe temperature
- Bit 2: set if excessive temperature on LD3 and LD4 output drivers, reset if 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
- Bit 7: set if LD3 output driver has been disabled, reset if LD3 output driver is active
- Bit 8: set if LD4 output driver has been disabled, reset if LD4 output driver is active

When any of the two thermal sensors on the output drivers reports an excessive temperature, the relevant error bit (1 or 2) is set and the relevant couple of output drivers is shut down. This happens when the measured temperature goes above about 90 °C. The same couple of output drivers is then reactivated when the temperature goes below about 80 °C.

When a hardware problem is detected on any of the four output channels, the relevant error bit (5, 6, 7 or 8) 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, SH2, SH3 and SH4 are never disabled.

The upper bits of this register are unused.

14.2.47. 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 a valid unlock code is entered in register **CAL_UNLOCK_CODE**.

14.2.48. Register CAL_UNLOCK_CODE

This register contains an unlock code to allow saving in non-volatile memory of the various calibration constants used for current generation, current measurement, voltage measurement and temperature measurement.

14.2.49. Register BOARD_COMMAND

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

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

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

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).

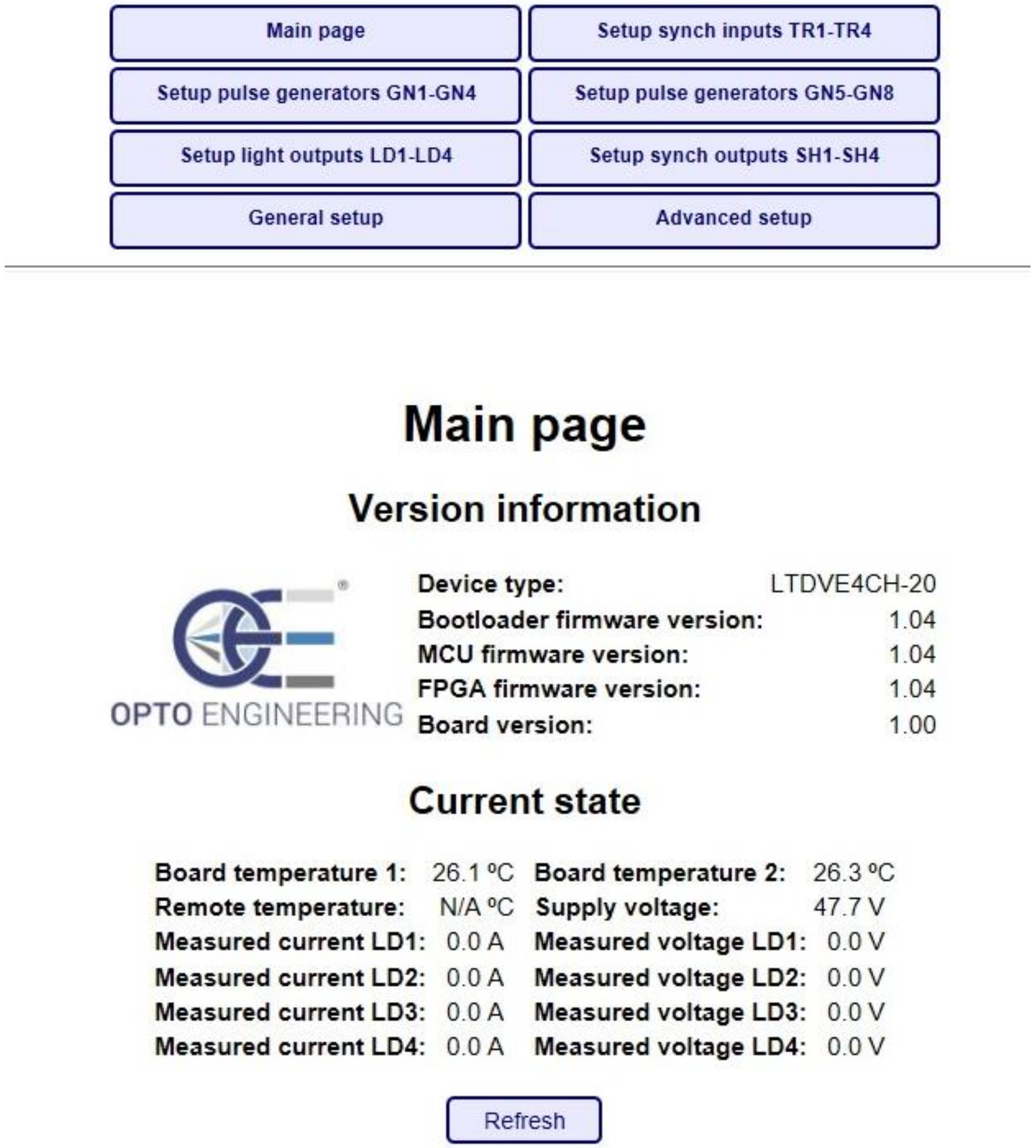
For more information about the Modbus/RTU, Modbus/TCP and Modbus/UDP protocols and the Modbus register file please refer to chapter 14.1 and chapter 14.2.

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
Setup synch inputs TR1-TR4
Setup pulse generators GN1-GN4
Setup pulse generators GN5-GN8
Setup light outputs LD1-LD4
Setup synch outputs SH1-SH4
General setup
Advanced setup
OPTO ENGINEERING
Refresh

Main page

Version information

Device type:	LTDVE4CH-20
Bootloader firmware version:	1.04
MCU firmware version:	1.04
FPGA firmware version:	1.04
Board version:	1.00

Current state

Board temperature 1:	26.1 ºC	Board temperature 2:	26.3 ºC
Remote temperature:	N/A ºC	Supply voltage:	47.7 V
Measured current LD1:	0.0 A	Measured voltage LD1:	0.0 V
Measured current LD2:	0.0 A	Measured voltage LD2:	0.0 V
Measured current LD3:	0.0 A	Measured voltage LD3:	0.0 V
Measured current LD4:	0.0 A	Measured voltage LD4:	0.0 V

*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 TR1-TR4**
- **Setup pulse generators GN1-GN4**
- **Setup pulse generators GN5-GN8**

- **Setup light outputs LD1-LD4**
- **Setup synch outputs SH1-SH4**
- **General setup**
- **Advanced setup**

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

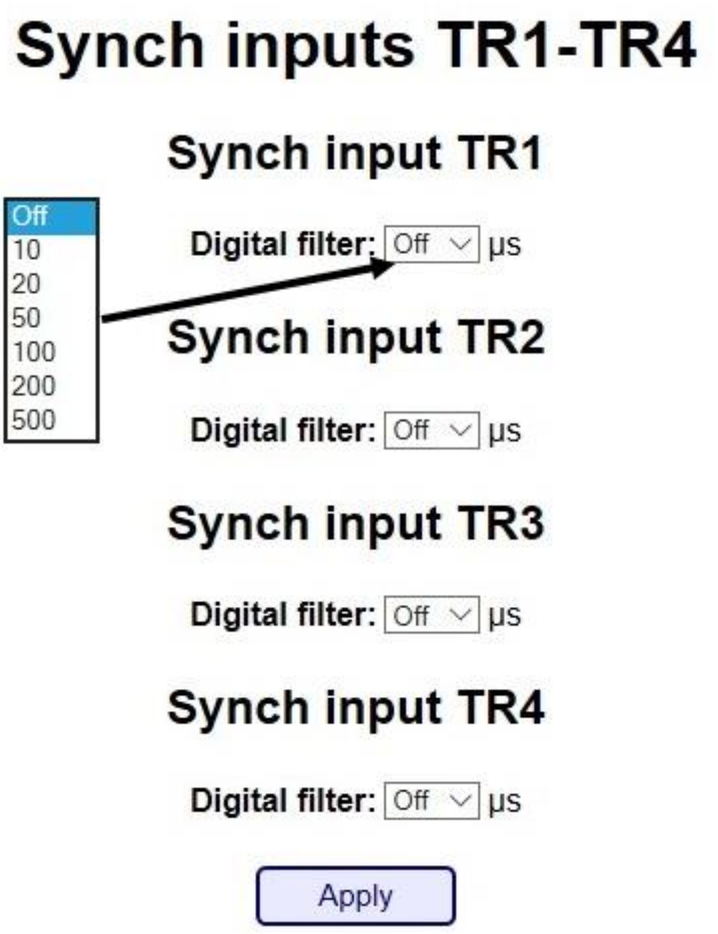
- **Version information**:
  - **Device type**: model of the controller
  - **Bootloader firmware version**: version of the MCU bootloader firmware
  - **MCU firmware version**: version of the MCU application firmware
  - **FPGA firmware version**: version of the FPGA firmware
  - **Board version**: version of the hardware board
- **Current state**:
  - **Board temperature 1**: average temperature of LD1 and LD2 power stages
  - **Board temperature 2**: average temperature of LD3 and LD4 power stages
  - **Remote temperature**: temperature measured by the remote sensor, if present
  - **Supply voltage**: actual supply voltage of the controller
  - **Measured current LD1**: measured current on light output LD1
  - **Measured current LD2**: measured current on light output LD2
  - **Measured current LD3**: measured current on light output LD3
  - **Measured current LD4**: measured current on light output LD4
  - **Measured voltage LD1**: measured voltage on light output LD1
  - **Measured voltage LD2**: measured voltage on light output LD2
  - **Measured voltage LD3**: measured voltage on light output LD3
  - **Measured voltage LD4**: measured voltage on light output LD4

If the 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 TR1-TR4

This page allows to inspect and change all the settings related to the four synchronization inputs. The **Setup synch inputs TR1-TR4** page is shown in the image of *Figure 13: the **Setup synch inputs TR1-TR4** web page*.



Synch inputs TR1-TR4

Synch input TR1

Digital filter: Off µs

Synch input TR2

Digital filter: Off µs

Synch input TR3

Digital filter: Off µs

Synch input TR4

Digital filter: Off µs

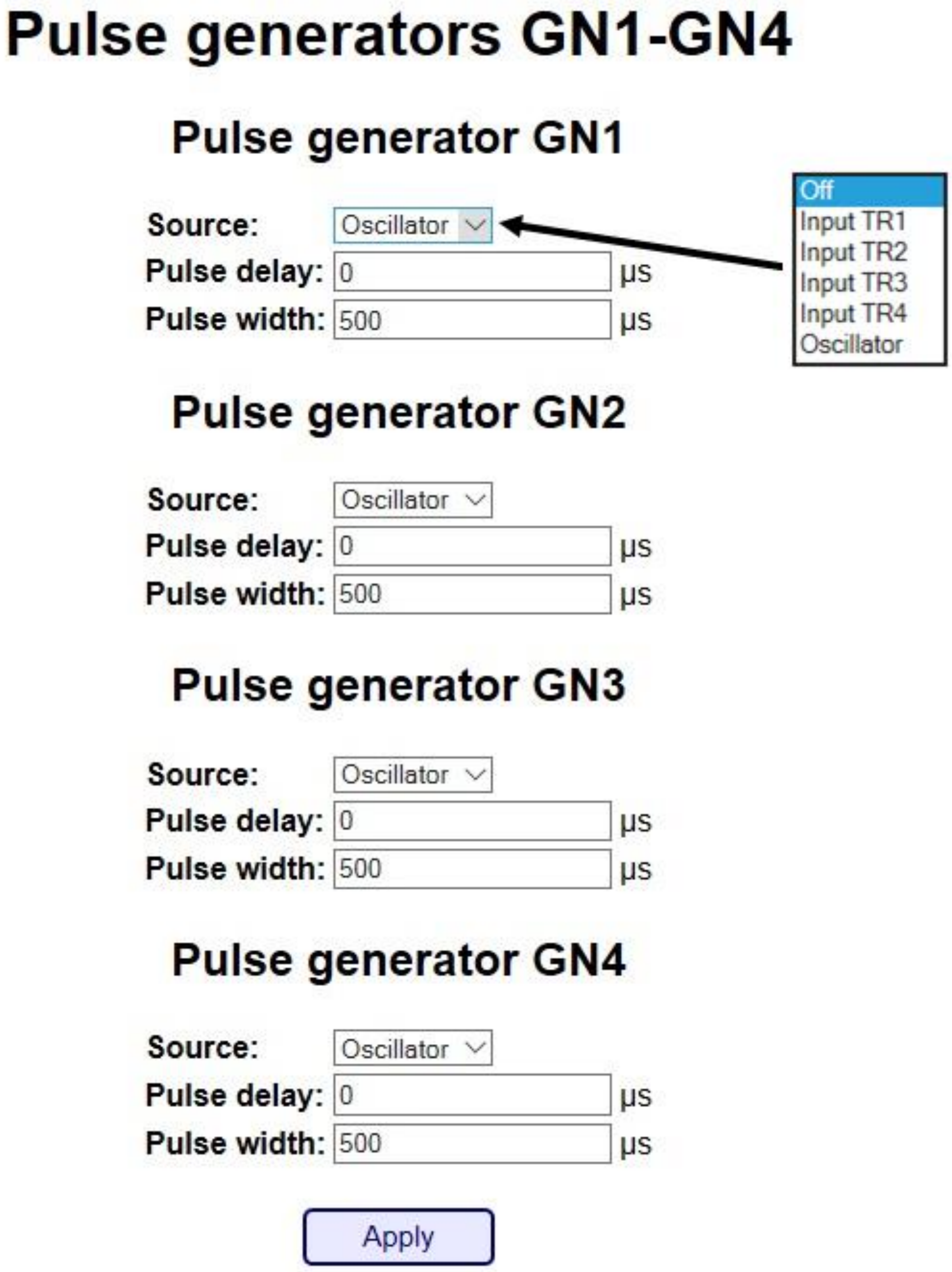
*Figure 13: the **Setup synch inputs TR1-TR4** web page*

Each synchronization input has a digital filter that can be set independently. Each of the four 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 GN1-GN4

This page allows to inspect and change all the settings related to the pulse generators from 1 to 4. The **Setup pulse generators GN1-GN4** page is shown in the image of *Figure 14: the **Setup pulse generators GN1-GN4** web page*.



Pulse generators GN1-GN4

Pulse generator GN1

Source: Oscillator
Pulse delay: 0 µs
Pulse width: 500 µs

Pulse generator GN2

Source: Oscillator
Pulse delay: 0 µs
Pulse width: 500 µs

Pulse generator GN3

Source: Oscillator
Pulse delay: 0 µs
Pulse width: 500 µs

Pulse generator GN4

Source: Oscillator
Pulse delay: 0 µs
Pulse width: 500 µs

*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 and pulse width.

- **Source** selects the trigger for the generator. It can be chosen between **Off** (default), **Input TR1**, **Input TR2**, **Input TR3**, **Input TR4** and **Oscillator**. When set to **Off** the generator is disabled. When set to **Oscillator** the internal free running oscillator is used as the trigger
- **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

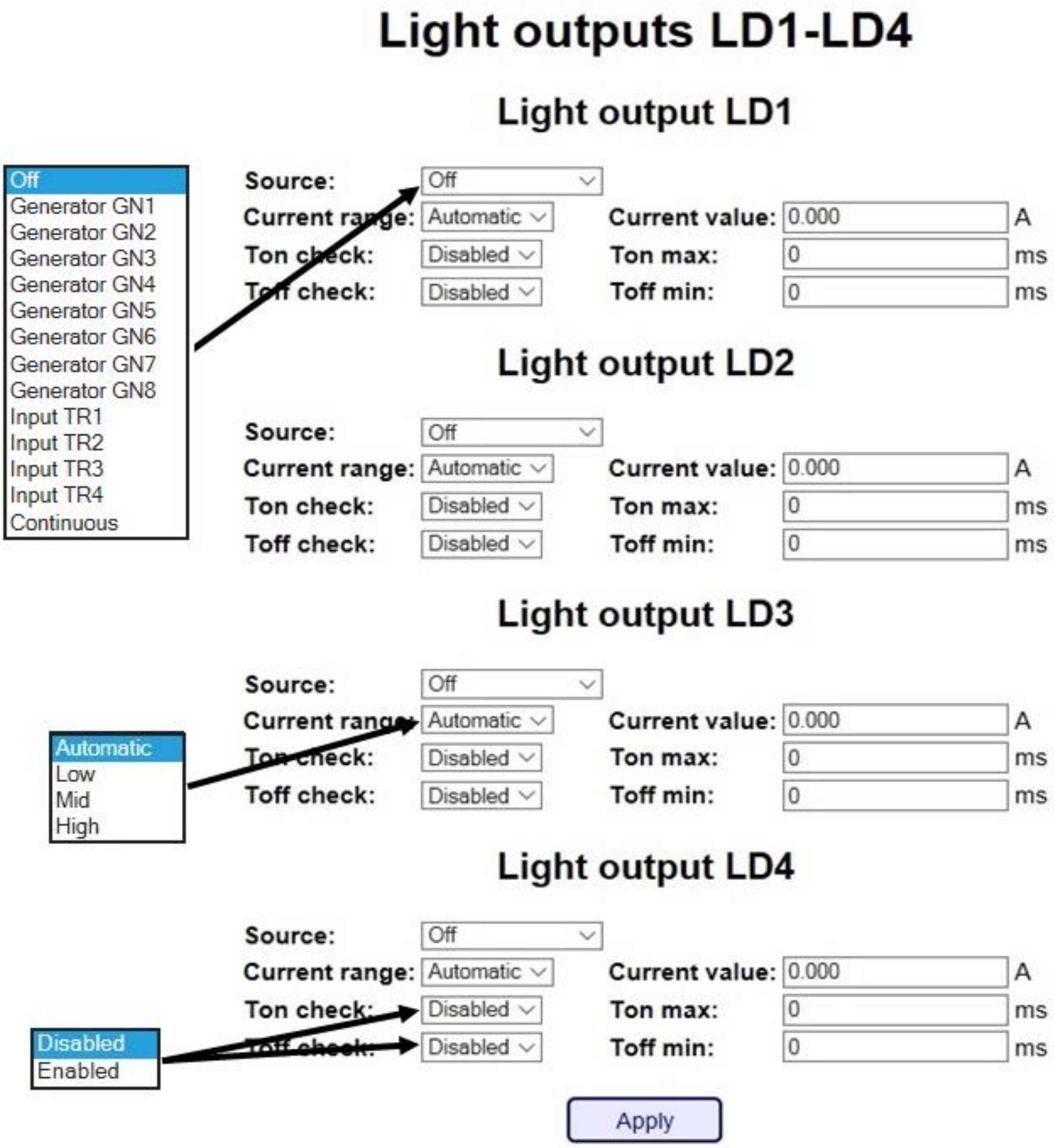
Settings **Pulse delay** and **Pulse width** are expressed in µs.

14.3.4. Setup pulse generators GN5-GN8

This page allows to inspect and change all the settings related to the pulse generators from 5 to 8. Please refer to the description of page **Setup pulse generators GN1-GN4** in chapter 14.3.3 for more information.

14.3.5. Setup light outputs LD1-LD4

This page allows to inspect and change all the settings related to the light outputs from 1 to 4. The **Setup light outputs LD1-LD4** page is shown in the image of *Figure 15: the **Setup light outputs LD1-LD4** web page*.



Light outputs LD1-LD4

Light output LD1

Source: Off
Current range: Automatic | Current value: 0.000 A
Ton check: Disabled | Ton max: 0 ms
Toff check: Disabled | Toff min: 0 ms

Light output LD2

Source: Off
Current range: Automatic | Current value: 0.000 A
Ton check: Disabled | Ton max: 0 ms
Toff check: Disabled | Toff min: 0 ms

Light output LD3

Source: Off
Current range: Automatic | Current value: 0.000 A
Ton check: Disabled | Ton max: 0 ms
Toff check: Disabled | Toff min: 0 ms

Light output LD4

Source: Off
Current range: Automatic | Current value: 0.000 A
Ton check: Disabled | Ton max: 0 ms
Toff check: Disabled | Toff min: 0 ms

*Figure 15: the **Setup light outputs LD1-LD4** web page*

Each light output can be set independently. 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**, **Generator GN5**, **Generator GN6**, **Generator GN7**, **Generator GN8**, **Input TR1**, **Input TR2**, **Input TR3**, **Input TR4** 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 200 mA), **Mid** range (from

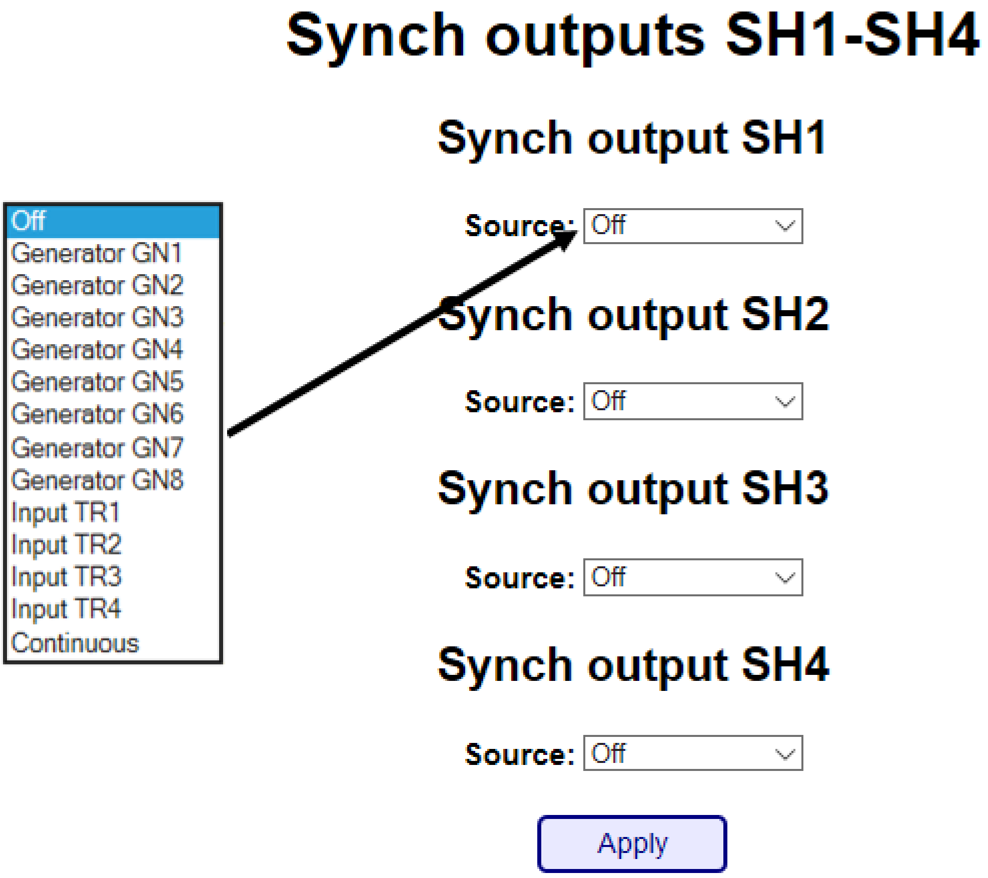
zero up to 4 A) and **High** range (from zero up to 20 A)

- **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. 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

Settings **Ton max** and **Toff min** are expressed in ms.

14.3.6. Setup synch outputs SH1-SH4

This page allows to inspect and change all the settings related to the synchronization outputs from 1 to 4. The **Setup synch outputs SH1-SH4** page is shown in the image of *Figure 16: the **Setup synch outputs SH1-SH4** web page*.



Synch outputs SH1-SH4

Synch output SH1

Source: Off

Synch output SH2

Source: Off

Synch output SH3

Source: Off

Synch output SH4

Source: Off

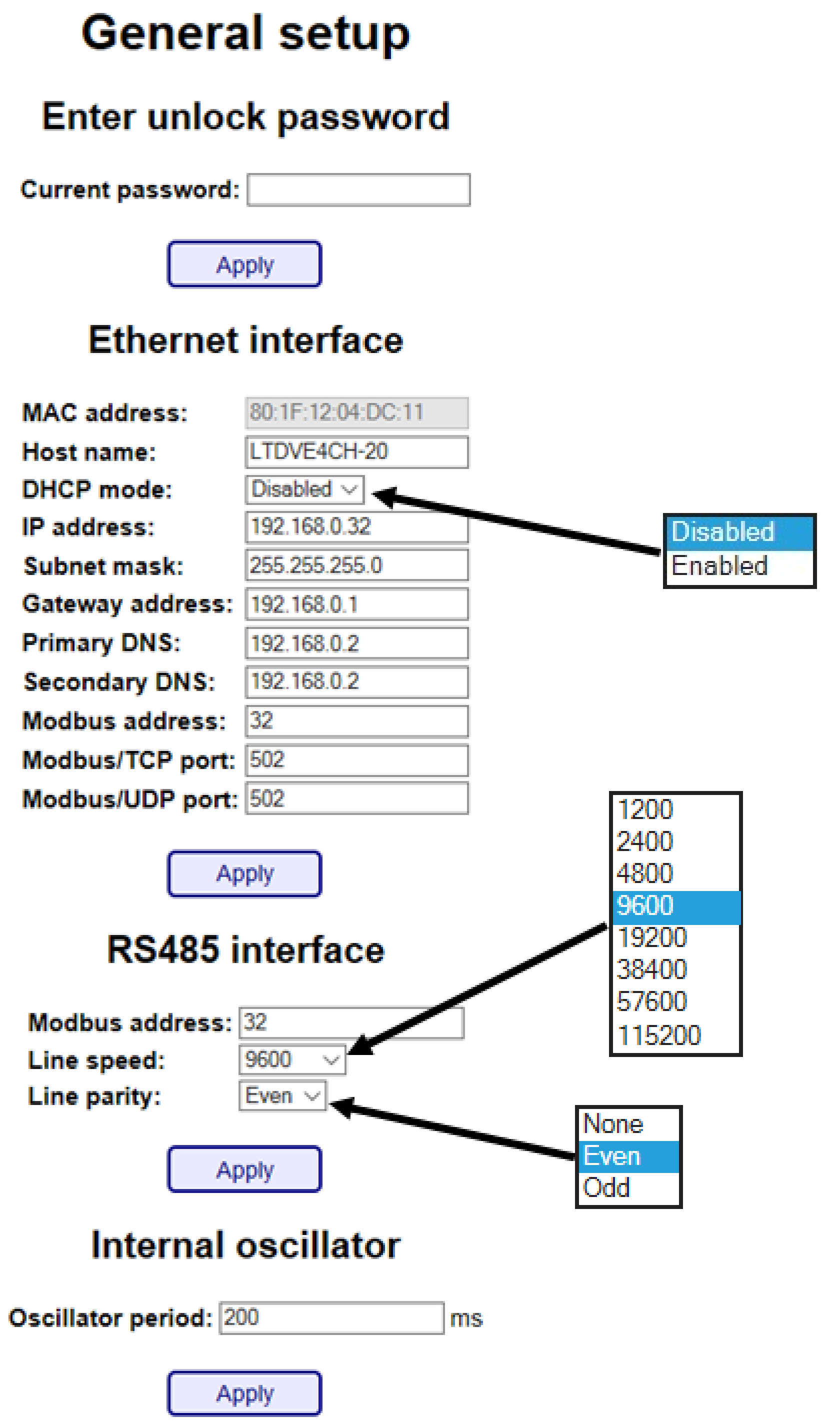
*Figure 16: the **Setup synch outputs SH1-SH4** web page*

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, **Generator GN5**, **Generator GN6**, **Generator GN7**, **Generator GN8, Input TR1**, **Input TR2**, **Input TR3**, **Input TR4** and **Continuous**. When set to **Off** the synchronization output is disabled. When set to **Continuous** the synchronization output is always active.

14.3.7. 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: 80:1F:12:04:DC:11
Host name: LTDVE4CH-20
DHCP mode: Disabled
IP address: 192.168.0.32
Subnet mask: 255.255.255.0
Gateway address: 192.168.0.1
Primary DNS: 192.168.0.2
Secondary DNS: 192.168.0.2
Modbus address: 32
Modbus/TCP port: 502
Modbus/UDP port: 502

RS485 interface

Modbus address: 32
Line speed: 9600
Line parity: Even

Internal oscillator

Oscillator period: 200 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 14.3.8).

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

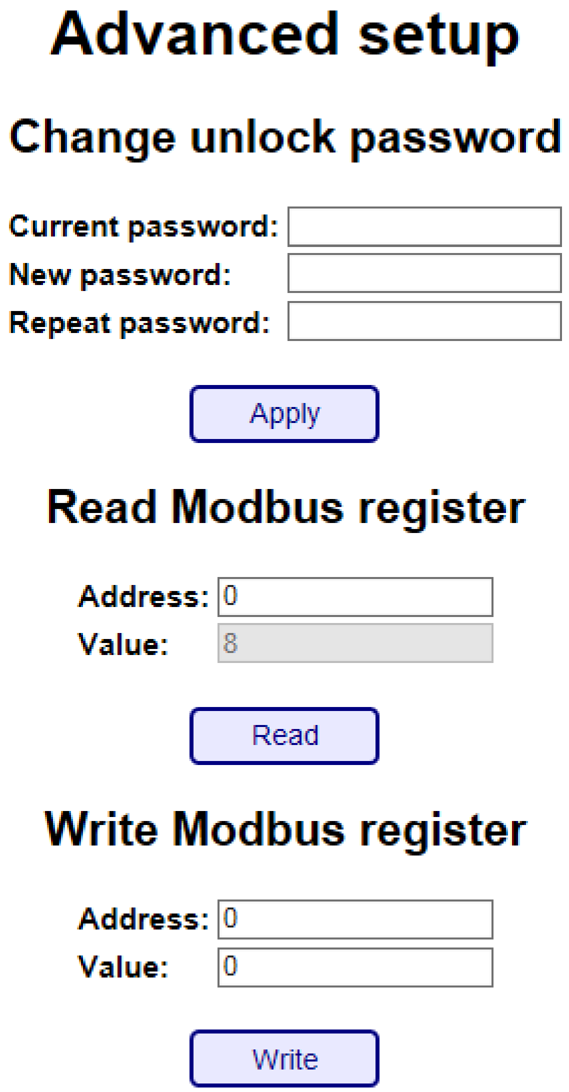
Note the current firmware supports only 9600 bits per second and even parity.

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)

14.3.8. 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:

Read Modbus register

Address: 0
Value: 8

Write Modbus register

Address: 0
Value: 0

*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.

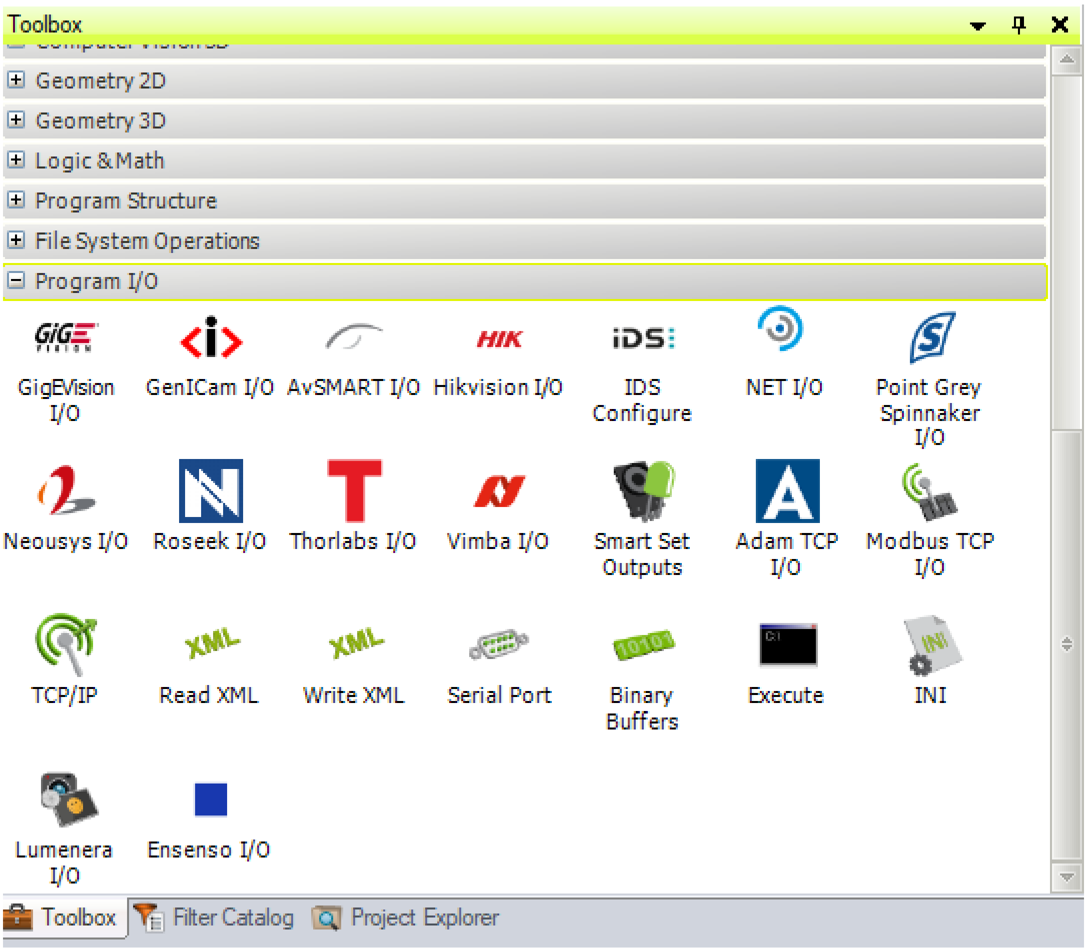
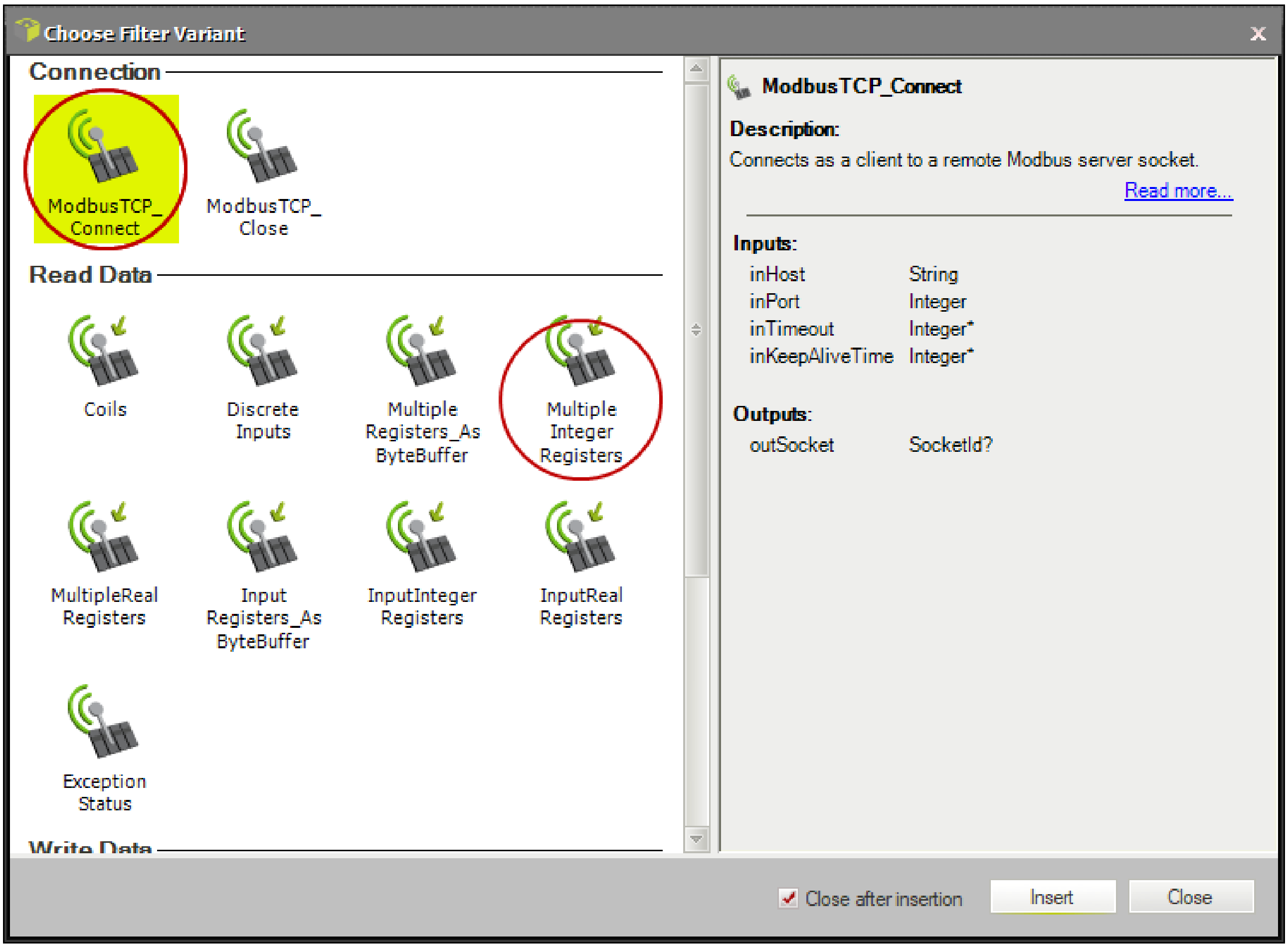


*Figure 19. The **Program I/O** group of filters in FabImage*

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.



ModbusTCP_Connect

Description:
Connects as a client to a remote Modbus server socket.
Read more...

Inputs:

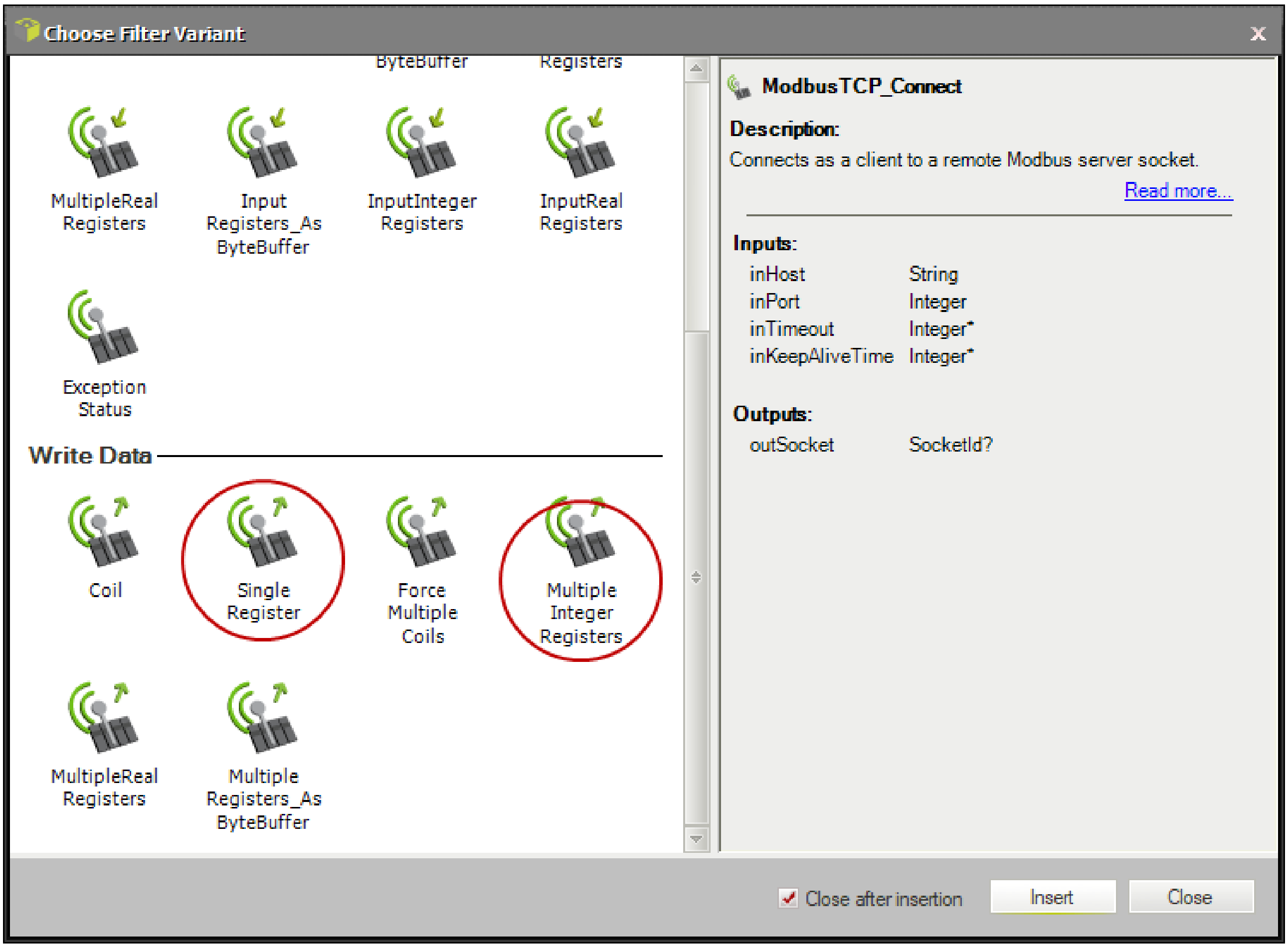
inHost	String
inPort	Integer
inTimeout	Integer*
inKeepAliveTime	Integer*

Outputs:

outSocket	SocketId?

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.



ModbusTCP_Connect

Description:

Connects as a client to a remote Modbus server socket.

Read more...

Inputs:

inHost	String
inPort	Integer
inTimeout	Integer*
inKeepAliveTime	Integer*

Outputs:

outSocket	SocketId?

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.

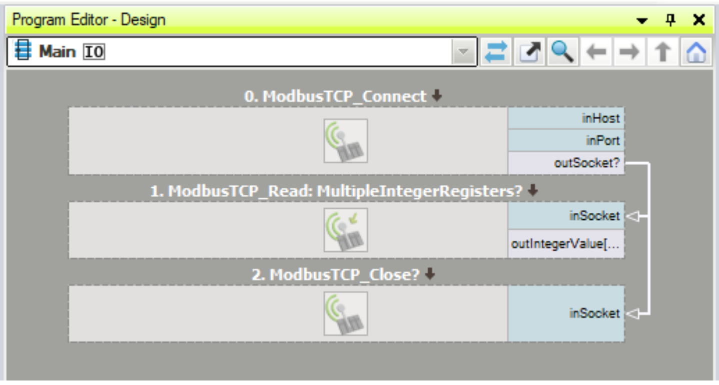
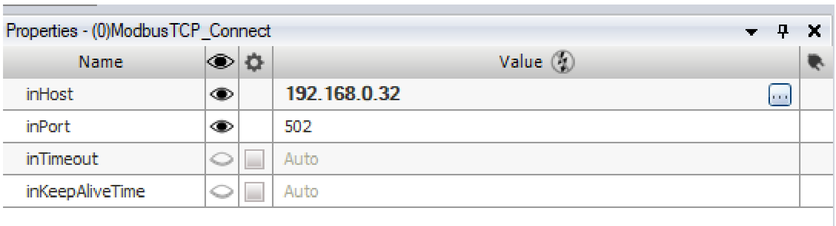


Figure 22. Example of data-flow code for reading a register

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.

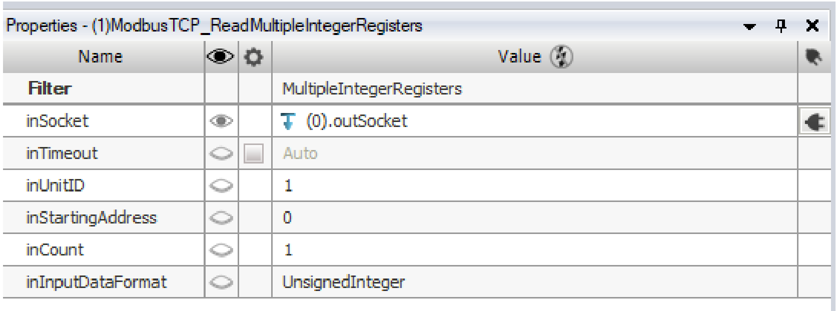


Properties - (0)ModbusTCP_Connect

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.



Properties - (1)ModbusTCP_ReadMultipleIntegerRegisters

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.

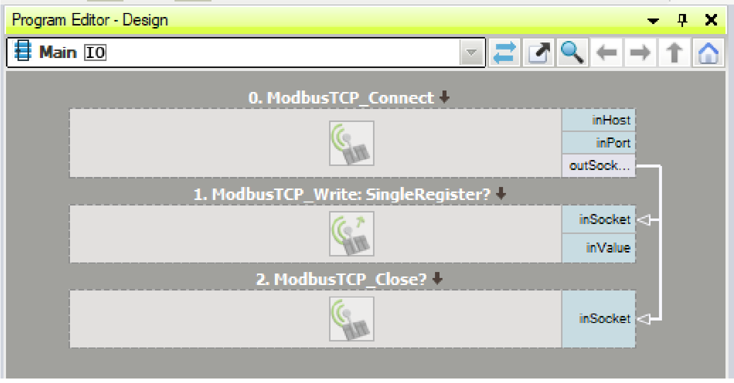
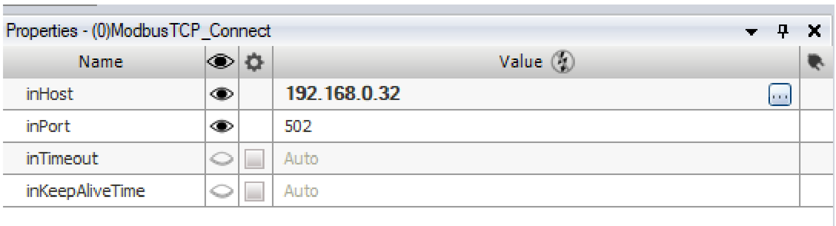


Figure 25. Example of data-flow code for writing a register

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.

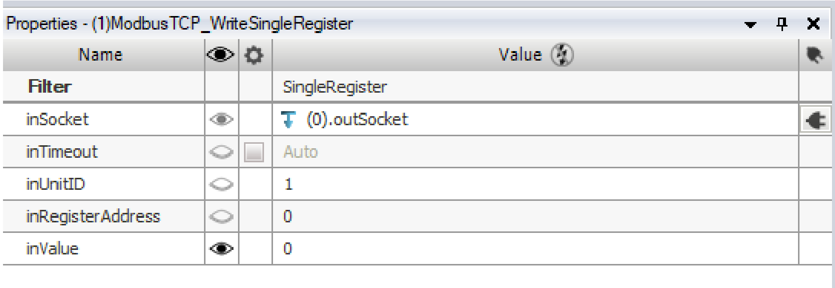
Configure the **ModbusTCP_Connect** filter as shown in the following figure.


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).

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 P4. Data signal D+ is pin 25, data signal D- is pin 12 and the reference ground GND is pin 13.

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.

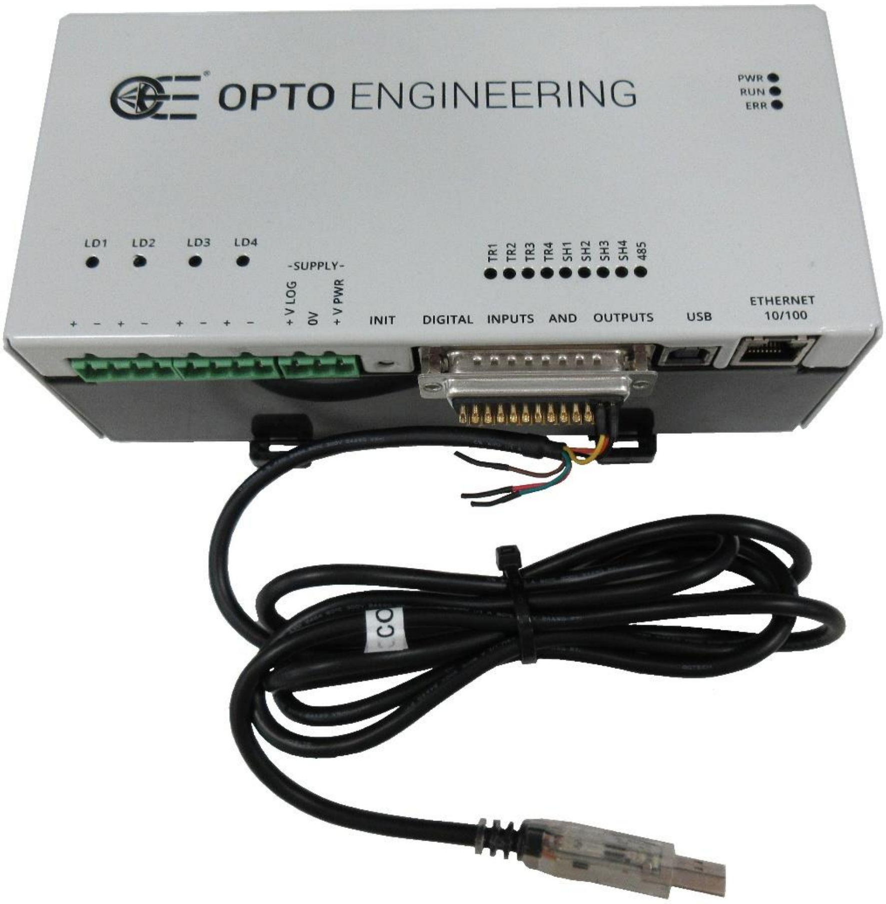


Figure 28: connection of RS485-USB adapter to the controller

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

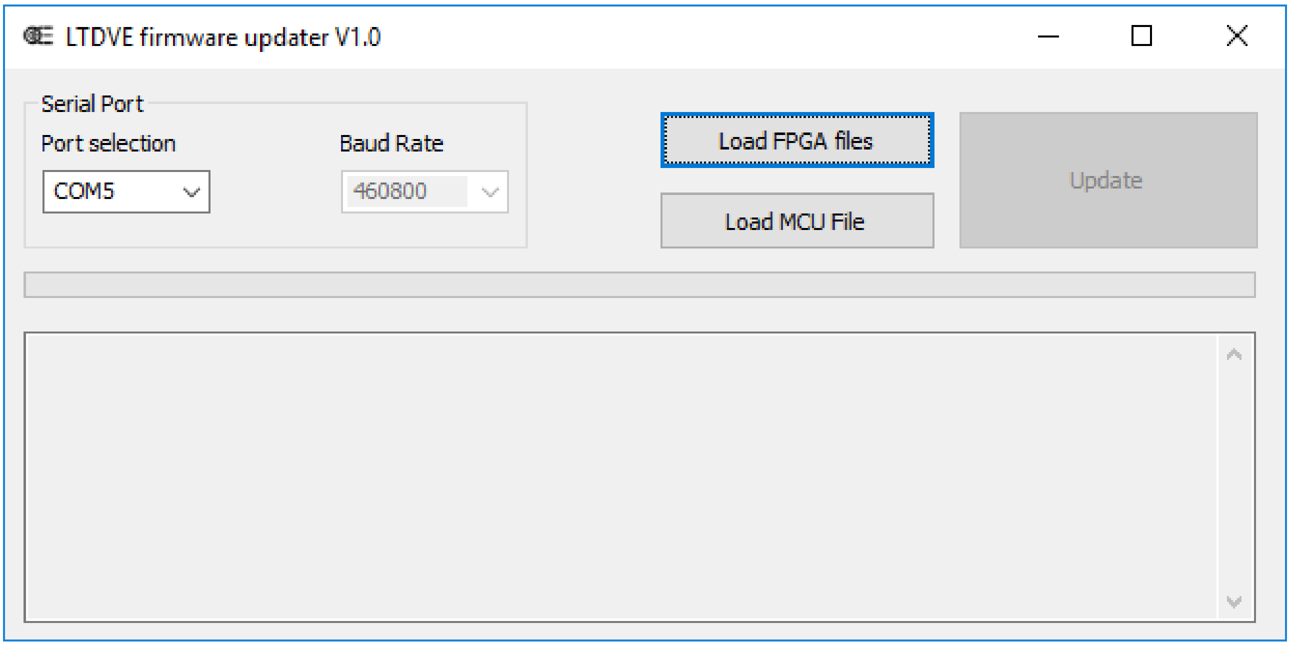
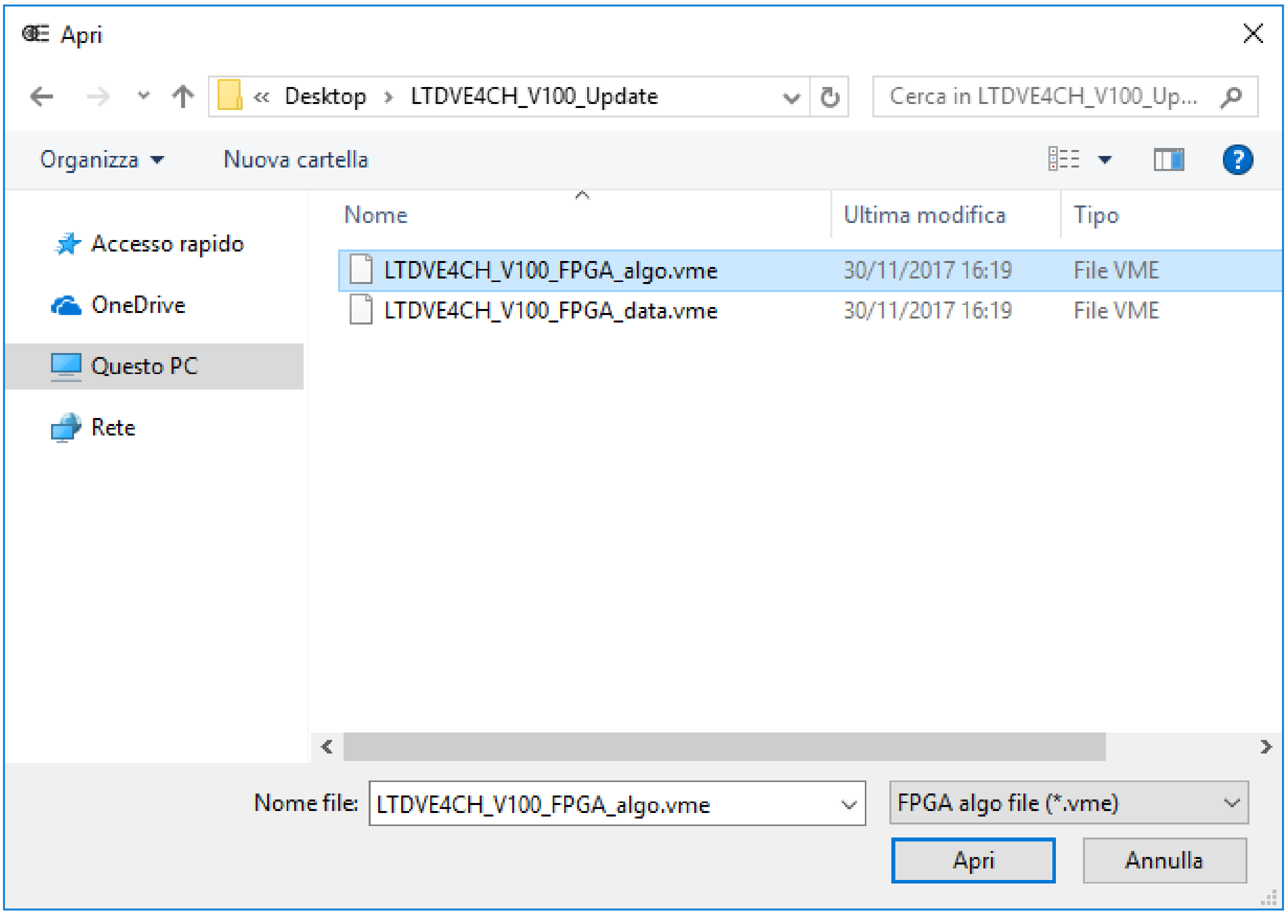


*Figure 29: main window of **LTDVE firmware updater** application*

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.



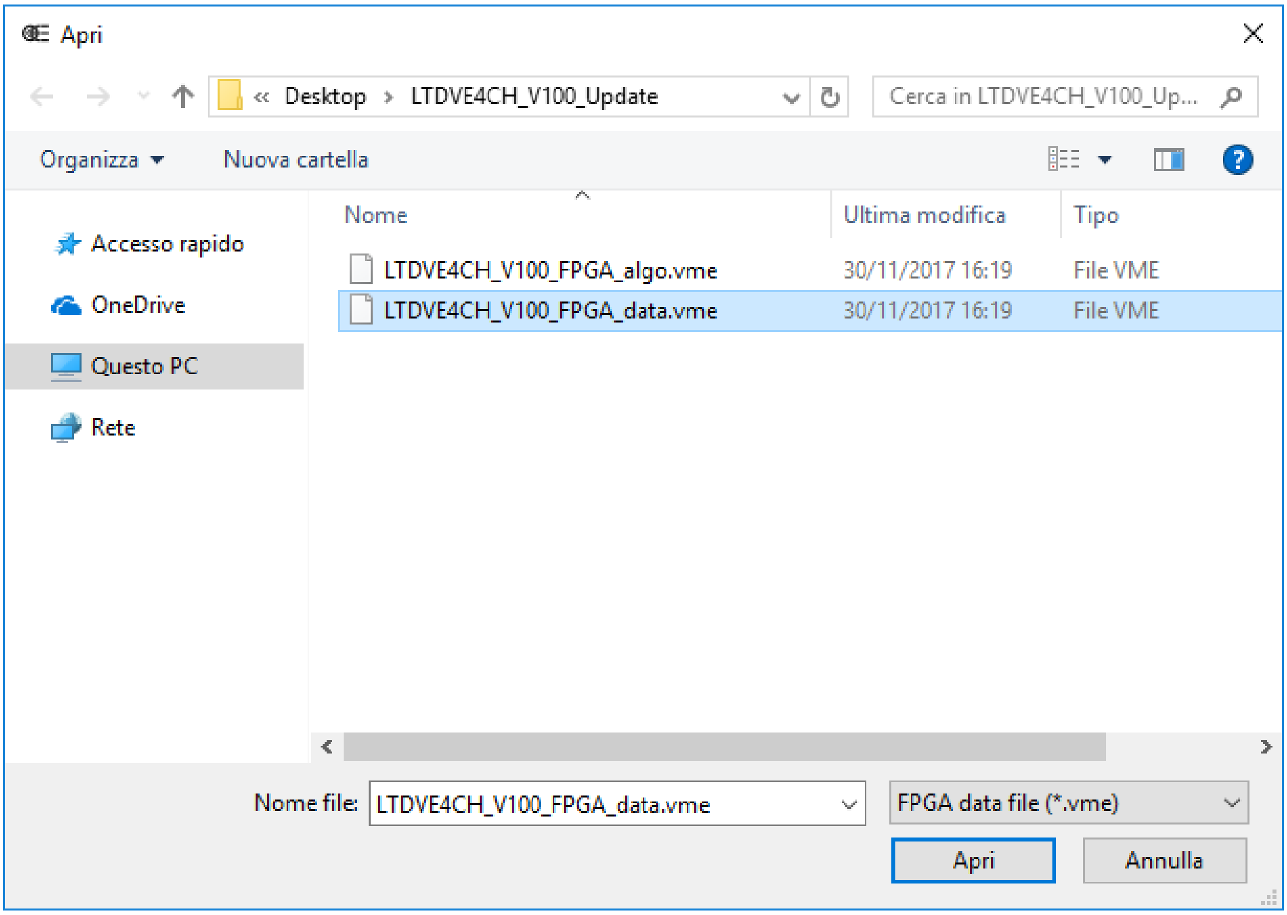
Nome	Ultima modifica	Tipo
LTDVE4CH_V100_FPGA_algo.vme	30/11/2017 16:19	File VME
LTDVE4CH_V100_FPGA_data.vme	30/11/2017 16:19	File VME

Nome file: LTDVE4CH_V100_FPGA_algo.vme

FPGA algo file (*.vme)

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.


Apri

« Desktop › LTDVE4CH_V100_Update

Cerca in LTDVE4CH_V100_Up...

Organizza ▾ Nuova cartella

Accesso rapido
OneDrive
Questo PC
Rete

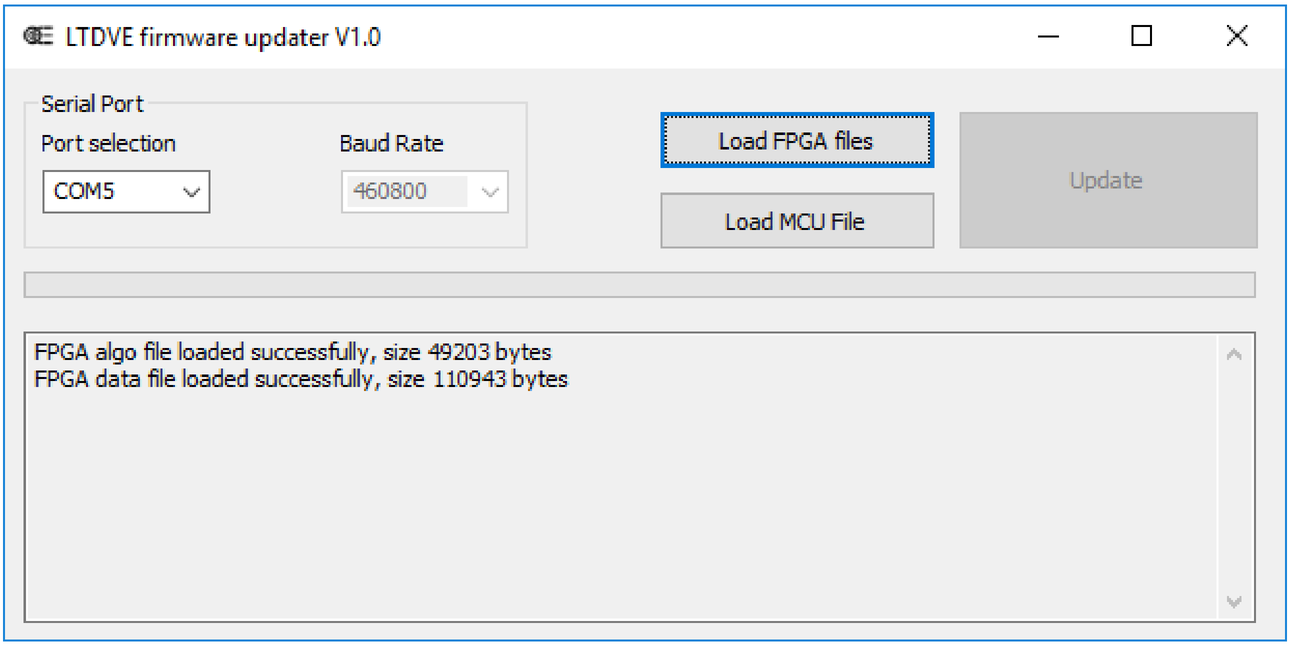
Nome	Ultima modifica	Tipo
LTDVE4CH_V100_FPGA_algo.vme	30/11/2017 16:19	File VME
LTDVE4CH_V100_FPGA_data.vme	30/11/2017 16:19	File VME

Nome file: LTDVE4CH_V100_FPGA_data.vme

FPGA data file (*.vme)

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

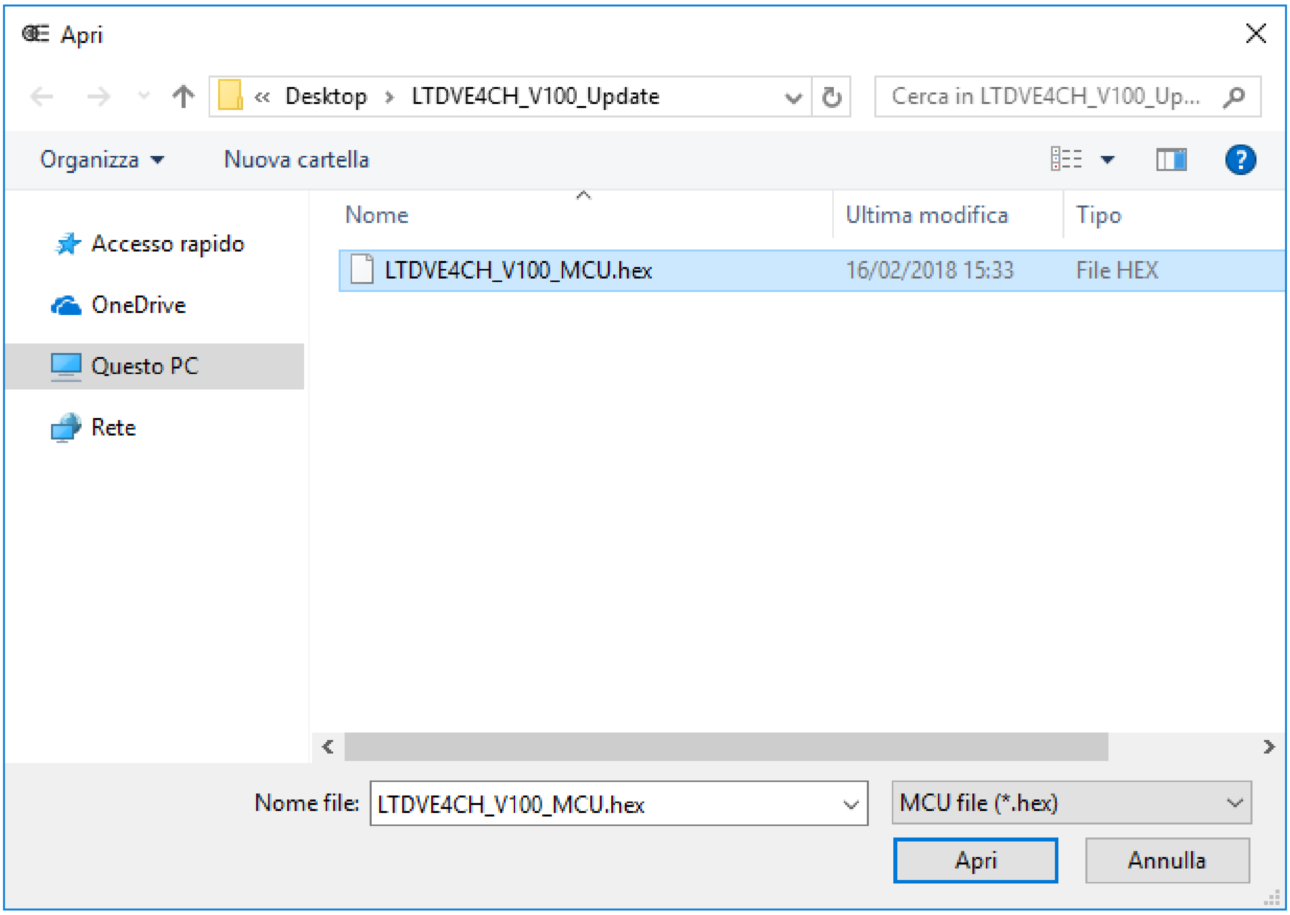
After the FPGA Data file has been specified, the relative modal dialog is closed. The couple of FPGA firmware files will then be loaded from the PC disk to the PC RAM. The text box at the bottom of the main window will then be updated with some information regarding the loading process (see image in *Figure 32: main window after the FPGA firmware files have been loaded*).



```
FPGA algo file loaded successfully, size 49203 bytes
FPGA data file loaded successfully, size 110943 bytes
```

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.



« Desktop › LTDVE4CH_V100_Update

Nome	Ultima modifica	Tipo
LTDVE4CH_V100_MCU.hex	16/02/2018 15:33	File HEX

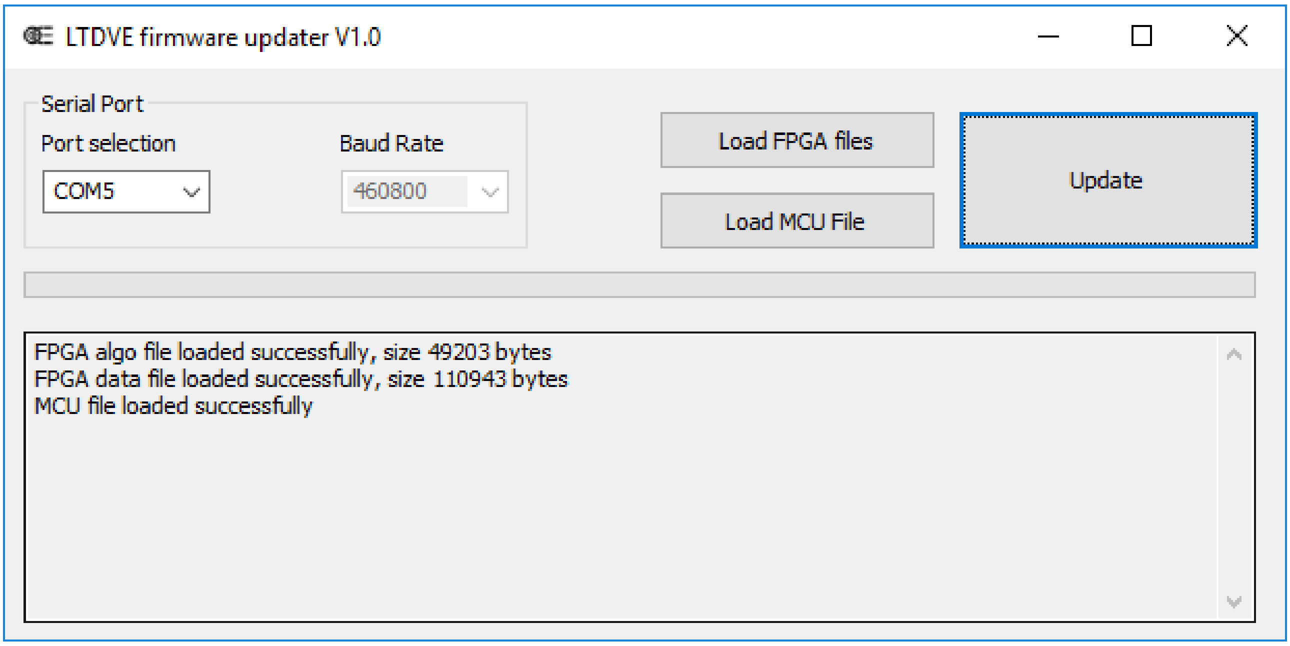
Nome file: LTDVE4CH_V100_MCU.hex

MCU file (*.hex)

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.



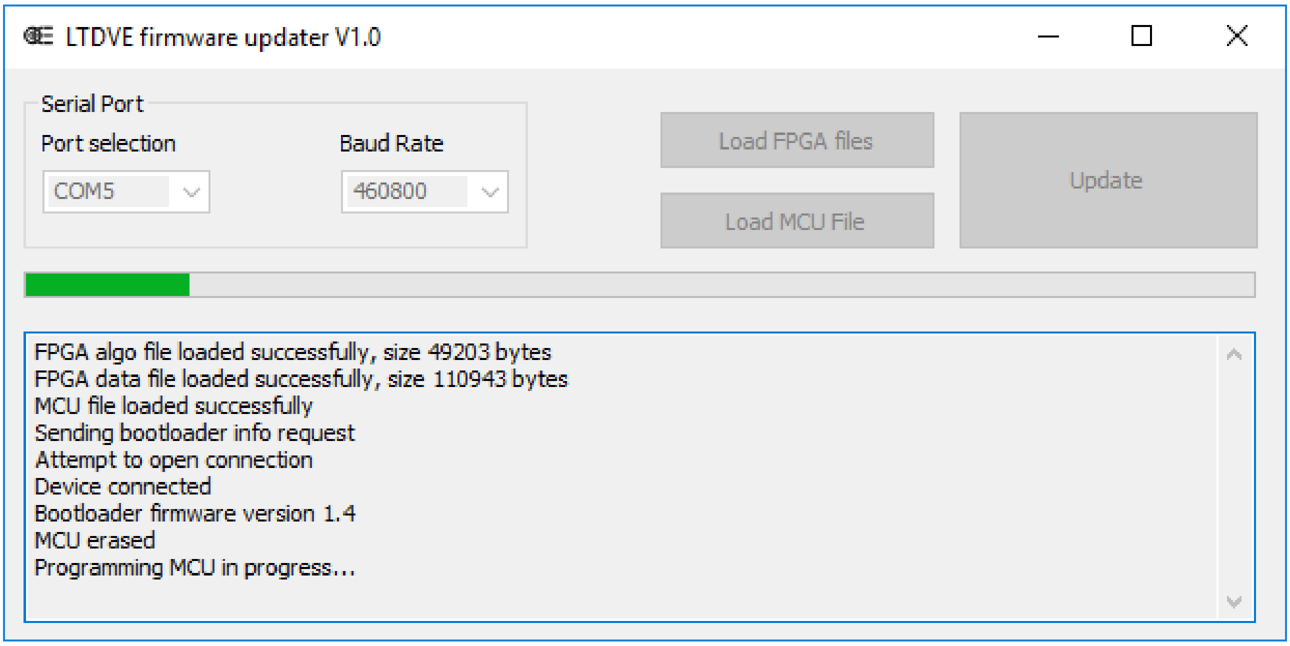
```
FPGA algo file loaded successfully, size 49203 bytes
FPGA data file loaded successfully, size 110943 bytes
MCU file loaded successfully
```

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.

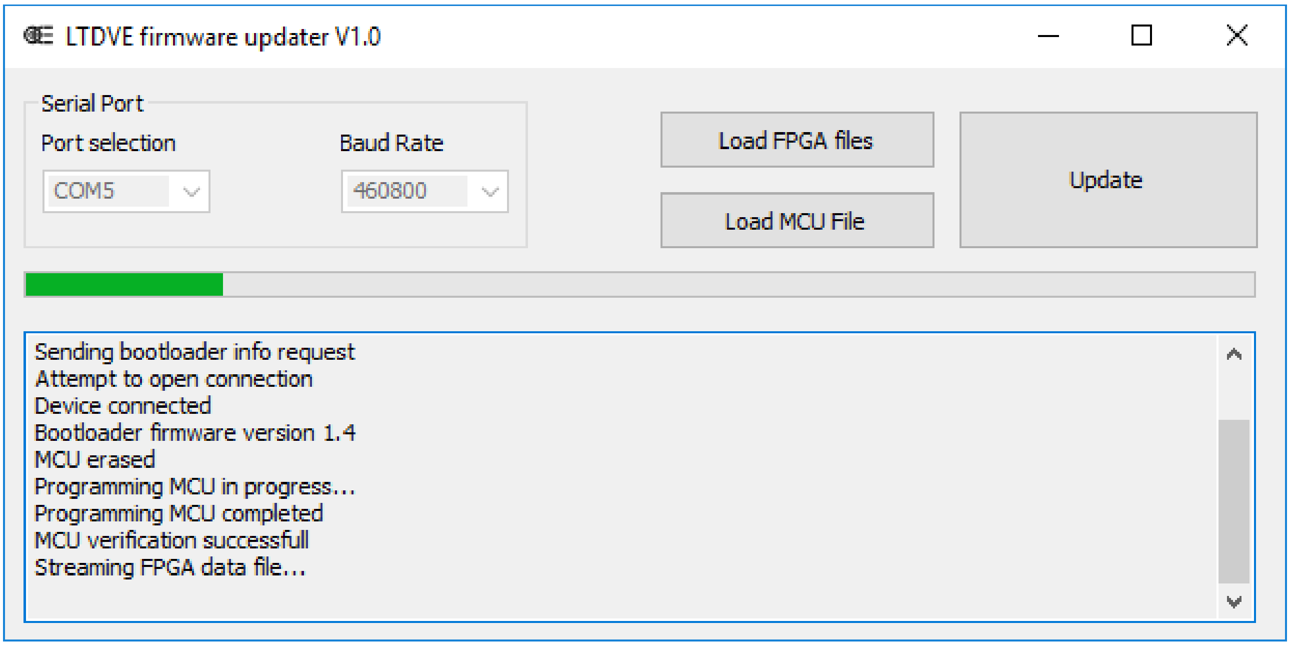
The first data transferred to the controller is the MCU firmware (see image in *Figure 35: MCU firmware update sequence*).



```
FPGA algo file loaded successfully, size 49203 bytes
FPGA data file loaded successfully, size 110943 bytes
MCU file loaded successfully
Sending bootloader info request
Attempt to open connection
Device connected
Bootloader firmware version 1.4
MCU erased
Programming MCU in progress...
```

Figure 35: MCU firmware update sequence

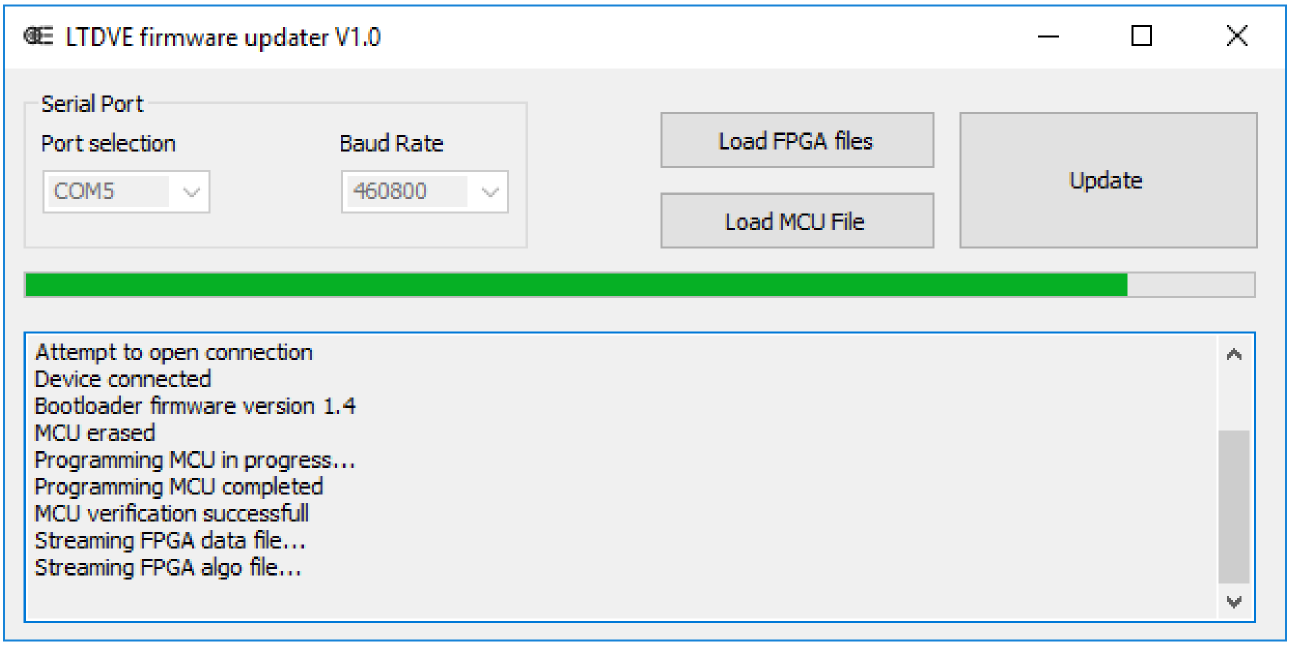
Next, the FPGA Data firmware is transferred (see image in *Figure 36: FPGA Data firmware update sequence*).



```
Sending bootloader info request
Attempt to open connection
Device connected
Bootloader firmware version 1.4
MCU erased
Programming MCU in progress...
Programming MCU completed
MCU verification successfull
Streaming FPGA data file...
```

Figure 36: FPGA Data firmware update sequence

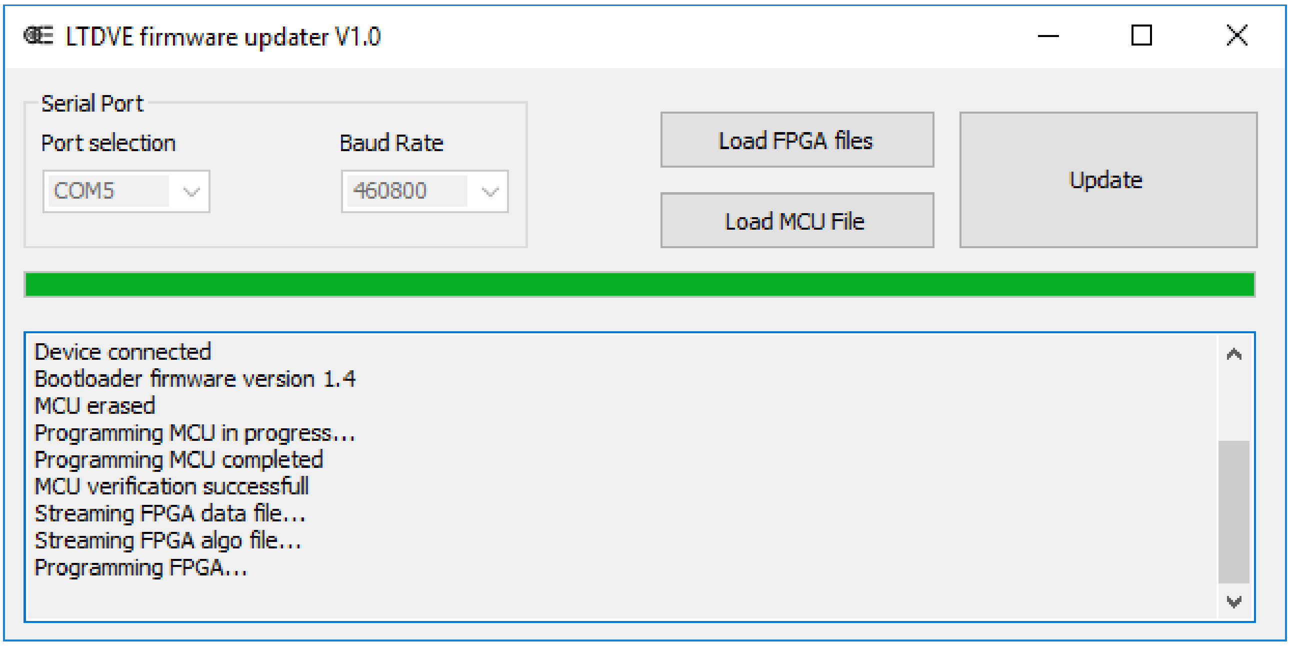
Last, the FPGA Algo firmware is transferred (see image in *Figure 37: FPGA Algo firmware update sequence*).



```
Attempt to open connection
Device connected
Bootloader firmware version 1.4
MCU erased
Programming MCU in progress...
Programming MCU completed
MCU verification successfull
Streaming FPGA data file...
Streaming FPGA algo file...
```

Figure 37: FPGA Algo firmware update sequence

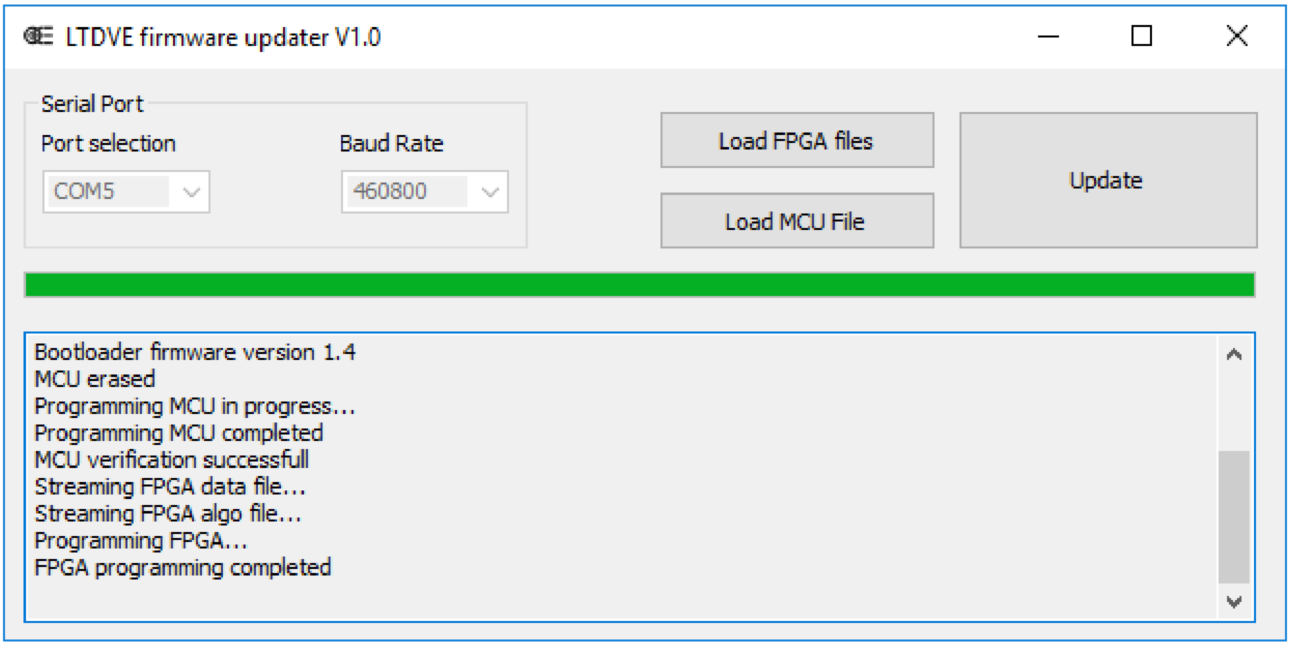
The FPGA is programmed just after both the FPGA Data and the FPGA Algo files have been transferred (see image in *Figure 38: main window after FPGA programming*).



```
Device connected
Bootloader firmware version 1.4
MCU erased
Programming MCU in progress...
Programming MCU completed
MCU verification successfull
Streaming FPGA data file...
Streaming FPGA algo file...
Programming FPGA...
```

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.



```
Bootloader firmware version 1.4
MCU erased
Programming MCU in progress...
Programming MCU completed
MCU verification successfull
Streaming FPGA data file...
Streaming FPGA algo file...
Programming FPGA...
FPGA programming completed
```

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.

18. Optional version with improved cooling

The LTDVE4CH-20-HS-01-FC is an alternative version of the LTDVE4CH-20 controller. The only difference compared to the LTDVE4CH-20 is a fan applied to the side of the heat sink to better dissipate the generated heat. The LTDVE4CH-20-HS-01-FC controller is shown in *Figure 40: picture of LTDVE4CH-20-HS-01-FC*.

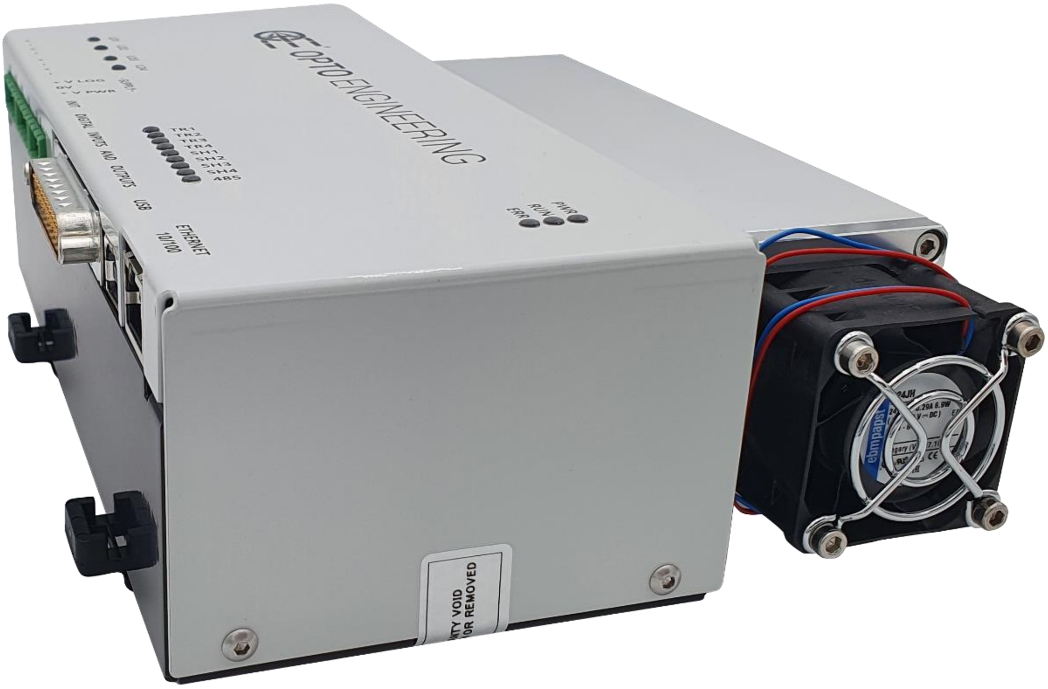


Figure 40: picture of LTDVE4CH-20-HS-01-FC

The fan is powered with two wires. Fan pinout is defined in *Table 16: pinout of fan*.

Wire	Colour	Description	Wire size
1	Red	Positive power supply	AWG 28
2	Blue	Negative power supply	AWG 28

Table 16: pinout of fan

The power supply for the fan is 24 V DC. The maximum current is 300 mA, for a maximum power consumption of 7 W. Fan power supply can be derived from the controller power supply if they have the same value.

OPTO ENGINEERING®

EUROPE

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

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

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

UNITED STATES

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

ASIA

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

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

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

WWW.OPTO-E.COM