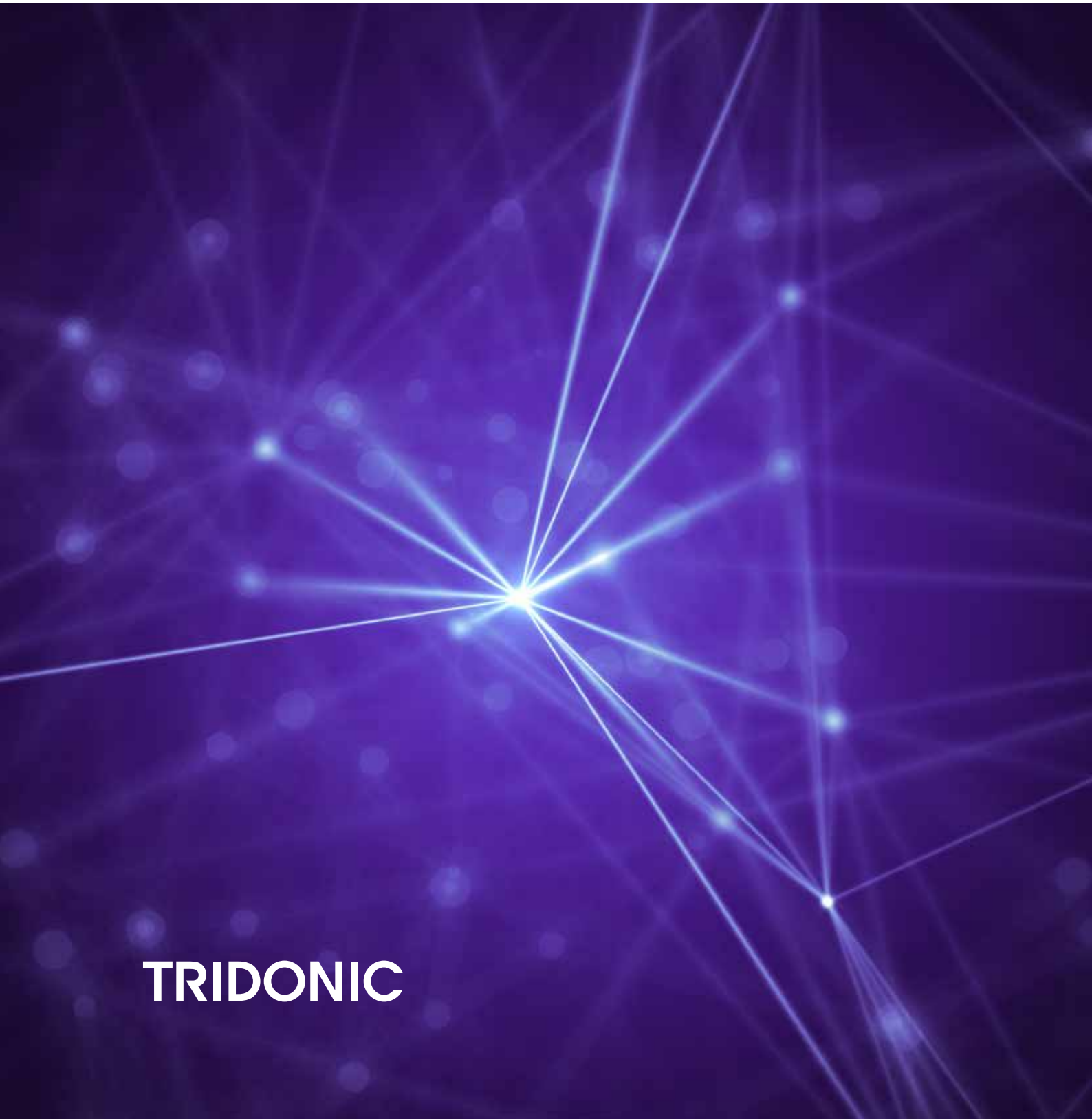
companionSUITE

# deviceCONFIGURATOR

Manual



TRIDONIC

# Table of Contents

# Table of Contents

# Introduction

The deviceCONFIGURATOR software is a tool to configure Tridonic LED Drivers within a luminaire production. Scripts created by the deviceGENERATOR are executed with the deviceCONFIGURATOR to store the parameters in different ways on Tridonic LED Drivers. Additional features of the deviceCONFIGURATOR are the possibility to load scripts via barcode scanner and the printing of labels after successful configuration.

**The following protocols and interfaces are supported:**

| Interface | Interface device |
|---|---|
| NFC | FEIG antennas |
| DALI | ready2mains Programmer / DALI USB |
| ready2mains | ready2mains Programmer |
| U6Me2 | ready2mains Programmer |

To run the deviceCONFIGURATOR Windows software, the following minimum system requirements are needed:

| Parameter | Description |
|---|---|
| Operating system | Windows 7 or higher |
| Minimum display resolution | 1024 x 768 pixels |
| USB | USB 2.0 port for each programming interface, printer or barcode scanner |
| Internet | No internet connection necessary for the deviceCONFIGURATOR software. |

# Setup

Within the companionSUITE InstallShield Wizard several components are available:

_ The deviceCONFIGURATOR software

_ The deviceANALYSER reader

_ The FEIG RFID driver

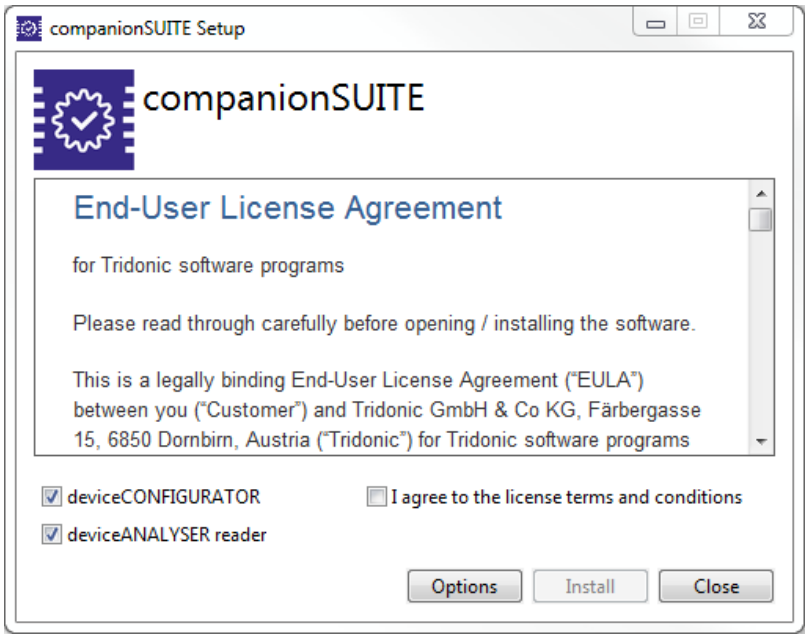
_ The DALI Monitor

NOTICE

It is recommended to uninstall the old companionSUITE version before a new installation.

To install the software, follow these steps:

_ To start the installation, execute the companionSUITE_Setup.exe file.




_ Select the components which should be installed

_ Change the installation folder in the option's sub menu

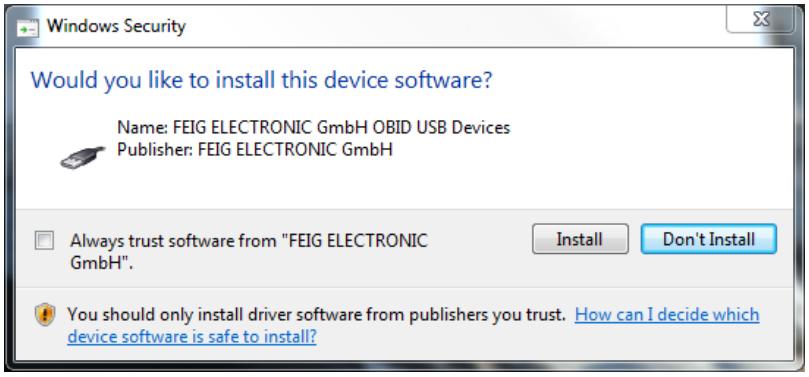
_ Agree to the license terms and conditions

_ Proceed with **Install**

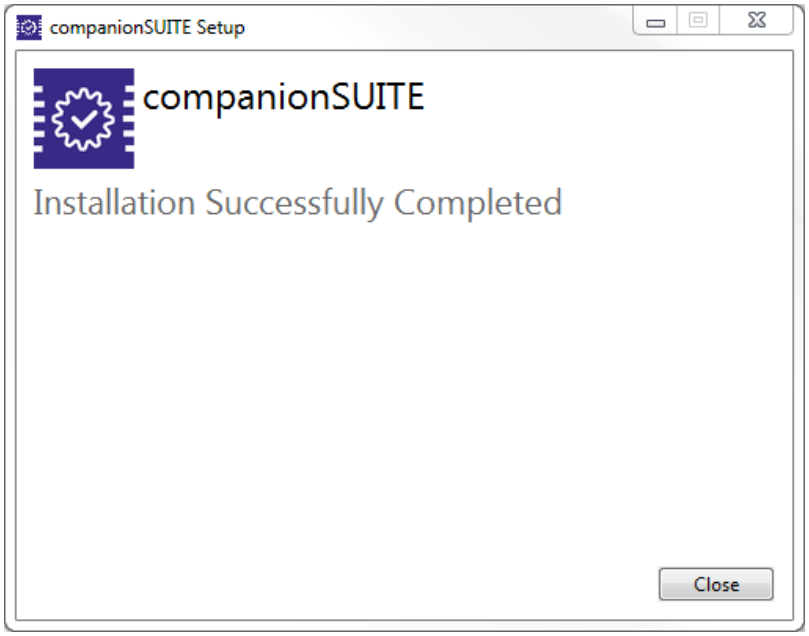
During the installation process, drivers for the necessary hardware are installed.

# Setup

_ Driver for the FEIG NFC antennas:




If all drivers and the software is installed, close the setup.




Afterwards you can launch the deviceCONFIGURATOR or deviceANALYSER reader.

# WORK window

## 4.1. General information


The **WORK** window is the main window from which the LED Driver configuration is controlled.

From here, the configuration files are loaded and executed. It is also possible to lock the **Work** window to prevent changes.

## 4.2. Open file

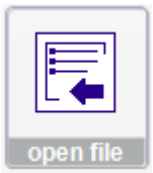
There are two ways to open a configuration file:

_ Via the **open file** button -or-

_ Via barcode

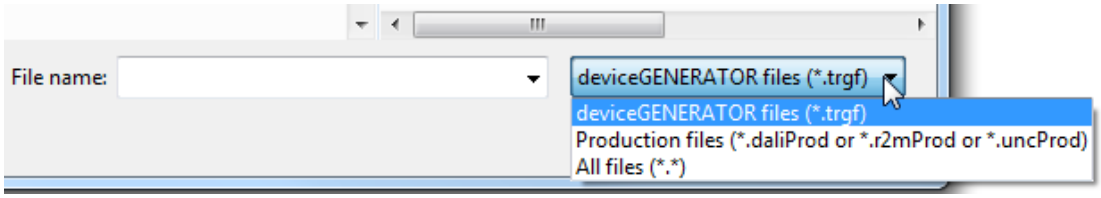
### 4.2.1. Open file via open button

With this button it is possible to load a file from the computer network.



# WORK window

There are several configuration file types available to load with the deviceCONFIGURATOR.
As a standard, the deviceGENERATOR type is selected.
To change to the scriptGENERATOR files, select **Production files**.




**deviceGENERATOR:** Files with ending *.trgf
**scriptGENERATOR:** Files with ending *.daliProd or *.r2mProd or *.uncProd

## 4.2.2. Open script via barcode

To open a configuration file via barcode, the filename must be encrypted in the barcode and the file must be available in the defined folder.
For more information about the use of barcodes and the barcode type, see barcode settings, p. 38.

# WORK window

## 4.3. Lock and unlock the Work window

The lock function is available to secure the deviceCONFIGURATOR against unauthorized changes. If the **Work** window is locked, it is not possible to change anything in the settings.

Locked **Work** window:


If the **Work** window is locked, it is also not possible to minimize or maximize the window. The buttons to do so are not available. It is necessary to set a password in the Lock settings, p. 36!

# WORK window

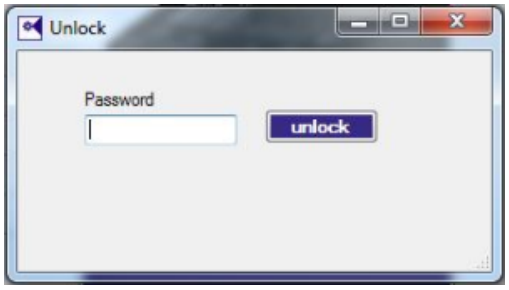
### 4.3.1. Lock the Work window

Lock the window by pressing the **Lock/Open** button:


### 4.3.2. Unlock the Work window

Unlock the window by pressing the **Lock/Open** button again:


If no password has been set, the window is immediately unlocked. If a password has been set, it must be entered in the **Password** window to unlock the deviceCONFIGURATOR.



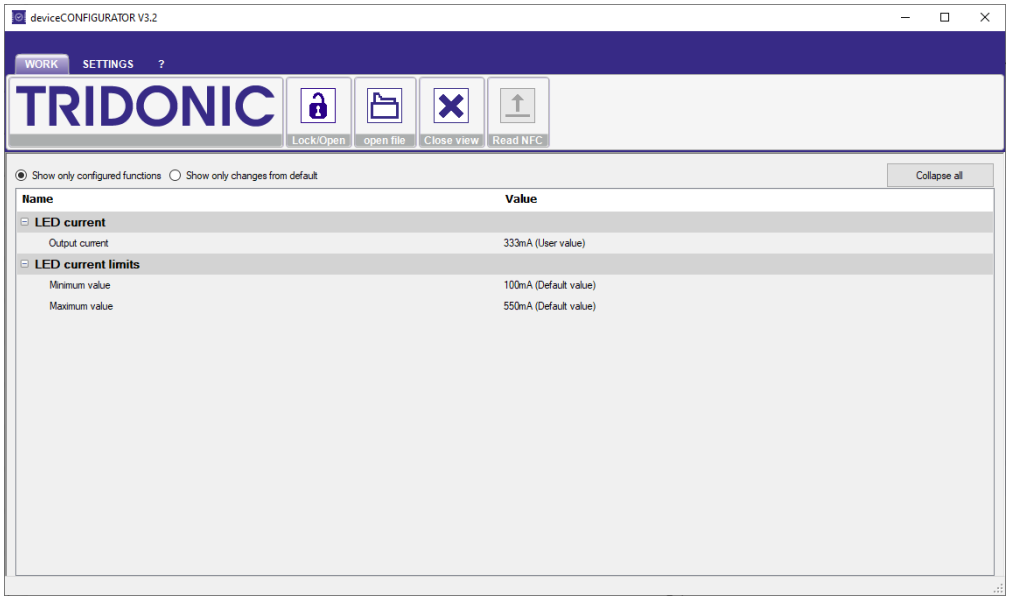

### 4.3.3. Summary

Shows a summary of the currently loaded programming file:

# WORK window


The default view shows which functions are configured with the current loaded programing file.

The option "Show only changes from default" hides the default values that are also written.




## 4.3.4. Close View

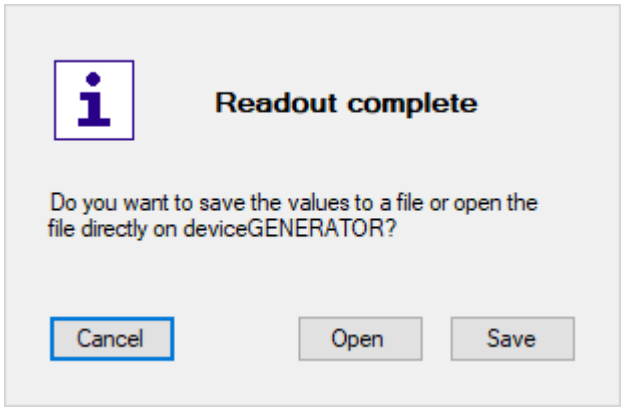
Closes the summary view.


## 4.3.5. Read NFC

Allows to read out a driver with NFC

# WORK window

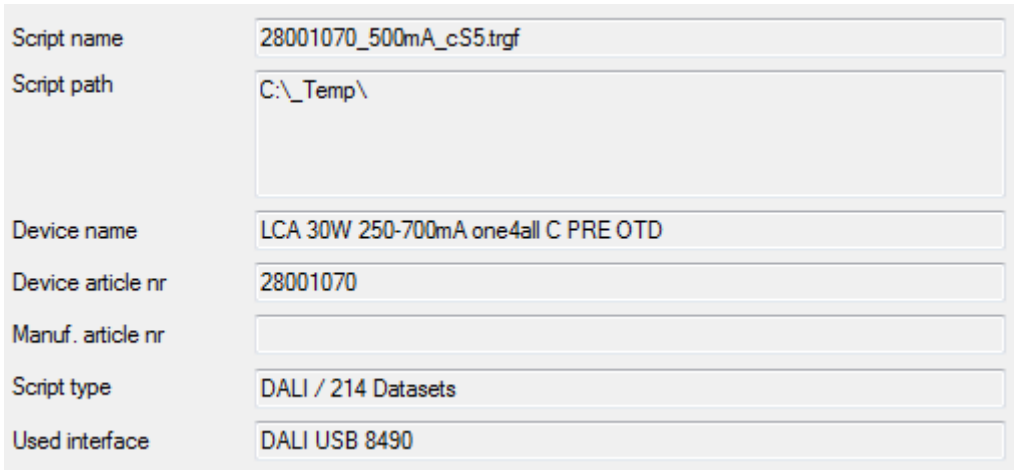

The readout can either be opened which will upload it directly to the deviceGENERATOR or it can be saved locally as a .traf file for later use.




## 4.4. Work window content

### 4.4.1. Script and interface information

This section shows information about the loaded script and the used interface.



# WORK window

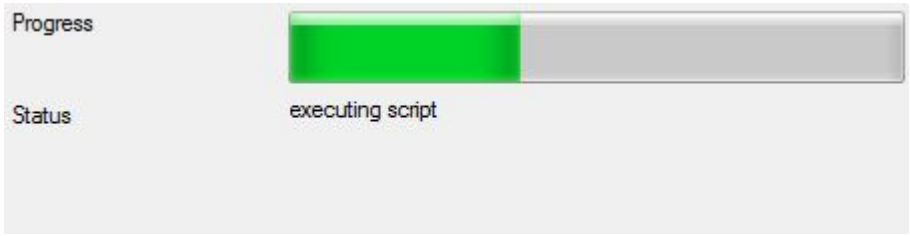
## 4.4.2. Label to print

This section shows a preview of the label content.


## 4.4.3. Progress and Status information

This section shows the **Progress** of programming and the **Status**.

If an error occurs, it is displayed in the **Status** section.




NOTICE

Information about errors can be found in the table Error messages, p. 17.

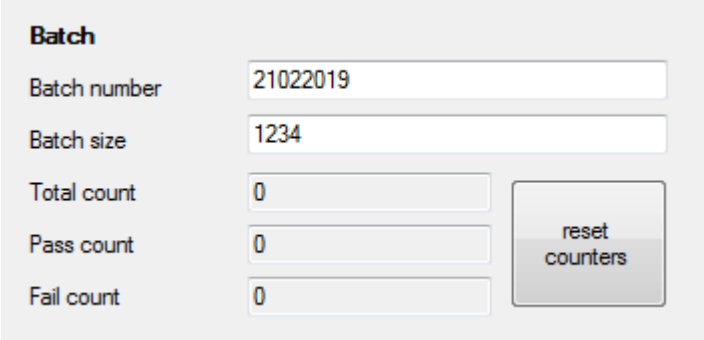
## 4.4.4. Operator ID

This section shows the Name of operator or production desc. The Operator ID can not be locked.

## 4.4.5. Batch

Within this settings, it is possible to type in a batch number and a batch size. It is also possible to reset the **Pass count**, **Fail count** and **Total count**.
The batch number and batch size and **reset counters** can not be locked.

# WORK window

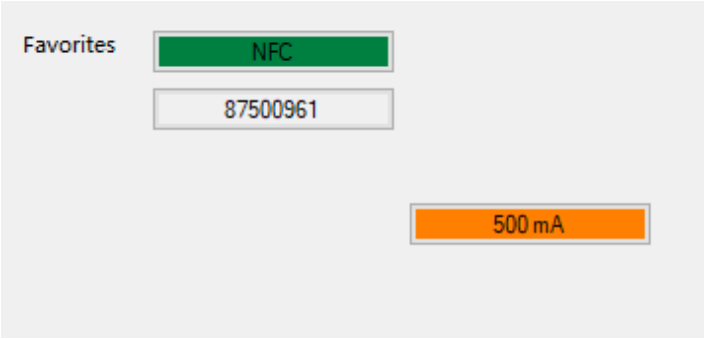



## 4.4.6. Identified Devices

The identified devices window is **only** visible if NFC is used! This counter shows if devices are connected and, if so, how many.


## 4.4.7. Favorites

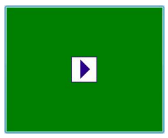
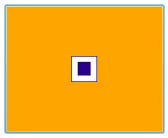
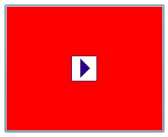
Allows to easily change between previously set up favorite files by pressing on the respective button.



# WORK window

## 4.4.8. Execute button

With the **Execute** button, the programming is started. The button also provides information about the programming status. This information is also available if programming has been started automatically or via barcode.

| Colour | Description |
|---|---|
|  | Blue after starting the program |
|  | Green after successful programming and ready to program the next driver |
|  | Orange during programming |
|  | Red after unsuccessful programming and ready to program the next driver |

There are several ways to start the programming of an LED Driver

_ By pressing the **Execute** button directly in the **Work** window

_ Automatically, by scanning a barcode

_ Automatically, by connecting an LED Driver

# WORK window

## 4.5. Reprint Label

If label printing is activated, labels are automatically printed after each successful programming cycle.

By pressing the **reprint Label** button, it is possible to reprint the last printed label.

## 4.6. Error messages

The following list shows possible error messages and explains their meaning and possible solutions.

Unspecific

| Error Message | Description | Solution |
|---|---|---|
| No valid interface selected for this script type | No interface for the script type selected | |
| No valid printer selected for label printing | No valid printer available | |
| Error comparing data, returned value is different from written value | This error message appears when a byte is not correctly written | |
| Identification of connected driver does not match driver definition in script | This error message appears when a driver is connected that is different from the one defined in the script | |
| Error switching output off/on / Error resetting output | This error occurs when at the moment of switch on a mains problem occurs (e.g. voltage dips, interruptions or burst/surge) | Check for distortions on main wires |
| Error switching output off/on, short circuit active / Error resetting output, short circuit active | Not possible to switch on because of short circuit | |
| Error switching output off/on, no mains connected / Error resetting output, no mains connected | Not possible to switch on because no mains connected | |
| Error switching DALI supply on/off / Error resetting DALI supply | Not possible to switch on DALI PS | Check USB connection to the ready2mains Programmer |
| Error switching DALI supply on/off, short circuit active | Not possible to switch on DALI PS because of short circuit on DALI | |
| Error switching DALI supply on/off, no mains connected / Error resetting DALI supply, no mains connected | Not possible to switch on DALI PS because no mains connected | |
| Error on execute DALI addressing / Error on execute DALI addressing, no DALI supply / Error on execute DALI addressing, short circuit on DALI bus | Addressing on DALI bus failed | Check DALI installation |

# WORK window

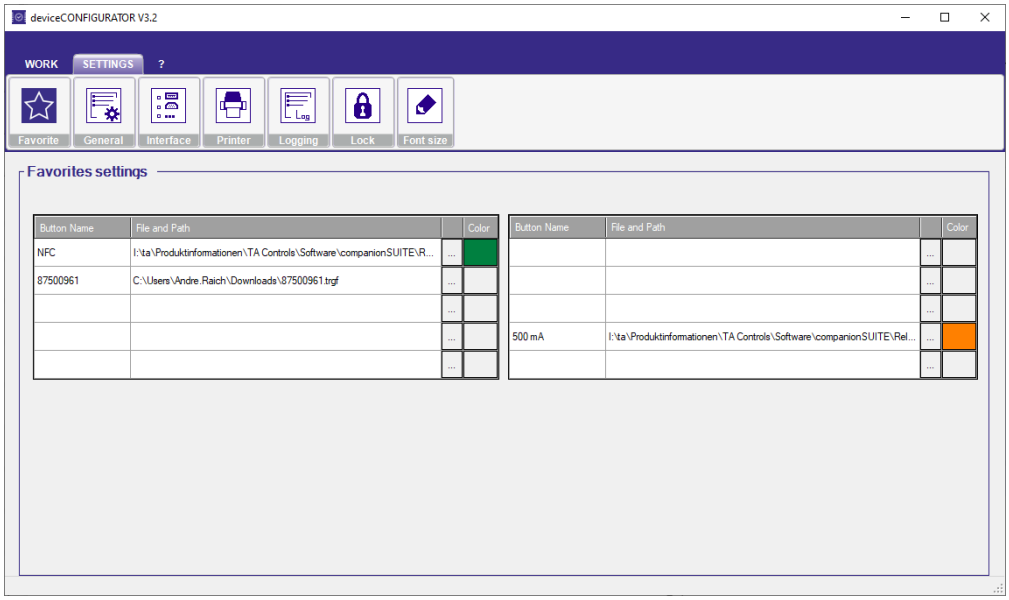
| | | |
|---|---|---|
| Error on execute DALI addressing, invalid address parameter given | Address given in script is out of DALI address range, or maybe the script is defect | Check script |
| Unknown communication error. DLL returned... | Communication to dali.dll is okay, problems to communicate with the bus | Check DALI installation |
| Error sending data through ready3mains Programmer device | Not possible to send data through ready2mains Programmer | Check USB connection to the ready2mains Programmer |
| DLL could not be loaded / BusServer not loaded | DALI dll. not started | Check if DALI bus server is installed |
| Error on writing/reading data: returned .... | DALI data verify problem because written value and read out value are different. The error message displays which value was read out. | Check wiring to bus |

ready2mains

| Error Message | Description | Solution |
|---|---|---|
| Error transmitting a ready2mains command /dimming frame | This error message appears when a frame is not sent.<br>This can happen if disturbance occurs at the start of the ready2mains frame mains (no correct zero cross detection possible) | Check for distortions on main wires |
| Error transmitting a ready2mains command /dimming frame, output is off | This error message appears when a dimming frame is sent but the output of the ready2mains Programmer is off | Switch the output of the programmer on via script or by pressing ON/OFF button on programmer keypad |
| Error transmitting a ready2mains command /dimming frame, output short circuit | This error message appears when a dimming frame is sent but there is a shortcut at the output | Check output wiring |
| Error transmitting a ready2mains command /dimming frame, no mains available | This error message appears when a dimming frame is sent but there is no mains available | Connect AC mains |

# Settings

## 5.1. Lock




## 5.2. Favorites settings

The **Favorite settings** enable you to adapt which scripts are loaded with the favorite buttons in the work window and how the buttons should look like.

### Option: Button Name

Allows to change the Name of the button displayed in the work window.

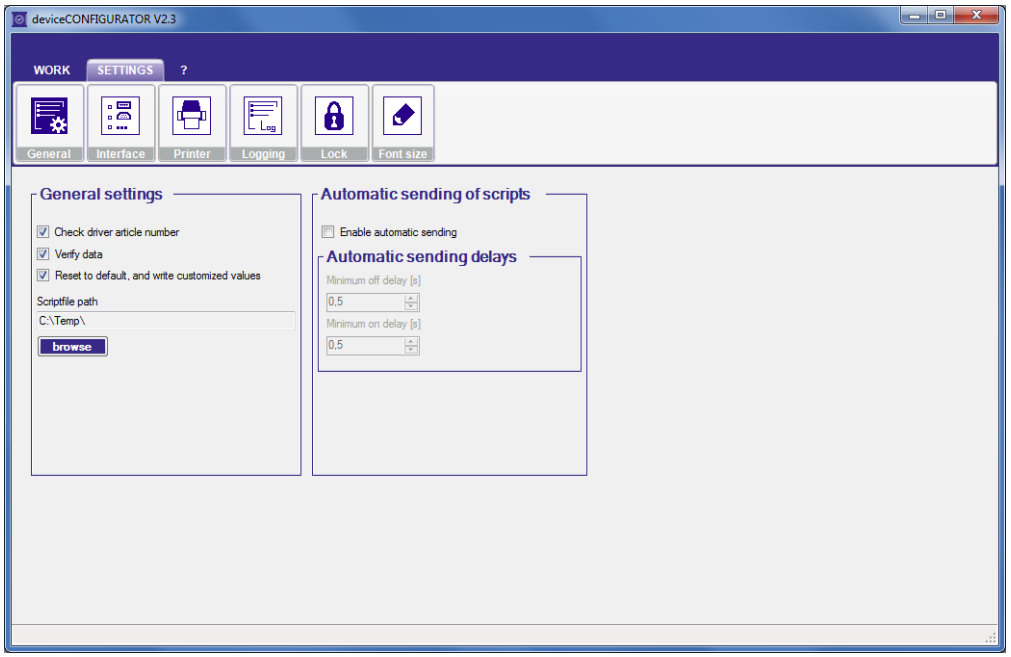
### Option: File and Path

Shows the file which is loaded when this button is pressed. The file can be changed by pressing on the button with the three dots:

### Option: Color

Select the color of the button.

## 5.3. General




## 5.4. General settings

### Option: Check driver article number

With this option it is possible to check that the correct driver is connected, before programming.

NOTICE

**NFC:** Check is possible with single and multi programming.
**DALI:** Check is only possible with single programming.
**ready2mains:** Check is not possible.

### Option: Verify data

With this option each parameter that is written to the driver is automatically checked for correctness.

NOTICE

**NFC:** Verify is possible with single and multi programming.
**DALI:** Verify is only possible with single programming.
**ready2mains:** Verify is not possible.

### Option: Reset to default, and write customized values

With this option all parameters are written. For values without change the default value is set! If this option is not set only the changed parameters are written (faster)

⚠ CAUTION!

Visibility depends on the selected interface type and deviceGENERATOR version of the file:

**NFC:** Not visible, settings changed into the configuration file into a own function
**DALI:** Not visible, settings changed into the configuration file into a own function
**ready2mains: Visible, because it is not possible to TODO !!**

ⓘ NOTICE

_ This option is necessary if already programmed drivers are used for a second time! In this case, the old parameters stored in the driver will be overwritten.

_ Only those parameter which are listed under show functions in the deviceGENERATOR can be set to reset value.

_ Via ready2mains it is not possible to set all possible functions to reset value.

### Scriptfile path

This path defines where the parameter files are stored. If a barcode scanner is used, the deviceCONFIGURATOR loads that file whose name is encoded in the barcode.

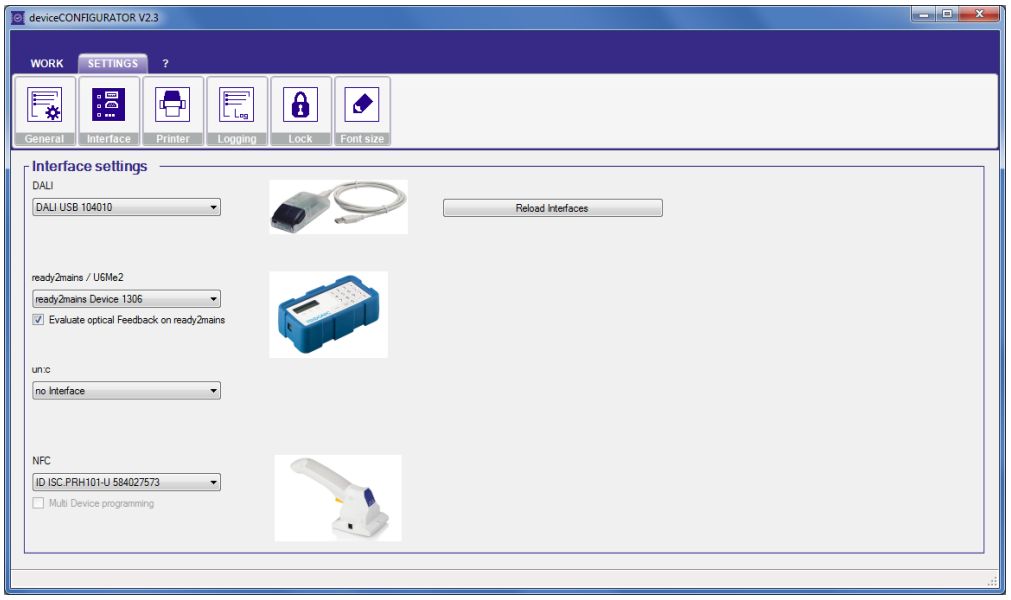
## 5.5. Automatic sending of scripts

If the checkbox **Enable automatic sending** is selected, the deviceCONFIGURATOR checks if there is an LED Driver connected to the DALI line or NFC antenna.
If an LED Driver is connected, the deviceCONFIGURATOR automatically starts the programming of the driver (which means that the script is sent to the driver).

Additionally, it is possible to set an **Minimum off delay** and an **Minimum on delay** time.
These parameters define how long a driver has to be connected or disconnected before the programming is automatically started.

# Settings

## 5.6. Interface




## 5.7. Interface settings

Within the Interface settings the used interface for the different protocols is configured.

The following interfaces are available to configure Tridonic products:

_ NFC Antennas from FAIG

_ DALI USB (Art. No. 24138923)
In the DALI USB there is no additional DALI Power supply built-in. This means, an additional DALI PS is necessary (e.g. DALI PS1 24034323)

_ ready2mains Programmer (Art. Nr. 28001206)
With the ready2mains Programmer, it is possible to configure Tridonic products via DALI, ready2mains and U6Me2. The advantage of the ready2mains Programmer is that the internal 20mA DALI power supply is available.

NOTICE

The un:c programmer was removed from deviceCONFIGURATOR with the release of version 2.6.

If the Interface is connected to the USB slot after the deviceCONFIGURATOR is started, it will be displayed after pressing the button **Reload Interfaces.**

# Settings

## DALI

For DALI there are two interfaces available:

_ ready2mains Programmer

_ DALI USB

If a ready2mains Programmer is in use, there are additional options available


With the ready2mains Programmer it is possible to switch the power output and the DALI PS on or off.
Both options can be activated in the interface settings:

_ Activate the checkboxes for **Automatic on/off for Programmer power output** and/or **Automatic on/off for Programmer DALI PS output**.

If **Automatic on/off for Programmer power output** is active, the output of the ready2mains Programmer is switched automatically on and off before and after sending a script.
If **Automatic on/off for Programmer DALI PS output** is active, the DALI PS of the ready2mains Programmer is switched automatically on and off before and after sending a script.

⚠ CAUTION!

In DALI scripts, only DALI commands can be used. Other commands that do not belong to the DALI standard are ignored.

## ready2mains / U6Me2

For ready2mains or U6Me2 it is necessary to use the ready2mains Programmer. Automatic on and off switching of the output is not available! This must be implemented in the script.

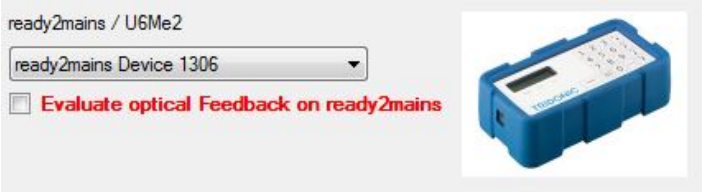
**Evaluate optical feedback on ready2mains**

If the driver is programmed via ready2mains, the load shows an optical feedback. If the programming was successful, the connected load will flash twice.

Additionally, the deviceCONFIGURATOR software can also evaluate the programming.

_ Activate the checkbox **Evaluate optical Feedback on ready2mains**.
-> If the programming was not successful the deviceCONFIGURATOR software will show a red error message.

# Settings




⚠ CAUTION!

Activating the checkbox **Evaluate optical Feedback on ready2mains** gives added safety.

But there are cases where it can be more useful to deactivate this option:

It is possible that the ready2mains programmer does not recognize the optical feedback of the load correctly which leads to unnecessary error messages in the deviceCONFIGURATOR.

If you decide to deactivate the checkbox **Evaluate optical Feedback on ready2mains** remember the following:

_ The deviceCONFIGURATOR software does not evaluate the programming anymore.

_ Make sure that you watch the connected load and control its optical feedback.

## NFC

To use NFC, it is necessary to use one of the antennas listed in the deviceCONFIGURATOR compatible antennas document at https://www.tridonic.com/com/de/download/technical/NFC_readers_companionSUITE.pdf


Activate the checkbock **Multi Device programming** if more than one driver should be programmed at once.

⚠ CAUTION!

Consider that not all antennas are suitable for multi device programming!

_ If **Multiple Device programming** is greyed out, this antenna is not able to program more than one device!

# Settings

⚠ CAUTION!

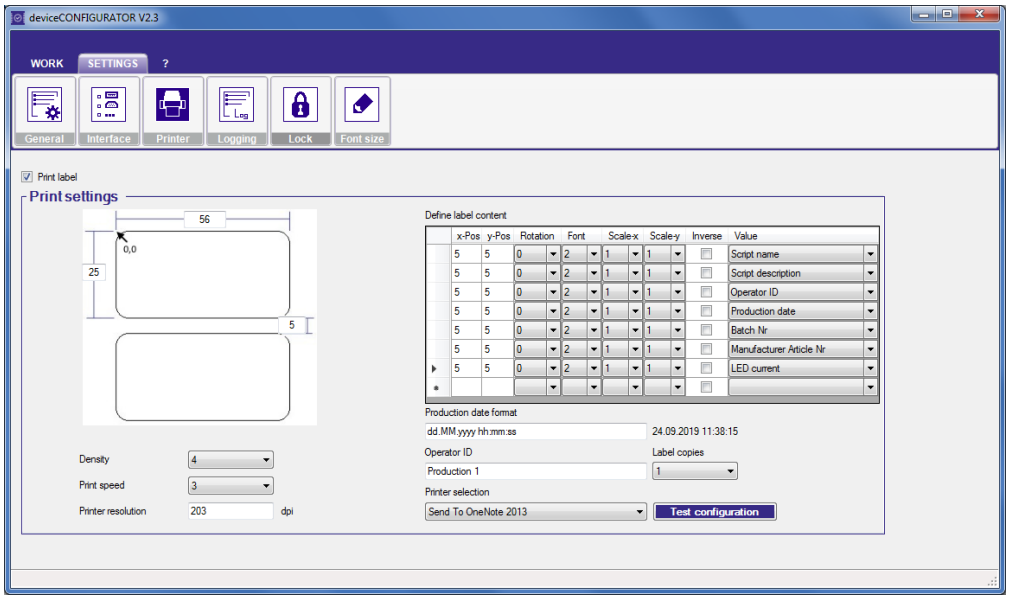
When installing a current driver, the old driver is automatically uninstalled.

_ Parallel use of old and new NFC devices is not possible!

Possible workaround: Old applications should also work with the current driver. Under WIN 10 it is necessary to use the current FEUSB.DLL. The current FEUSB.DLL can be found in the driver package or the current SDK or in the ISOStart / CPRStart directory.

# Settings

## 5.8. Printer




### Printer requirement

**CAUTION!**

The deviceCONFIGURATOR uses the EPL 2 printing protocol.
The used printer so must support the EPL 2 protocol.
There are several printers available which support this protocol. Tridonic made tests with Intermec PC43d and PC23d which are compatible.
To avoid compatibility problems between driver and printer, in some cases it will be necessary to install the driver before the printer is connected.
Follow the installation description from the printer manual !

## 5.9. Printer settings

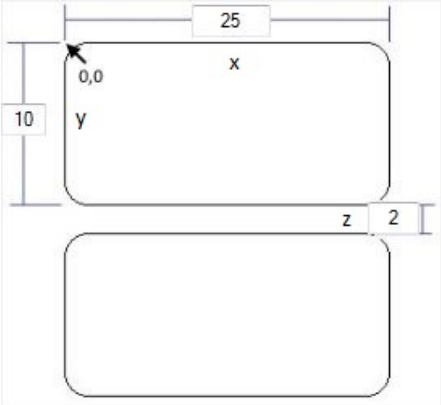
### Label

Label printing is active if print label is activated in the **Print settings**. After successfully programming an LED Driver, a label will be printed automatically.
If label printing is activated, a preview of the label will be visible in the **Work** window. The following image shows an example for how the label looks like.

# Settings

| | |
|---|---|
| Label to print | Max Mustermann<br>500mA cS5.daliProd |

### Label size




| Parameter | Description |
|---|---|
| x | Label width |
| y | Label height |
| z | Width between Labels |
| 0,0 | Label origin |

### Label content

With the label content, it is possible to define what will be printed on the label. Each row of the table corresponds to a row on the label.

# Settings


**CAUTION!**

_ It is necessary to set all the parameters of a row. If not all the parameters are set, the printer will not print this row! Also have a look at the EPL 2 protocol definition which describes the different parameters in more detail.

_ Rows can be deleted with the delete key from the computer keyboard.

_ Using the button **Test configuration**, it is possible to print a sample label.

Available content to write on the label

| Value | Description |
|---|---|
| Script name | Complete name of the script file |
| Script description | File description<br>(defined within the deviceGENERATOR or scriptGENERATOR) |
| Operator ID | Name or acronym of operator |
| Production Date | Production date in the format as defined in the field **Date Format** |
| Batch No. | Batch number as defined in **General** settings / Batch number |
| Serial No. | The Serial number is the **Total count number** |
| Manufacturer Article No. | Manufacturer Article Number<br>(defined within the deviceGENERATOR or scriptGENERATOR) |
| Driver Batch-Serial Number | Batch and Serial Number read out from the driver |
| LED Current | LED current of the driver |

# Settings

| | |
|---|---|
| EAN128 Batch No. | Batch number encoded in EAN128 barcode |
| EAN128 Serial No. | Serial number encoded in EAN128 barcode |
| EAN128 Batch-Serial | Batch and Serial number encoded in EAN128 barcode |
| EAN128 Driver Batch-Serial No. | Driver Batch-Serial Number encoded in EAN128 barcode |

### Row Settings

| Parameter | Description |
|---|---|
| x-Pos | Defines x position on the label, calculated from the origin |
| y-Pos | Defines y position on the label, calculated from the origin |
| Rotation | Defines the rotation of the text. Available options are 0°, 90°, 180° and 270° |
| Font | Defines the font described in the EPL 2 protocol |
| Scale-x | Defines the Scale-x as described in the EPL 2 protocol |
| Scale-y | Defines the Scale-y as described in the EPL 2 protocol |
| Inverse | Inverses black and white |
| Value | Defines content of the row |

### Printing Barcodes

The size of the barcodes can be changed by adapting the Scale-x and Scale-y.

### Date format

The date format uses the following placeholders:

_ d -> day

_ M -> month

_ y -> year

_ h -> hours

_ m -> minutes

_ s -> seconds

Both two-digit or four-digit formats can be used. Custom delimiters can be used to separate the placeholders.

**Example:** Using a date format in the form of "dd.MM.yyyy hh:mm:ss" will output a date like "01.01.2017 15:22:12".

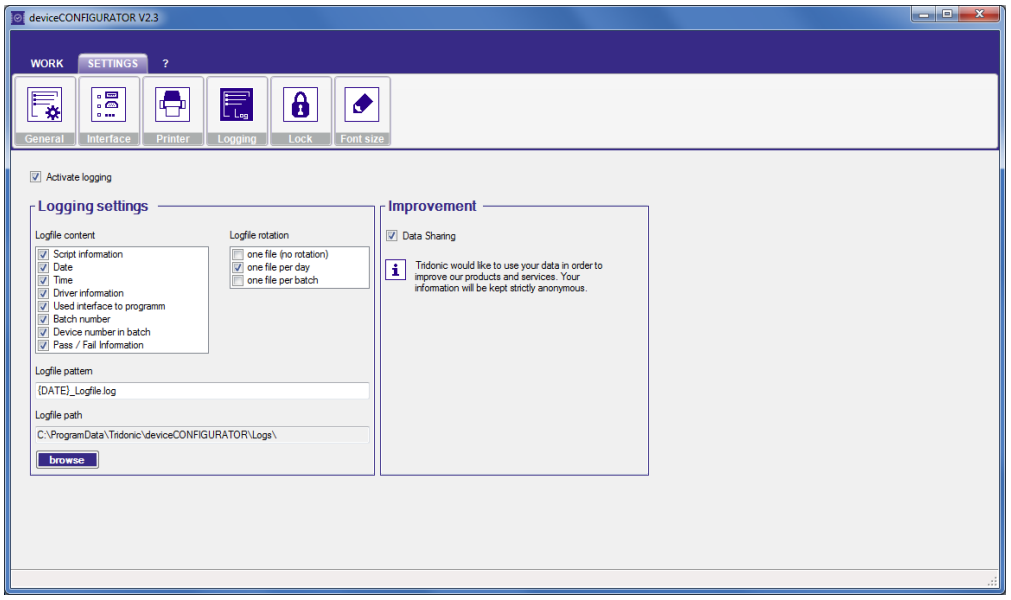
# Settings

General printer settings

| Parameter | Description |
|---|---|
| Density | Defines the density of the printed text |
| Print speed | Defines the speed of the label printer |
| Print resolution | Defines the printer resolution in dpi |
| Label copies | Defines the amount of labels to print at each successful programming cycle |
| Printer selection | Selects the used label printer |

# Settings

## 5.10. Logging




With the check box **Activate logging** it is possible to activate production logging with the selected logging settings.

NOTICE

Error logging is always active and cannot be disabled.

## 5.11. Logging settings

### 5.11.1. Log file content

#### Option: Script information

| Name of log file information | Where does this information come from | Possible values |
|---|---|---|
| LED Current from file; | LED current value out of file | mA (min - max) |
| Manufacturer ArtNo. from file; | Function:<br>Luminaire information - Luminaire article number | Example:<br>Lamp 1 |
| Label text from file; | Function:<br>Luminaire information - Luminaire description | Example:<br>Downlight 1000 lx |
| deviceGENERATOR version | deviceGENERATOR version at file creation | 2.3.0.508 |
| Filename | Out of the loaded file | <name>.trgf |

# Settings

| | | |
|---|---|---|
| Script GUID | Unic ID of script - for identification | code |
| Script downloaded | Date and time at file creation | date + time |
| Script dataset count | Number of datasets = functions | 0 - 254 |

Option: Date

| Name of log file information | Where does this information come from | Possible values |
|---|---|---|
| Time stamp (including Date) | Current production date | 2019.01.10 |

Option: Time

| Name of log file information | Where does this information come from | Possible values |
|---|---|---|
| Time stamp (including Time) | Current production time | 14:36:53 |

Option: Driver information

| Name of log file information | Where does this information come from | Possible values |
|---|---|---|
| Device description from file; | Device name out of file | Example: LCA 25W 350-1050mA one4all SR PRE |
| Device ArtNo. from file; | Device Article number out of file | Example: 28000671 |
| Device GTIN from file; | Device GTIN number out of file | Example: 9006210523721 |
| Read device GTIN; | GTIN number read from connected device | Example: 9006210523712 |
| Read device Batch #; | Batch number read from connected device | Example: 20252247 |
| Read device Serial #; | Serial number read from connected device | Example: 4400 |
| Read device Firmwareversion; | Firmware version number read from connected device | Example: V4.7.131 |
| Read device Productiondate; | Production date read from connected device | Example: 24.09.2019 |

Option: Used interface to program

| Name of log file information | Where does this information come from | Possible values |
|---|---|---|
| Interface name | Name of the used interface type (**Interface setting site**) | Example: DALI USB 104010 |

# Settings

Option: Batch number

| Name of log file information | Where does this information come from | Possible values |
|---|---|---|
| Batch #; | Number of input box **Batch number** (**General setting site**) | 0 - 2.147.483.647 |

Option: Device number in batch

| Name of log file information | Where does this information come from | Possible values |
|---|---|---|
| Batch count; | Continuous number of pass devices (**Work site**) | 1 - max |

Option: Pass / Fail Information

| Name of log file information | Where does this information come from | Possible values |
|---|---|---|
| Pass/Fail | Programming status | pass, fail |

No Option selected - this information are always located in the logging file

| Name of log file information | Where does this information come from | Possible values |
|---|---|---|
| Operator ID from settings; | Information out of text filed Operator ID (**Printer setting site**) | 30 character |
| Number of connected drivers; | When NFC is the programming interface, the number of connected drivers is stored here. | 1 to max |
| Write all values including defaults; | Status of Option: Reset to default, and write customized values (**General setting site**) | True / False |
| Verify enabled; | Status of Option: Verify data (**General setting site**) | True / False |
| Check driver article Number; | Status of Option: Check driver article number (**General setting site**) | True / False |
| Automatic mode enabled; | Status of Option: Enable automatic sending (**General setting site**) | True / False |
| Automatic sending on barcode scan enabled; | Fixed true - not visible. | True |
| Automatic sending minimum off time [ms]; | Status of Option: Minimum off delay [s] (**General setting site**) | 500 to 999900 ms |
| Automatic sending minimum on time [ms]; | Status of Option: Minimum on delay [s] (**General setting site**) | 500 to 999900 ms |

# Settings

| | | |
|---|---|---|
| Automatic enable of power output on Programmer; | Status of Option: Automatic on/off for Programmer output ( **General setting site**) | True / False |
| Automatic enable of DALI power output on programmer; | Status of Option: Automatic on/off for Programmer DALI PS ( **Interface setting site**) | True / False |
| Enable feedback evaluation on ready2mains Programming; | Status of Option: Evaluate optical Feedback on ready2mains ( **Interface setting site**) | True / False |
| Multi programming on NFC enabled; | Status of Option: Multi Device programming (**Interface setting site**) | True / False |
| AutoProgramming mode active; | This device where programmed in automatic sending mode. | True / False |
| Print label; | Status of Option: Print label (**Printer setting site**) | True / False |
| Application locked; | Status of Option: Lock/Open (**Work site**) | True / False |
| Has lock password; | Password available (**Lock setting site**) | True / False |
| Allow script change; | Status of Option: Lock file change (**Lock setting site**) | True / False |
| Warn script change; | Status of Option: Warn on file change (**Lock setting site**) | True / False |
| Configuration file path; | Path defined in the **General setting site** | C:\... |
| Logfile path; | Path defined in the **Logging setting site** | C:\... |
| Log message; | Text information about the programming status | Programming successful |

### Log file rotation

This defines the number of possible log file options:

_ one file (no rotation): Complete logging is done in one file

_ one file per day: Every day a new log file is created

_ one file per batch: With every batch a new log file is created

### Log file pattern

With the log file pattern, it is possible to define the log file name by using wildcards. Clicking on the input field shows the list of possible wildcards.

### Log file path

Path where log files are saved. Use the key "**browse**" to change the log file path.

## 5.12. Improvement

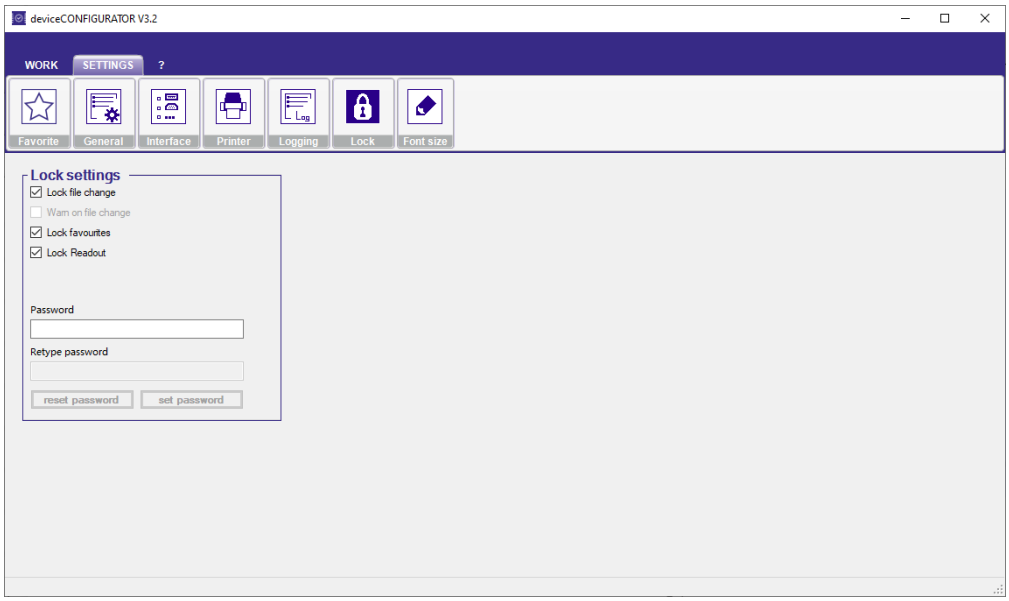
With the check box **Data Sharing** the user can allow the use of tracking in order to help improve the quality of the software.

NOTICE

If the **Data Sharing** is activated, an image of the driver will be visible in the **Work** window. This makes it easy to recognize it the option is active or not.

# Settings

## 5.13. Lock




## 5.14. Lock settings

The **Lock settings** enable you to adapt script loading settings and to set the lock password.

### Option: Lock file change

If **Lock file change** is selected, it is not possible to change a script via barcode or open a file if the **Work** window is locked. If this option is not selected, it is possible to change the script via barcode or open a file even if the **Work** window is locked!

### Option: Warn on file change

If **Warn on file change** is selected, a message will warn the user about script changes!

### Option: Lock favorites

If **Lock favorite** is selected, it is not possible to change a script via the favorite button if the **Work** window is locked. If this option is not selected, it is possible to change the script via the favorite buttons even if the **Work** window is locked!

### Option: Lock Readout

If **Lock Readout** is selected, it is not possible to read out a driver if the **Work** window is locked

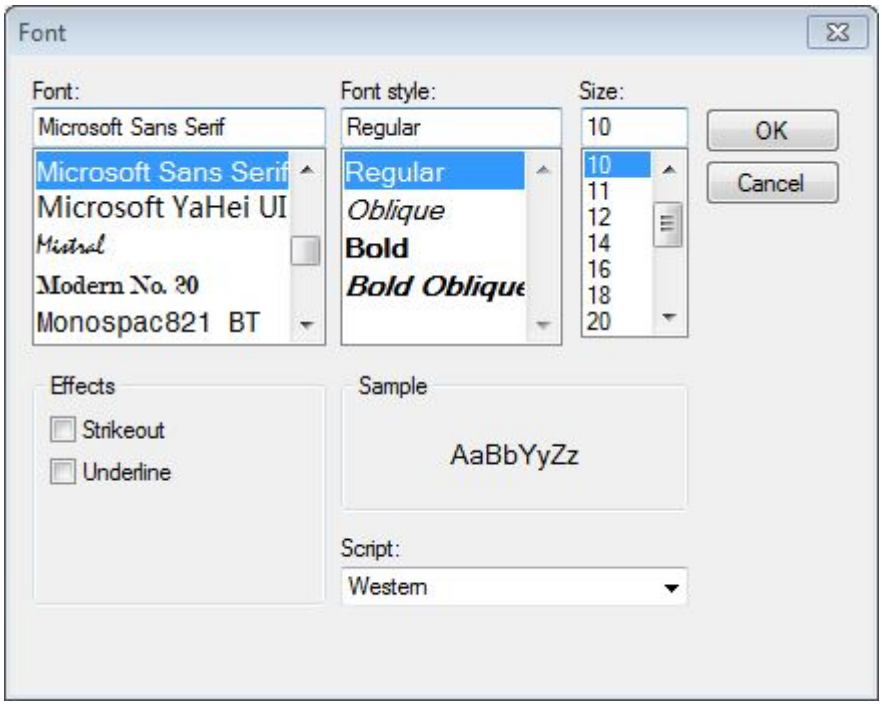
### Password

With the input field **Password,** it is possible to set or reset the password.

## 5.15. Font settings

The **Font** settings enable you to adapt font family, style and size.

To change **Font** settings, make a selection and press **OK.**



## 5.16. Barcode settings

It is possible to load the configuration file with a barcode scanner.
The script name is directly encoded in the barcode. The deviceCONFIGURATOR loads the script corresponding to the barcode information. It is possible to change this folder in **General settings**.

The following barcode type is used:

_ EAN128 with max. 48 characters

⚠ CAUTION!

The maximum number of characters for the filename (including extension) is 48! Filenames with more than 48 characters cannot be linked.

## 5.17. Barcode configuration

Before a barcode scanner can be used, it has to be configured.

Make sure that the barcode scanner is configured as follows:

_ start character - STX
_ end character - EOT
_ Function key emulation (Windows mode control) activated
_ The keyboard layout of the scanner has to be adjusted to the standard keyboard layout used on the computer to identify the correct character set.

⚠ CAUTION!

The barcode scanner must be configured with the supplied software for the scanner!

## 5.18. Predefined Standard Folders

The deviceCONFIGURATOR uses the following folders as standard folders. The **Standard Path** can be changed in the settings.

| File type | Standard Path |
|---|---|
| Log files | C:\ProgramData\Tridonic\deviceCONFIGURATOR\Logs |
| Settings files | C:\ProgramData\Tridonic\deviceCONFIGURATOR |
| Default script path | C:\ProgramData\Tridonic\deviceCONFIGURATOR\Scripts |

## 6.1. Programming TRIDONIC drivers via DALI and companionSUITE in production

In this chapter the standard procedure to program TRIDONIC LED Drivers via DALI and companionSUITE is described.

**CAUTION!**

_ If a TRIDONIC LED Driver supports NFC or DALI interface, it is strongly recommended to use NFC or DALI as the interface and not ready2mains.

**NOTICE**

_ DALI is faster than ready2mains

_ DALI is a two way communication which allows to verify the written parameters and the connected driver

_ With DALI, it is not possible to program more than one driver in one programming cycle (framing errors because of two answering drivers)

_ 230V mains power is necessary to program TRIDONIC LED Drivers via DALI

# Programming modes and examples


### 6.1.1. Hardware and wiring to program via DALI

The available hardware to program a TRIDONIC LED Driver via DALI is the DALI USB interface or the ready2mains programmer. Additionally to the DALI USB, a computer with the latest version of the companionSUITE is required.

# Programming modes and examples

**DALI USB interface**

The DALI USB interface provides only a DALI USB interface without DALI power supply. An additional power supply is necessary.

⚠ CAUTION!

_ Additional to the DALI USB interface a DALI power supply is necessary !

_ It's recommended to build up frame where you can put in the driver during programming to avoid electrical shock to the operator and switch on and off mains automatically.

**ready2mains Programmer**

The ready2mains programmer provides a DALI USB interface with an integrated power supply. There is no need for an additional DALI power supply if the ready2mains programmer is used.

# Programming modes and examples

**⚠ CAUTION!**

_ To use the ready2mains programmer in combination with the companionSUITE the firmware version must be 1.1.45

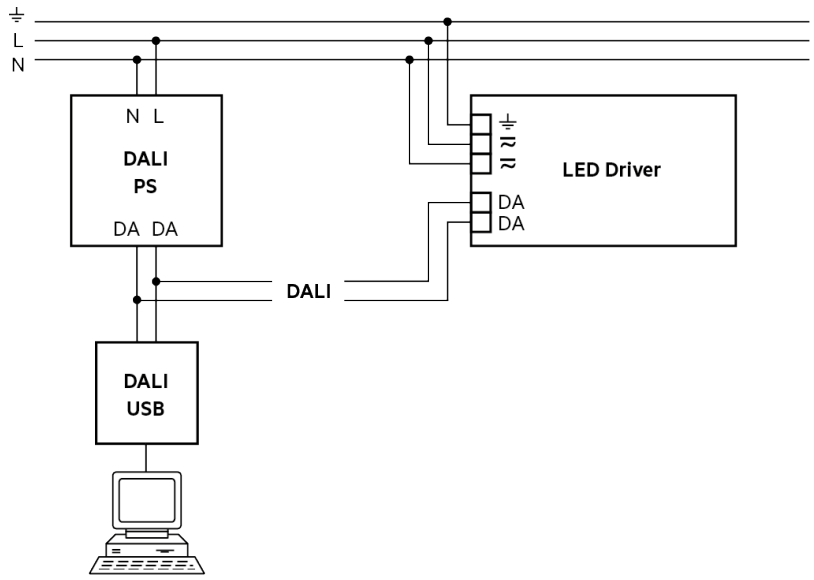
_ It is possible to power the drives via the ready2mains programmer. To avoid damage to the programmer, two things have to considered: It must be ensured that no ground fault occurs at the output (isolation transformer necessary) and HV tests have to be done while the ready2mains programmer is disconnected. It is recommended to use mains directly switched over a contactor.

_ Fully automatic programming is only possible with an external mains power! Fully automated programming with the power output of the ready2mains programmer will lead to wrong short circuit detection caused by hotplugging the power output! Therefore only manual programming is available when the power output is in use!

_ It is recommended to build up a frame where you can put in the driver during programming to avoid electrical shock to the operator and switch the mains on and off automatically.

## Wiring with DALI USB as DALI interface

When using the DALI USB as the interface to program Tridonic LED Drivers, it is necessary to build up a DALI line with a DALI power supply and the DALI USB interface.




## Wiring with ready2mains programmer as DALI interface

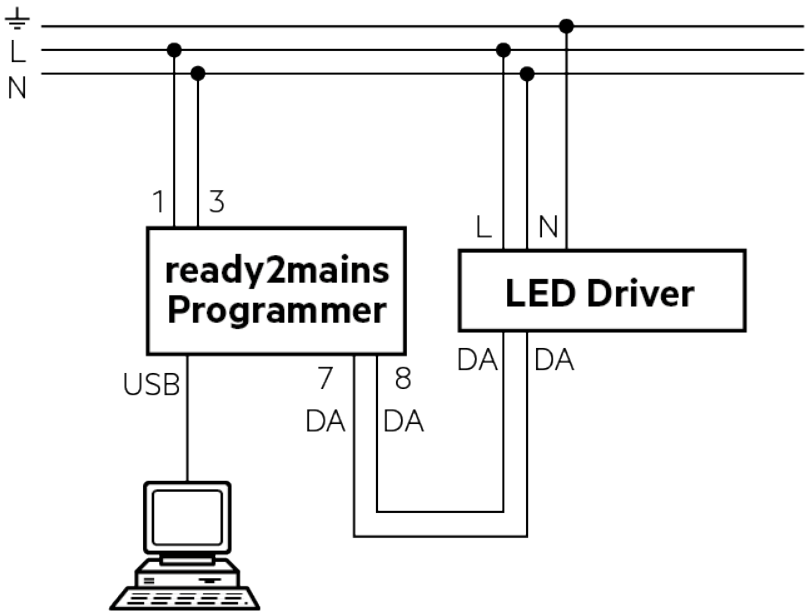
If the ready2mains programmer is used as the DALI interface, it is recommended to use only the DALI interface of the ready2mains programmer. If the ready2mains programmer should also supply the mains power, ensure that the ready2mains programmer is secured against ground fault with an isolation transformer and that the ready2mains programmer is not connected when HV tests are executed.

# Programming modes and examples

## Wiring with DALI USB interface of Programmer

CAUTION!

_ It is recommended to build up frame where you can put in the driver during programming to avoid electrical shock to the operator and switch on and off mains automatically.




## Wiring with DALI USB and mains power supply of Programmer

CAUTION!

_ **This chapter has been inserted for the sake of completeness.**

_ This chapter describes how to use the power output of the ready2mains programmer.

_ Be careful if the output is used because wrong usage can lead to damaged programmers in the production or to troubles during changing the driver connected to the ready2mains programmer

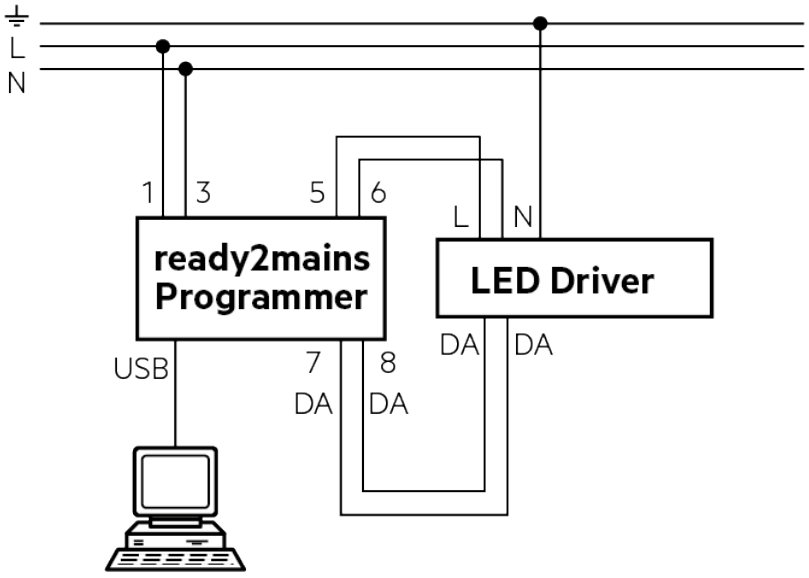
_ To prevent the ready2mains programmer from damage, use an isolation transformer (only if power output is used)

_ For HV tests the ready2mains programmer must be disconnected (only if power output is used)

_ Due to short circuit detection while hotpluging the power output, no automatic detection of Tridonic LED Drivers is possible! Only manual start stop mode is available.

_ It is recommended to build up frame where you can put in the driver during programming to avoid electrical shock to the operator and switch on and off mains automatically.

# Programming modes and examples




CAUTION!

_ To prevent ready2mains programmer from damage use an isolation transformer.

_ For HV tests ready2mains programmer must be disconnected

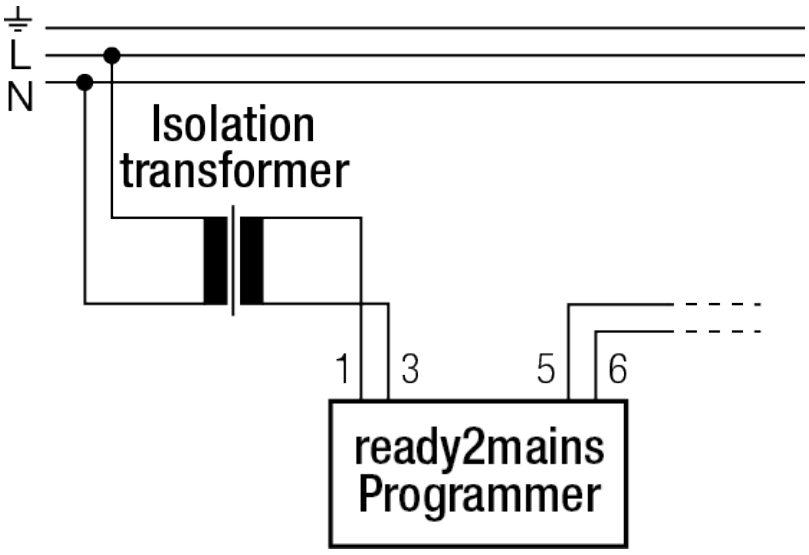
_ It is recommended to build up frame where you can put in the driver during programming to avoid electrical shock to the operator and switch on and off mains automatically

**Isolation Transformer**

To prevent damage to the ready2mains programmer, it is recommended to use an isolation transformer between mains and ready2mains programmer. This is necessary because the ready2mains programmer is not able to recognize ground faults. If a groundfault occurs, it is possible that the output stage of the ready2mains programmer gets destroyed!

Connect the isolation transformer in this way:

# Programming modes and examples




**Tested transformator**

RS Pro 500VA Isolating Transformer
Reference number at RS components: 504-228

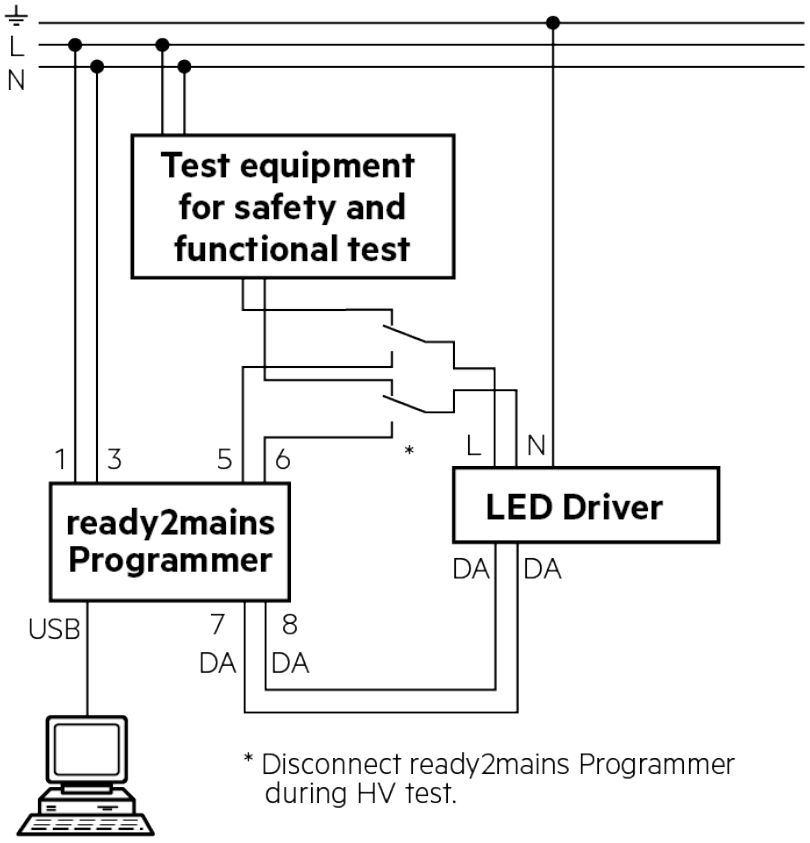
⚠ CAUTION!

_ Isolation transformer should be protected against secondary short-circuit (B10 circuit breaker primary or a fuse secondary).

**Additionally used test equipment**

To avoid wrong measurements and destruction of the ready2mains programmer, we strongly recommend to disconnect the ready2mains programming during additional testing procedures like high voltage testing.

# Programming modes and examples




**CAUTION!**

_ If a ready2mains programmer is connected during HV testing, it is possible that the ready2mains programmer is tested and not the luminaire itself!

## 6.1.2. Setup of deviceCONFIGURATOR

To use the companionSUITE with a DALI USB interface or with ready2mains programmer set up the deviceCONFIGURATOR as described below.
This document only describes how to set the deviceCONFIGURATOR up with verifying data and driver check.
It is not recommended to switch these functions off. While switching off this functions, it is not guaranteed that all parameters are written or that a correct driver is connected !

**Parameters to set up in deviceCONFIGURATOR which affect DALI programming**

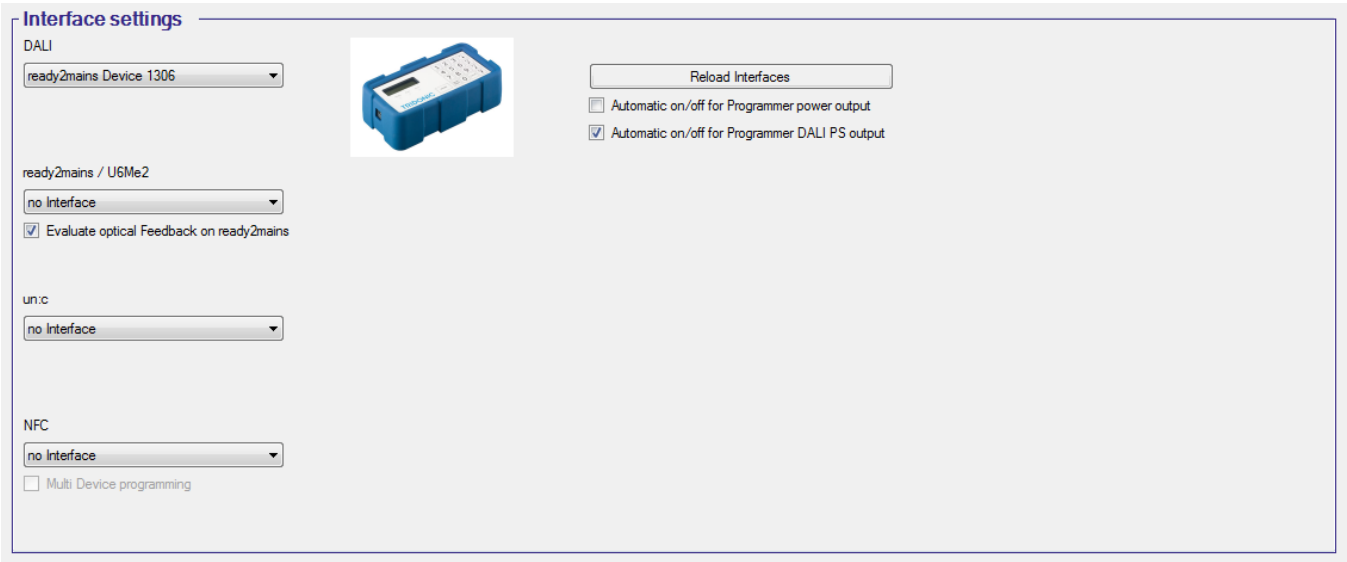
_ Interface settings

_ General settings / Reset to default and write customized values

**Interface settings**

Depending on the available DALI interface choose the DALI USB or the ready2mains Programmer.
If the DALI USB interface is used, choose the DALI USB in the Interface settings for DALI.
No additional interface configurations are necessary.


If the ready2mains programmer is used, choose the ready2mains programmer in the interface settings for DALI. Additional to the interface itself, it is necessary to set the DALI PS and power output settings.



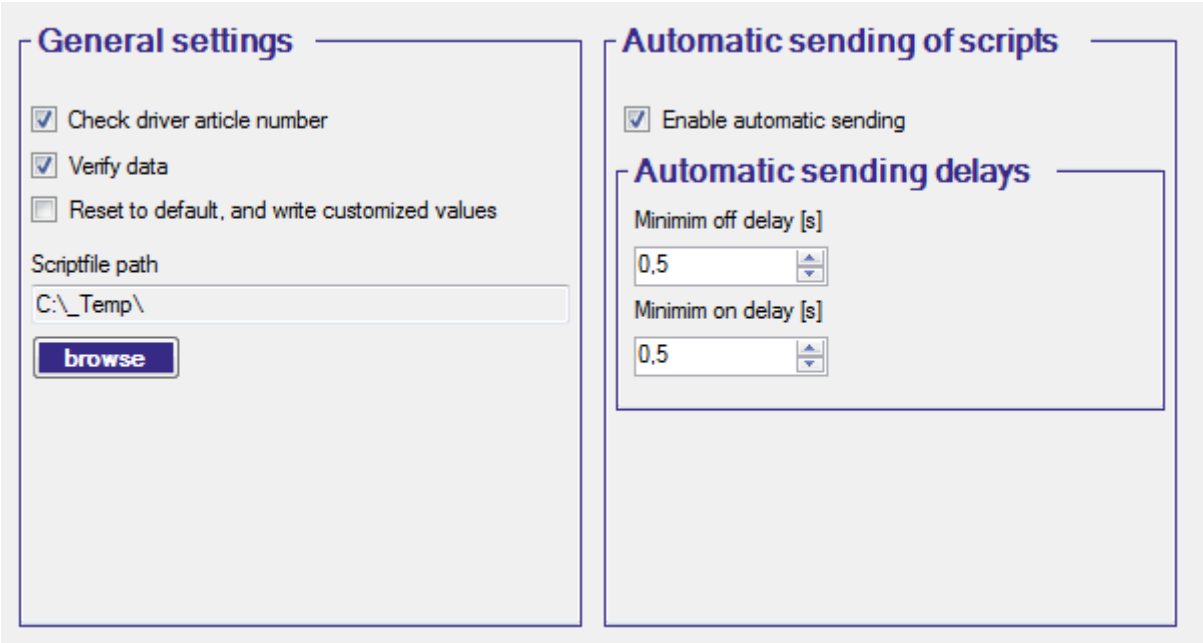
# Programming modes and examples

With automatic on/off settings for the power output and/or the DALI PS, it is possible to control this with the deviceCONFIGURATOR.
If automatic on/off setting is activated, the power output and/or the DALI PS is switched on automatically before each programming cycle!
To use the integrated DALI PS of the ready2mains programmer it is essential that automatic on/off for programmer DALI PS output is activated.
Only use the **automatic on/off Programmer power output** if the power output is used in combination with manual programming of each device.
For more information see workflow manual programming via DALI.
In the fully automated mode where the deviceCONFIGURATOR detects if a driver is connected to the line DO NOT USE, the power output of the ready2mains programmer!
Connecting and disconnecting loads from the output until the output is on can lead to spikes which the ready2mains programmer detects as short circuit and switches off.
Hot plugging doesn't lead to damage of the ready2mains programmer but to security shut downs!

⚠ CAUTION!

_ It is recommended not to use the power output of the ready2mains programmer. Use extra mains power switched by a contactor !

**General Settings:**




In the general settings for DALI programming, all values are affected during programming.

# Programming modes and examples

_ When **checking the driver article number** it is essential that the deviceCONFIGURATOR checks if the attached driver is the correct driver related to the configuration file.

_ **Verify data** is essential to 100% guarantee that the programmed parameters are correct! The deviceCONFIGURATOR reads each parameter back from the driver and compares the value with the content of the configuration file.

_ **Reset to default and write customized values** is necessary if the driver is not brand new from stock. With this function it is possible to reset all parameters to default except the ones which are set in the script! This slows down the programming process! Use this for drivers which are not brand new from stock! It is recommended to use this function. Only switch this function off if it is guaranteed that only brand new drivers are programmed!

_ **Batch Settings** affect the batch overview in the work window. The batch settings have no direct influence on the programmed values, but only on the values visible in the batch area of the work window and the content of the log files

_ **Enable automatic sending** is recommended while programming DALI drivers. If this is activated, the deviceCONFIGURATOR constantly checks the DALI line for attached drivers. If the operator attaches a driver to the DALI line, the deviceCONFIGURATOR will recognize this and automatically start the programming process. It is recommended to use automatic sending. This reduces the interaction of the operator with the deviceCONFIGURATOR and is faster than a manual start. Only disable this option if a manual process is desired.

Options for automatic programming:

_ **Check driver article number** -> activated

_ **Verify data** -> activated

_ **Enable automatic sending** -> activated

Options for manual programming:

_ **Check driver article number** -> activated

_ **Verify data** -> activated

_ **Enable automatic sending** -> deactivated

⚠ CAUTION!

_ Always use **check driver article number** and **verify data** to ensure correct programming of the driver

_ Use **reset to default and write customized values** if the driver is NOT brand new!

_ It is recommended to use **Enable automatic sending.** Without this option each programming cycle must be started manually.

# Programming modes and examples

NOTICE

From deviceGENERATOR >= 2.3 it is possible to set the functionality "**Reset to default** in the configuration file.

In this case, the function in the deviceCONFIGURATOR will be hidden.

### 6.1.3. Programming procedure in production

This procedure describes the workflow to program Tridonic LED Drivers via DALI in a production line.

⚠ CAUTION!

Before this procedure will work, it is essential that:

_ Setup and wiring is built up correctly

_ Interfaces are connected to the computer

_ deviceCONFIGURATOR software is installed and running

_ deviceCONFIGURATOR settings are correct (interfaces, general settings...)

_ Access to the configuration files is available

In the schematics below, the principle workflows to program LED Drivers via DALI are illustrated. In the table below the principle workflow paths are described:

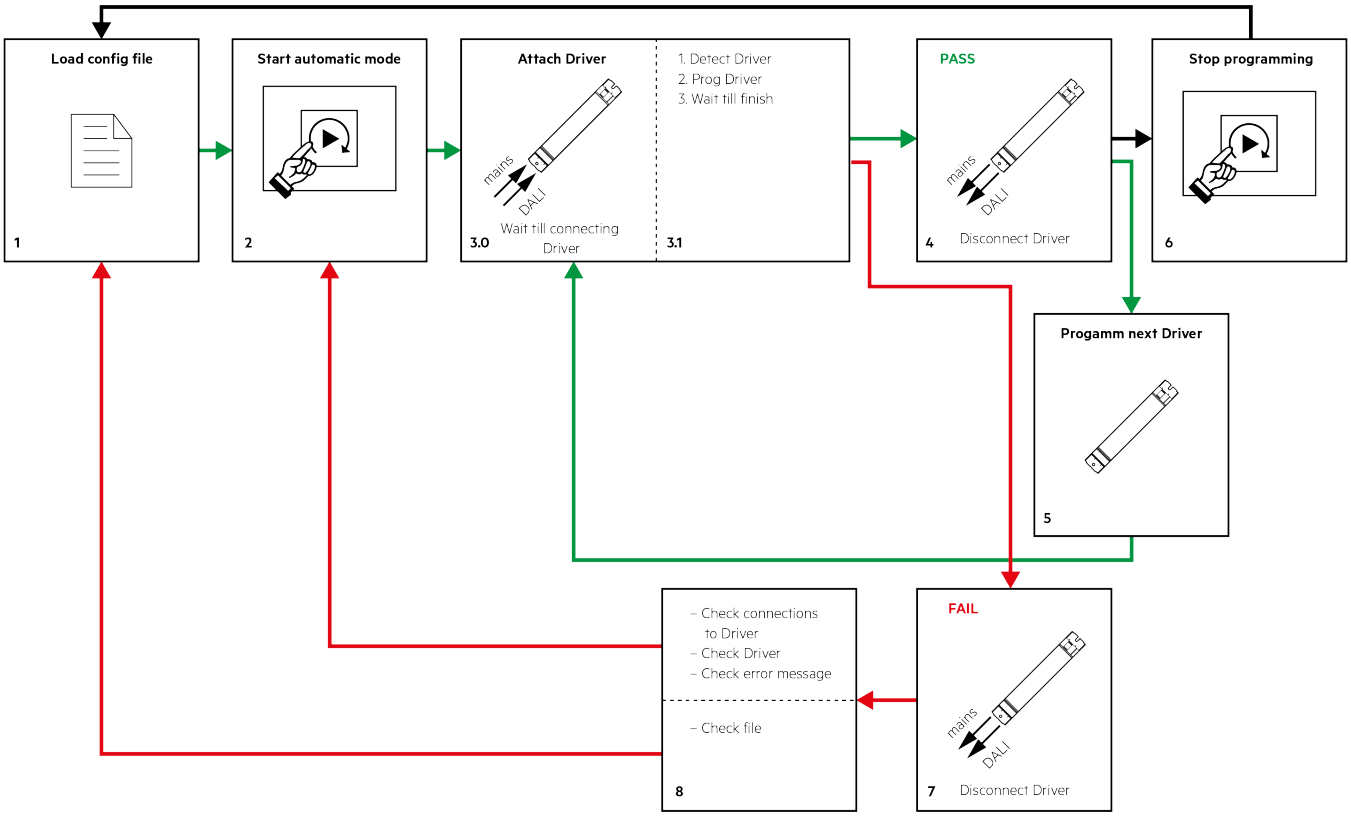
| Colour | Description |
|---|---|
| **GREEN** | The green path in the schematic shows the workflow from loading the file and programming the drivers in case everything works correctly |
| **BLACK** | The black path in the schematic is the stop path and describes what to do to stop the old batch and start a new batch |
| **RED** | The red path in the schematic is shown if the workflow and programming of a driver fails. |

⚠ CAUTION!

_ This schematic does not describe faults which could happen caused by a wrong installation of the system!

# Programming modes and examples

## 6.1.4. Schematic of the automatic programming workflow (recommended)




**Green path:**

| No. | Description |
|---|---|
| 1 | The first step to start a batch is to load a configuration file. This can be done manually by choosing a file from the file system or automated with a barcode scanner. For more information how to integrate a barcode scanner see Barcode configuration, p. 38. |
| 2 | After the correct configuration file is loaded, start the automatic programming mode by pressing the start button in the work window. |
| 3.0 | Attach the driver to the DALI bus and mains. Now the system automatically detects if a driver is attached and starts the programming of the driver.<br>For safety and speed reasons, it is recommended to build up a closed box with fix connections to plug the driver fast to DALI and mains. |

# Programming modes and examples

| | |
|---|---|
| 3.1 | Wait until the programming is finished.<br>_ If programming is done correctly, proceed with the green path and step number 4.<br>_ If programming failed, proceed with the red path and step number 7. |
| 4 | The programming is finished correctly if the start button turns to green.<br>_ When printing is activated, this is the moment when the label is printed.<br>Consider: A label is only printed if the programming was correct!<br><br>Disconnect the driver from the DALI line and mains.<br>_ To program the next driver, proceed with the green path and step number 5<br>_ To stop programming, proceed with the black path and step number 6 |
| 5 | To program the next driver, proceed with the green path and step number 2, repeat this procedure for each driver |

**Black path:**

| No. | Description |
|---|---|
| 6 | To stop the batch, press the start button again and proceed with the black path and step number 1. |

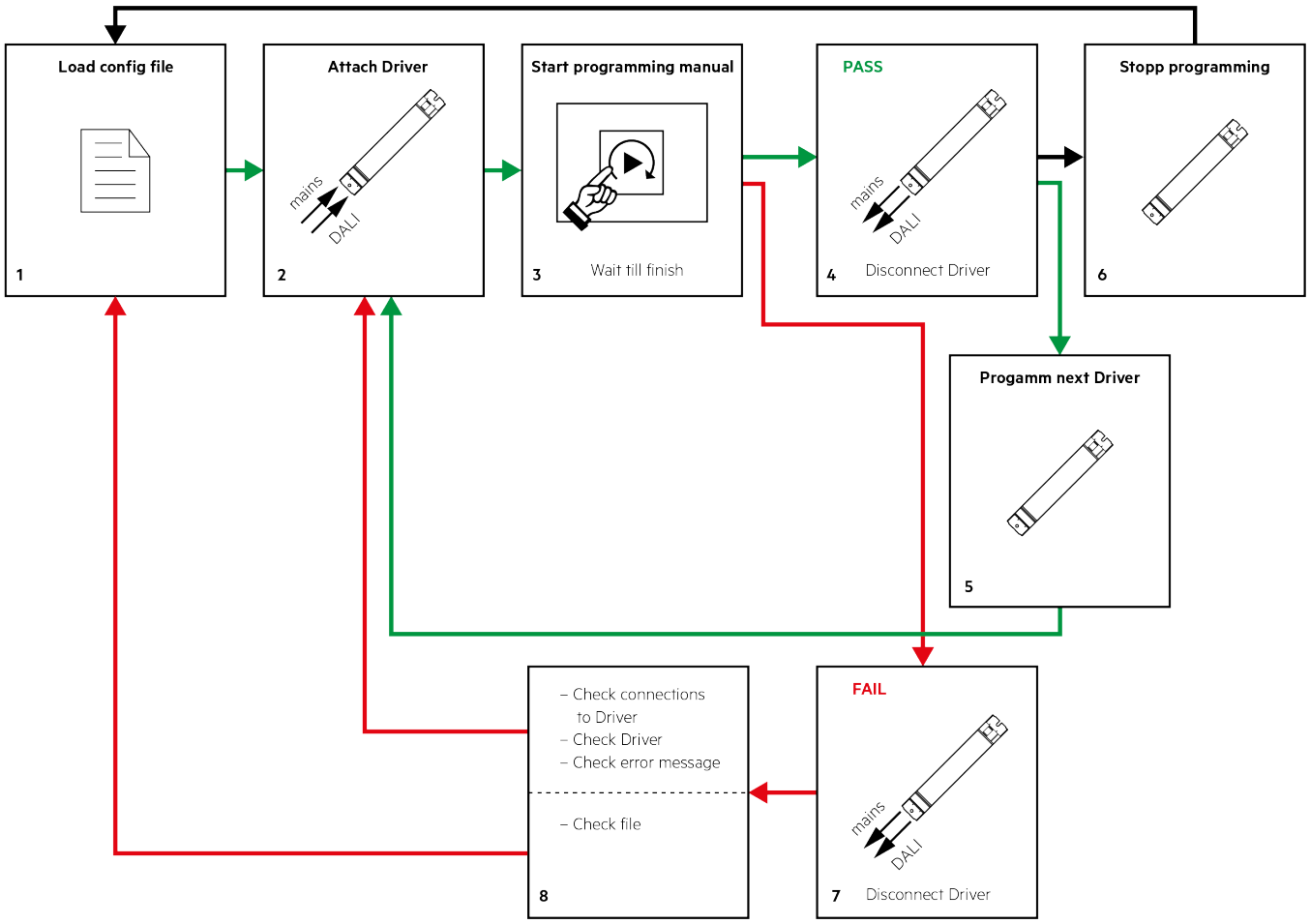
**Red path:**

| No. | Description |
|---|---|
| 7 | If the programming is finished with a fail, the start button turns to red!<br>_ Note: If printing is activated, no label is printed in this case!<br><br>Disconnect the driver from the DALI line and mains. Proceed with the red path and step number 8 |

# Programming modes and examples

| | |
|---|---|
| 8 | Before you proceed with a second try, check the error message in the deviceCONFIGURATOR window and do the following:<br>_ Check the mains and DALI line connections to the driver and from the driver.<br>_ Check if the correct driver is used<br>If the connections are bad or the wrong driver is connected, proceed with the red path and step number 2.<br>If both is correct:<br>_ Check the configuration file if the correct one is loaded<br>If the wrong configuration file is loaded, proceed with the red path and step number 1. |

## 6.1.5. Schematic of the manual programming workflow



# Programming modes and examples

Green path:

| No. | Description |
|---|---|
| 1 | The first step to start a batch is to load a configuration file. This can be done by choosing a file from the file system or automated with a barcode scanner.<br>For more information how to integrate a barcode scanner see Barcode configuration, p. 38. |
| 2 | After the correct configuration file is loaded, attach the driver to the DALI bus and mains. For safety and speed reasons, it is recommended to build up a closed box with fix connections to plug the driver fast to DALI and mains. |
| 3.0 | When the driver is connected, start the programming of the driver manually by pressing the start button on the screen. |
| 3.1 | Wait until the programming is finished.<br>_ If programming is done correctly, proceed with the green path and step number 4.<br>_ If programming failed, proceed with the red path and step number 7 |
| 4 | The programming is finished correctly if the start button turns green.<br>_ When printing is activated, this is the moment when the label is printed. Consider: A label is only printed when the programming was correct!<br>Disconnect the driver from the DALI line and mains.<br>_ To program the next driver, proceed with the green path and step number 5<br>_ To stop programming, proceed with the black path and step number 6 |
| 5 | To program the next driver, proceed with the green path and step number 2, repeat this procedure for each driver |

Black path:

| No. | Description |
|---|---|
| 6 | To stop the batch and start a new batch, proceed with the black path and step number 1. |

Red path:

| No. | Description |
|---|---|
| 7 | If programming is finished with a fail, the start button turns to red!<br>_ Note: When printing is activated, no label is printed in this case!<br>Disconnect the driver from the DALI line and mains. Proceed with the red path and step number 8 |

| | |
|---|---|
| 8 | Before you proceed with a second try, check the error message in the deviceCONFIGURATOR window and do the following:<br>_ Check the mains and DALI line connections to the driver and from the driver.<br>_ Check if the correct driver is used<br><br>If the connections are bad or the wrong driver is connected, proceed with the red path and step number 2.<br>If both is correct:<br>_ Check the configuration file if the correct one is loaded<br><br>If the wrong configuration file is loaded, proceed with the red path and step number 1. |

## 6.2. Programming TRIDONIC drivers via NFC and companionSUITE in production

In this chapter the standard procedure to program TRIDONIC LED Drivers via NFC and companionSUITE is described.

**CAUTION!**

_ If a TRIDONIC LED Driver supports NFC or DALI interface, it is strongly recommended to use NFC or DALI as the interface and not ready2mains.

**NOTICE**

_ NFC is a method to transfer data contact-free.

_ NFC is faster than DALI or ready2mains.

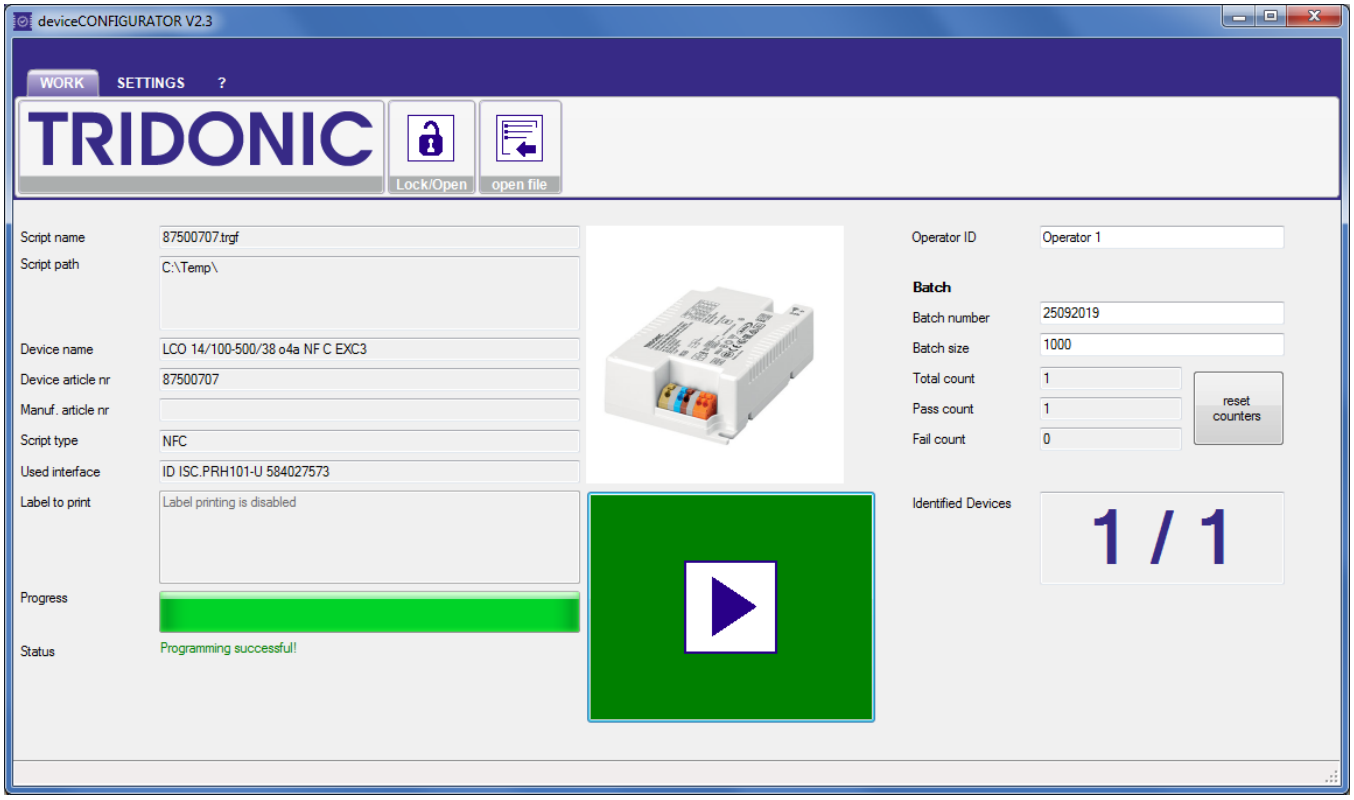
_ NFC is a two way communication which allow to verify the written parameters.

_ It is possible with NFC to check if a TRIDONIC LED Driver is connected or not.

_ Via NFC it is possible to program each single driver separately or program a complete box of drivers.

_ NFC programming works fully automated.

# Programming modes and examples




## 6.2.1. Hardware and wiring to program via NFC

To program via NFC, different antennas are available for single device programming and multi device programming. Every antenna can be used for single programming but not for multi device programming. Tridonic supports NFC antennas from FAIG. The NFC antenna document should help to choose the correct antenna.
Antennas for single and multi device programming compatible with companionSUITE:

https://www.tridonic.com/com/en/download/technical/NFC_readers_companionSUITE.pdf

### Wiring of FAIG NFC antenna as NFC interface

This means there is no need for mains wiring to supply the driver. The antenna powers the NFC tag of the TRIDONIC LED Driver and transfers the data to the driver. The wiring of the NFC interface depends on the used antenna. For the smallest antenna only a USB connection is necessary. For antennas with more power as used for multi device programming, an additional power supply for the antenna is necessary.
For more information about the wiring of the antennas see the original documents of FAIG electronic. For some antennas it is necessary to adjust the antenna before the antenna is usable, this affects especially antennas where the reader is separated from the antenna.

⚠ CAUTION!

_ In some cases it is necessary to adjust the NFC antenna before a proper workflow is guaranteed

_ Positioning of drivers is essential to program drivers. To guarantee a correct connection check where the NFC tag is and position the LED Driver in the correct way. Where the antenna is and how to place them on the antenna is described in the data sheet of the driver

_ If multi programming is used the arrows on the box show in which direction the box should be placed on the antenna

## 6.2.2. Setup of deviceCONFIGURATOR

To use the companionSUITE with FAIG NFC interfaces set up the deviceCONFIGURATOR as described below.
This document only describes how to set up the deviceCONFIGURATOR with verifying data and driver check.
It is not recommended to switch this functions off. When switching off this functions, it is not guaranteed anymore that all parameters are written and that the correct driver is connected !

**Parameters to set up in deviceCONFIGURATOR which affect NFC programming:**

_ Interface settings

_ General settings / reset to default and write customized values

**Interface settings**

Choose the correct NFC antenna as the interface to program via NFC.
No additional interface configurations are necessary.

# Programming modes and examples

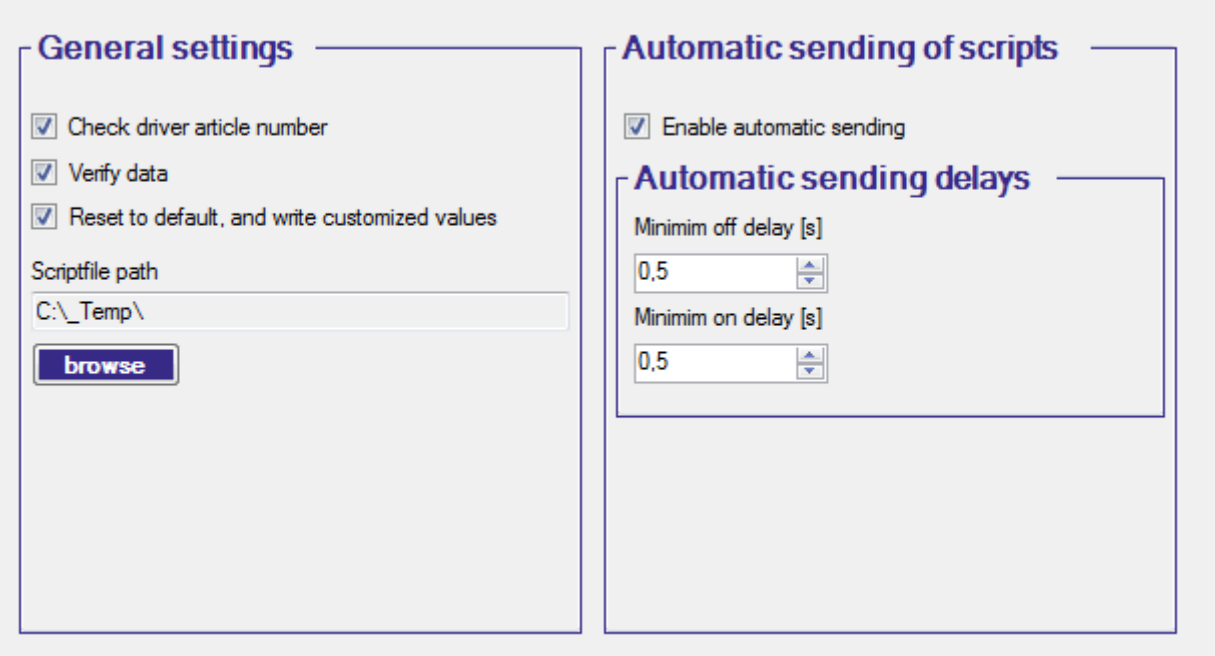
With the **Multi Device programming** option, it is possible to activate and deactivate multi programming.

⚠ CAUTION!

_ Only activate multi programming if more than one driver should be programmed

_ Not each antenna supports multi device programming

**General Settings:**




In the general settings for NFC programming all values have effect during programming.

_ When **checking the driver article number** it is essential that the deviceCONFIGURATOR checks if the attached driver is the correct driver related to the configuration file.

_ **Verify data** is essential to 100% guarantee that the programmed parameters are correct! The deviceCONFIGURATOR reads each parameter back from the driver and compares the value with the content of the configuration file.

_ **Reset to default and write customized values** is necessary if the driver is not brand new from stock. With this function it is possible to reset all parameters to default except the ones which are set in the script! This slows down the programming process! Use this for drivers which are not brand new from stock!

_ **Batch Settings** affect the batch overview in the work window. The batch settings have no direct influence on the programmed values, but only on the values visible in the batch area of the work window and the content of the log files

_ **Enable automatic sending** is recommended while programming DALI drivers. If this is activated, the deviceCONFIGURATOR constantly checks the DALI line for attached drivers. If the operator attaches a driver to the

DALI line, the deviceCONFIGURATOR will recognize this and automatically start the programming process. It is recommended to use automatic sending. This reduces the interaction of the operator with the deviceCONFIGURATOR and is faster than a manual start. Only disable this option if a manual process is desired.

Options for NFC programming:

_ **Check driver article number** -> activated

_ **Verify data** -> activated

_ **Enable automatic sending** -> activated

⚠ CAUTION!

_ Always use **check driver article number** and **verify data** to ensure correct programming of the driver

_ Use **reset to default and, write customized values** if the driver is NOT brand new!

_ It is recommended to use **Enable automatic sending,** without this option each programming cycle must be started manually

ⓘ NOTICE

From deviceGENERATOR >= 2.3 it is possible to set the functionality "Reset to default" in the configuration file.
In this case the function in the deviceCONFIGURATOR will be hidden.

### 6.2.3. Programming procedure in production

This procedure describes the workflow to program Tridonic LED Drivers via NFC in a production line.

⚠ CAUTION!

Before this procedure will work, it is essential that:

_ Setup and wiring is built up correctly

_ Interfaces are connected to the computer

_ deviceCONFIGURATOR software is installed and running

_ deviceCONFIGURATOR settings are correct (interfaces, general settings...)

_ Access to the configuration files is available

_ After programming the device via NFC, the device is powered up once for one second so that the deviceANALYSER can read out the parameters

# Programming modes and examples

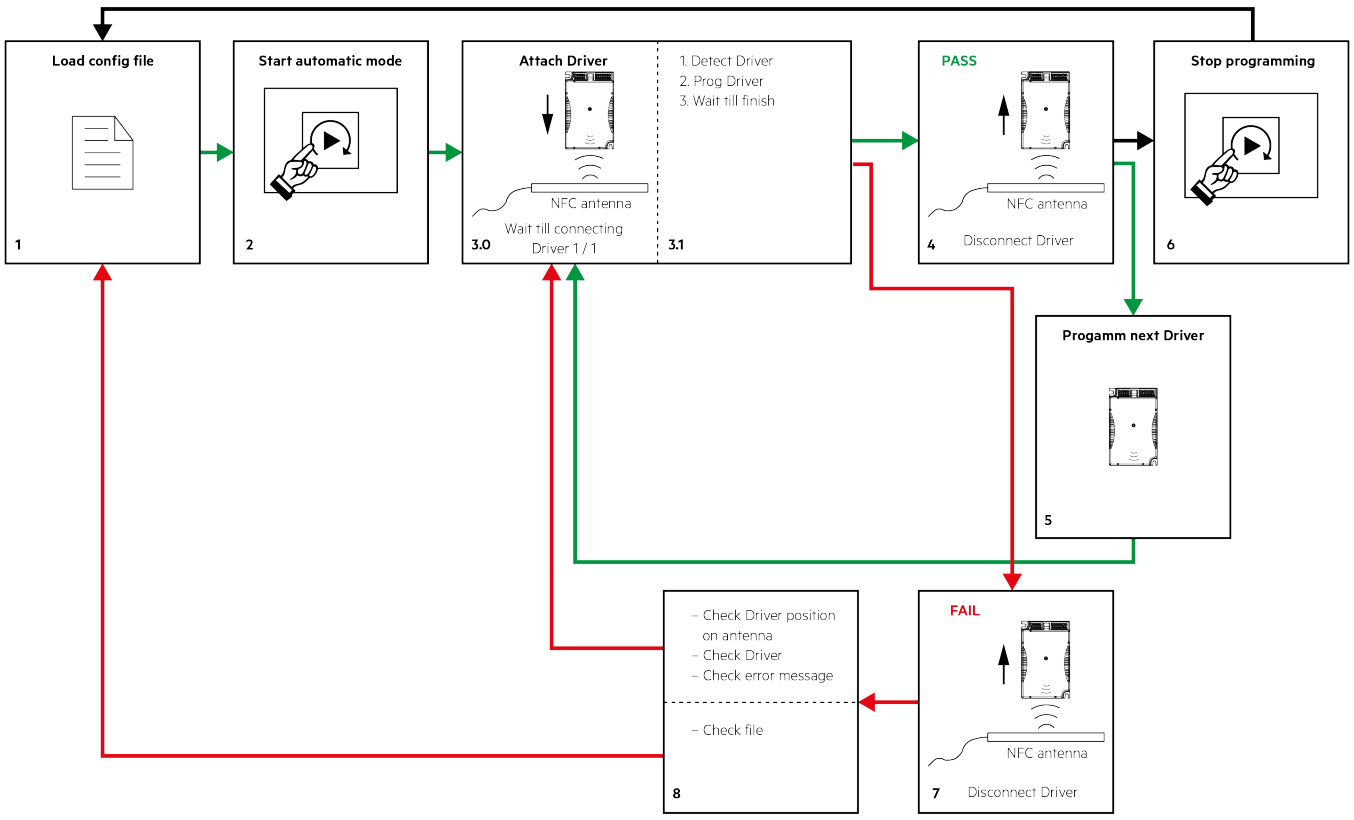
In this schematics below, the principle workflows to program LED Drivers via NFC are illustrated. In the table below the principle workflow paths are described:

| Colour | Description |
|---|---|
| GREEN | The green path in the schematic shows the workflow from loading the file and programming the drivers if everything works correct |
| BLACK | The black path in the schematic is the stop path and describes whats to do to stop the old batch an start a new batch |
| RED | The red path in the schematic is shown if the workflow and the programming of a driver fails. |

⚠ CAUTION!

_ This schematics do not describe faults which could happen caused by a wrong installation of the system!

## Schematic of NFC single device programming workflow




Green path:

| No. | Description |
|---|---|

# Programming modes and examples

| | |
|---|---|
| 1 | First step to start a batch is to load a configuration file. This can be done manually by choosing a file from the file system or automated with a barcode scanner.<br>For more information how to integrate a barcode scanner see Barcode configuration, p. 38. |
| 2 | After the correct configuration file is loaded, start the automatic programming mode by pressing the start button in the work window |
| 3.0 | Put the driver in the correct way to the NFC antenna. Now the system automatically detect if a driver is attached and starts the programming of the driver. If a driver is detected, it is visible in the identified devices area |
| 3.1 | Wait until the programming is finished.<br>_ If programming is done correctly, proceed with the green path and step number 4.<br>_ If programming failed, proceed with the red path and step number 7. |
| 4 | The programming is finished correctly if the start button turns to green.<br>_ When printing is activated, this is the moment when the label is printed.<br>Consider: A label is only printed if the programming was correct!<br><br>Disconnect the driver from the DALI line and mains.<br>_ To program the next driver, proceed with the green path and step number 5<br>_ To stop programming, proceed with the black path and step number 6 |
| 5 | To program the next driver, proceed with the green path and step number 3, repeat this procedure for each driver |

Black path:

| No. | Description |
|---|---|
| 6 | To stop the batch, press the start button again and proceed with the black path and step number 1. |

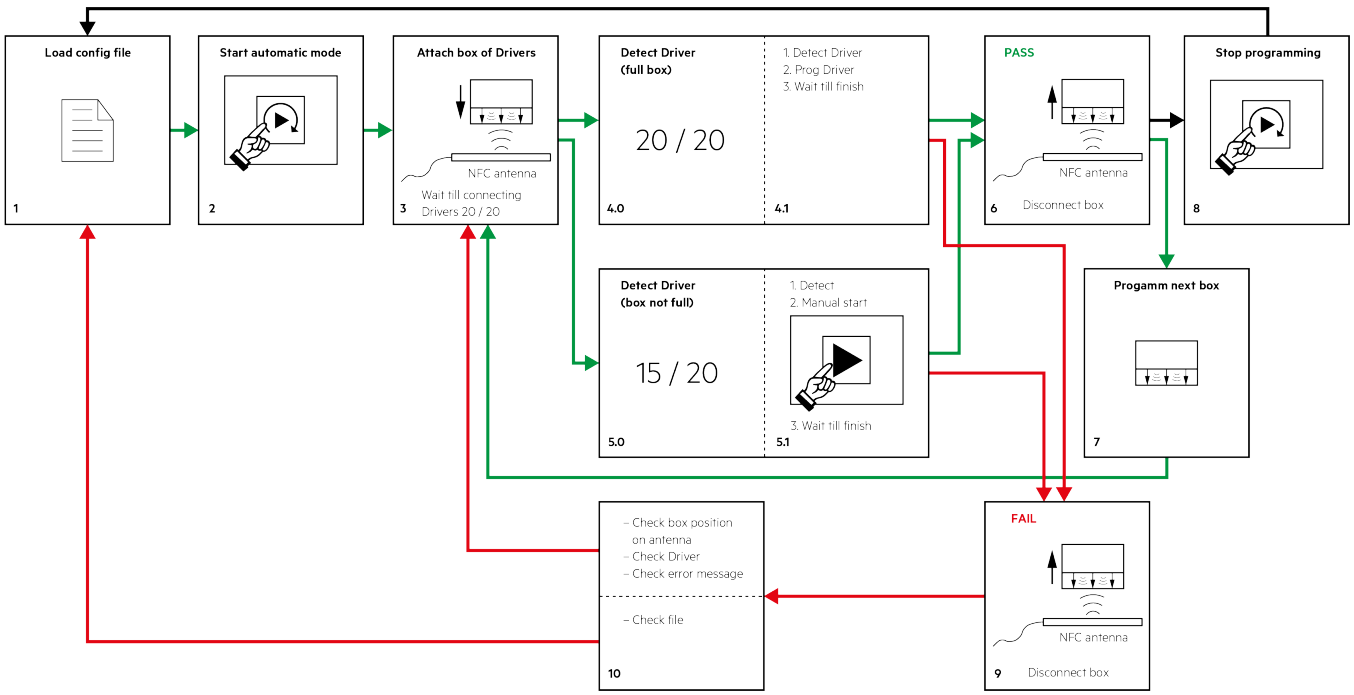
Red path:

| No. | Description |
|---|---|
| 7 | If the programming is finished with a fail, the start button turns to red!<br>_ Note: If printing is activated, no label is printed in this case!<br><br>Disconnect the driver from the DALI line and mains. Proceed with the red path and step number 8 |

# Programming modes and examples

| | |
|---|---|
| 8 | Before you proceed with a second try, check the error message in the deviceCONFIGURATOR window and do the following:<br>_ Check the position of the driver on the antenna.<br>_ Check if the correct driver is used<br>If the driver is positioned bad or the wrong driver is connected, proceed with the red path and step number 3.<br>If both is correct:<br>_ Check the configuration file if the correct one is loaded<br>If the wrong configuration file is loaded, proceed with the red path and step number 1. |

Schematic of NFC multi device programming workflow




Green path:

| No. | Description |
|---|---|
| 1 | The first step to start a batch is to load a configuration file. This can be done by choosing a file from the file system or automated with a barcode scanner. For more information how to integrate a barcode scanner see Barcode configuration, p. 38. |
| 2 | After the correct configuration file is loaded, start the automatic programming mode by pressing the start button in the work window |

# Programming modes and examples

| | |
|---|---|
| **3** | Put the box with driver on the NFC antenna. Now deviceCONFIGURATOR starts to detect the attached devices. Depending on the amount of drivers, deviceCONFIGURATOR starts programming automatically or not! The arrows on the box show how to put the box on the antenna (arrow in direction of the NFC antenna)<br>_ If a full box with drivers is placed on the antenna, proceed with green path and step number 4<br>_ If an opened box where drivers are missing in the box is placed on the antenna, proceed with green path an step number 5<br><br>Example for a box with 20 devices:<br>_ 20/20 devices detected: automatic start of programming<br>_ 1/20 up to 19/20 devices detected: manual start is necessary |
| **4.0** | After deviceCONFIGURATOR detected a full box on the antenna, deviceCONFIGURATOR automatically starts to program this box. The deviceCONFIGURATOR knows how many drivers should be in a full box. The amount of detected drivers is visible in the detected devices area of the work window. If the box is full but only less devices are detected move the box on the antenna until all devices are detected. |
| **4.1** | Wait until the programming is finished.<br>_ If programming is done correctly, proceed with the green path and step number 6.<br>_ If programming failed, proceed with the red path and step number 6 |
| **5.0** | If a open box should be programmed, place this box on the antenna. In the detected devices area of the work window the amount of detected devices is visible. The deviceCONFIGURATOR doesn't start programming automatically if the amount of devices on the antenna deviate from the max possible devices in a box! Move the box on the antenna until all drivers in the box are detected by the deviceCONFIGURATOR. If all devices in the box are detected, start the programming process automatically by pressing the start button. |
| **5.1** | Wait until the programming is finished.<br>_ If programming is done correctly, proceed with the green path and step number 6.<br>_ If programming failed, proceed with the red path and step number 6 |

# Programming modes and examples

| | |
|---|---|
| 6 | The programming is finished correctly if the start button turns to green.<br>_ When printing is activated, this is the moment when the label is printed.<br>Consider: A label is only printed when the programming was correct! The amount of programmed drivers is the amount of printed labels.<br><br>Remove the drivers from the NFC antenna.<br>_ To program the next driver, proceed with the green path and step number 7<br>_ To stop programming, proceed with the black path and step number 8 |
| 7 | To program the next driver, proceed with green path and step number 3, repeat this procedure for each driver |

Black path:

| No. | Description |
|---|---|
| 8 | To stop the batch, press the start button again and proceed with the black path and step number 1. |

Red path:

| No. | Description |
|---|---|
| 9 | If the programming is finished with a fail, the start button turns to red!<br>_ Note: When printing is activated, no label is printed in this case!<br><br>Disconnect the driver from the DALI line and mains. Proceed with the red path and step number 8 |
| 10 | Before you proceed with a second try, check the error message in the deviceCONFIGURATOR window and do the following:<br>_ Check the position of the drivers on the antenna.<br>_ Check if the correct driver is used<br><br>If the driver is positioned wrongly or the wrong driver is connected, proceed with the red path and step number 3.<br>If both is correct:<br>_ Check the configuration file if the correct one is loaded<br><br>If the wrong configuration file is loaded proceed with the red path and step number 1. |

## 6.3. Programming TRIDONIC drivers via ready2mains and companionSUITE in production

In this chapter the standard procedure to program TRIDONIC LED Drivers via ready2mains and companionSUITE is described.

**CAUTION!**

_ If a TRIDONIC LED driver supports NFC or DALI interface, it is strongly recommended to use NFC or DALI as the interface and not ready2mains.

**NOTICE**

_ It is only recommended to use ready2mains for drivers where only a ready2mains interface is available.

_ 230V mains power is necessary to program TRIDONIC LED Drivers via ready2mains

# Programming modes and examples

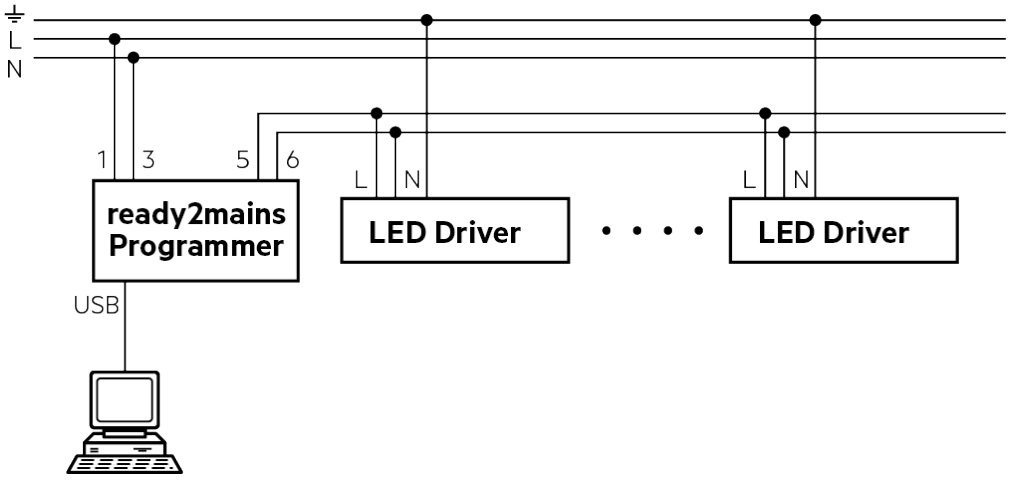
## 6.3.1. Hardware and wiring to program via ready2mains

The available hardware to program a TRIDONIC LED driver via ready2mains is the ready2mains programmer.
Additional to the ready2mains programmer, a computer with the latest version of the companionSUITE is required.


⚠ CAUTION!

_ To use the ready2mains programmer in combination with companionSUITE the firmware version must be 1.1.45.

**General wiring without safety equipment:**



# Programming modes and examples

⚠ CAUTION!

_ To prevent ready2mains programmer from damage use an isolation transformer.

_ For HV tests ready2mains programmer must be disconnected.

_ It is necessary to connect a load!

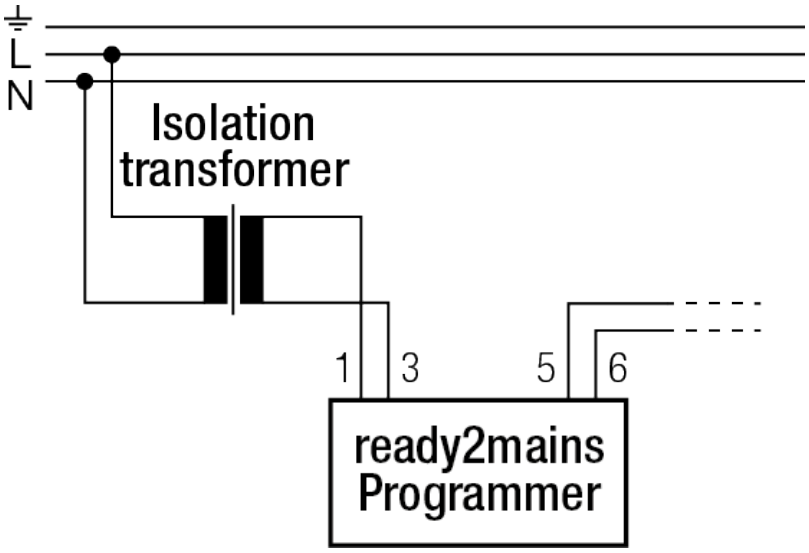
_ It is recommended to build up frame where you can put in the driver during programming to avoid electrical shock to the operator!

_ If more than one driver is connected consider the 400VA max load and the maximum of 5 drivers

**Isolation transformer:**

To prevent a damage to the ready2mains programmer it is recommended to use an isolation transformer between mains and ready2mains programmer. This is necessary because the ready2mains programmer is not able to recognize ground faults. If a ground fault occurs, it is possible that the output stage of the ready2mains programmer gets destroyed!

Connect the isolation transformer in this way:




**Tested transformator:**

_ RS Pro 500VA Isolating Transformer
Reference number at RS components: 504-228

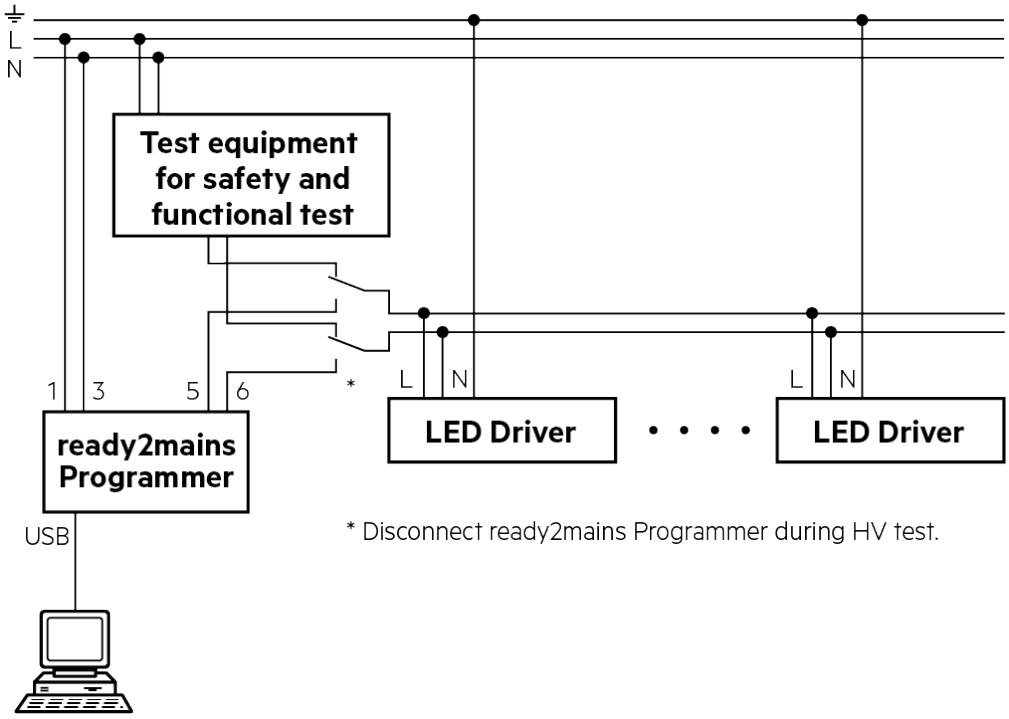
# Programming modes and examples

**CAUTION!**

_ The Isolation transformer should be protected against secondary short-circuit (B10 circuit breaker primary or a fuse secondary).

**Additionally used test equipment:**

To avoid wrong measurements and destruction of the ready2mains programmer, we strongly recommend to disconnect the ready2mains programming during additional testing procedures like high voltage testing.




**CAUTION!**

_ If ready2mains programmer is connected during HV testing it is possible that the ready2mains programmer is tested and not the luminaire itself!

## 6.3.2. Setup of deviceCONFIGURATOR

To use the companionSUITE with ready2mains programmer set up the deviceCONFIGURATOR as described below.

# Programming modes and examples

Consider that a manual programming start is only available if using ready2mains! Each programming cycle has to be started manually by the operator.
This is because of the one way communication of ready2mains. Because of this, only few settings affect the programming itself. All other settings that are relevant for other interfaces are ignored.

**Parameters to set up in deviceCONFIGURATOR which affect ready2mains programming:**

_ Interface settings

_ General settings / Reset to default and, write customized values

**Interface settings:**

Select the ready2mains Programmer as the interface for ready2mains.
No additional interface configurations are necessary.


⚠ CAUTION!

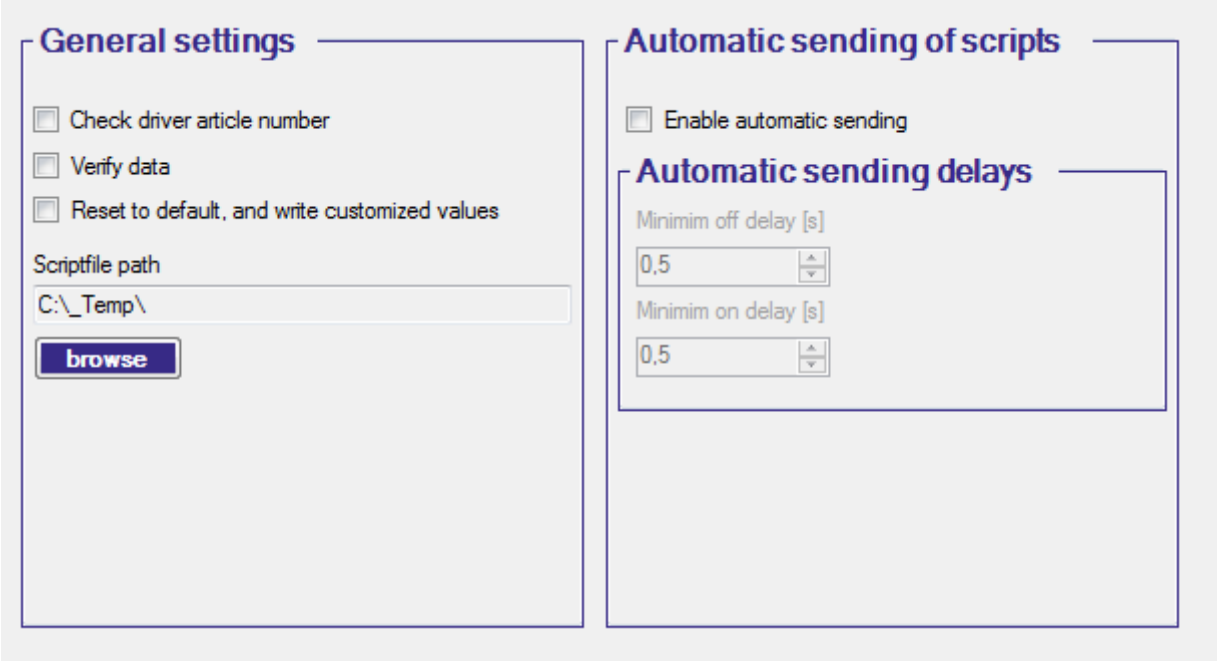
In some cases it can happen that the ready2mains programmer is not able to recognize optical feedback on ready2mains.

_ Only disable **Evaluate optical Feedback on ready2mains** if there are major problems during programming!

_ If **Evaluate optical Feedback on ready2mains** is disabled, it is essential to check if the optical feedback really appears.

**General Settings:**

# Programming modes and examples




In the general settings for ready2mains programming only the **Reset to default and write customized values** and the Batch settings have effect during programming.

_ **Reset to default and, write customized values** is necessary if the driver is not brand new from stock.
With this function it is possible to reset all parameters to default except for those that are set in the script!

NOTICE

If the setting **Reset to default and, write customized values** is enabled, this slows down the programming process a lot!

_ Use the setting **Reset to default and, write customized values** only for drivers that are not brand new from stock!

_ Batch Settings affect the batch overview in the **Work** window.

All the settings that are not usable in ready2mains are ignored! ready2mains uses one-directional communication. The only feedback that is recognized is optical feedback. After each save command this feedback is executed and recognized by the ready2mains Programmer!

Because of this form of communication, the following settings are ignored during ready2mains programming:

_ Check driver article number

_ Verify data

_ Enable automatic sending

NOTICE

From deviceGENERATOR >= 2.3 it is possible to set the functionality **Reset to default** in the configuration file.
In this case, the functions in the deviceCONFIGURATOR will be hidden.

### 6.3.3. Programming procedure in production

This procedure describes the workflow to program Tridonic LED Drivers via ready2mains in a production line.

CAUTION!

Before this procedure will work it is essential that:

_ Setup and wiring is built up correctly.

_ Interfaces are connected to the computer.

_ deviceCONFIGURATOR software is installed and running.

_ deviceCONFIGURATOR settings are correct (interfaces, general settings...).

_ Access to the configuration files is available.

# Programming modes and examples

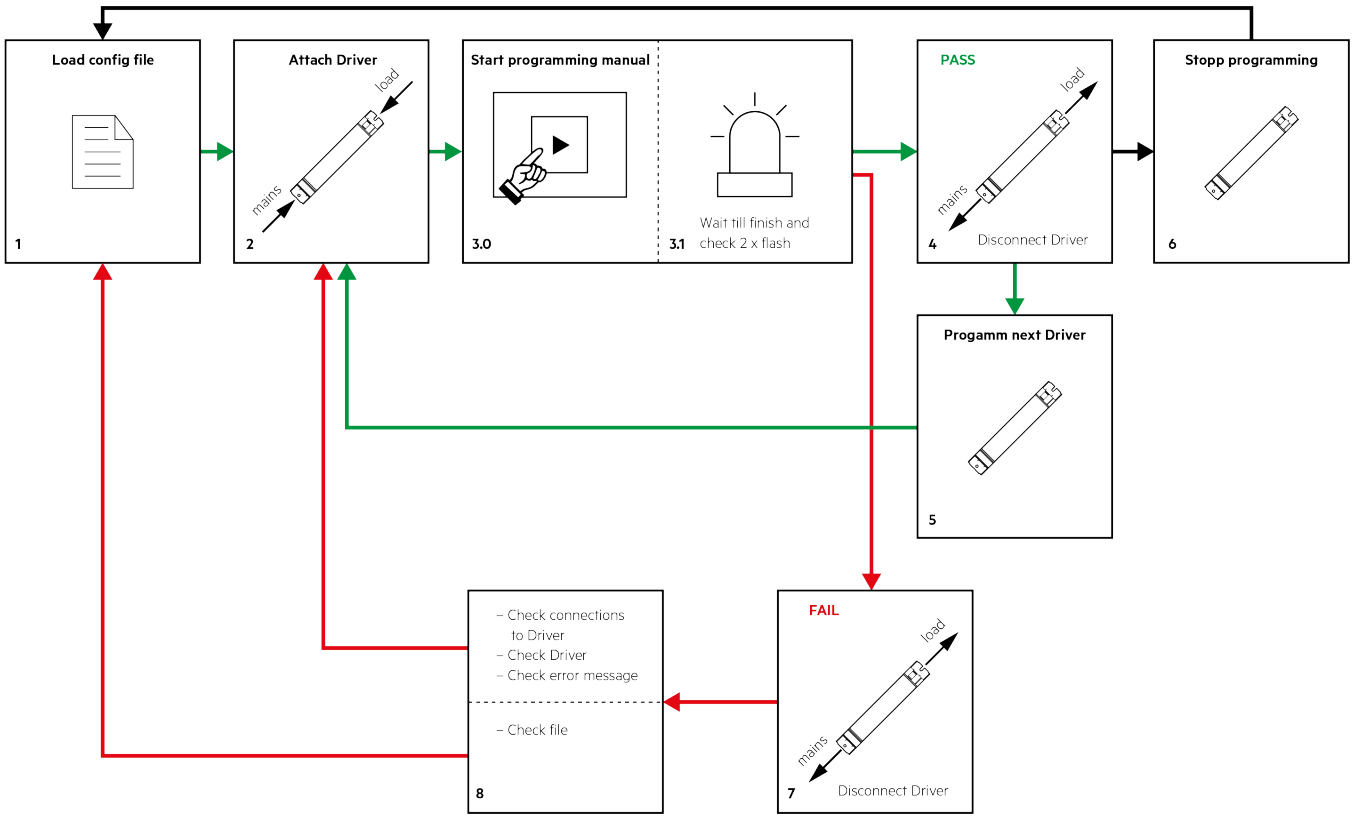
## Schematic of the programming workflow

In this schematic the principle workflow to program LED Drivers via ready2mains is illustrated.

| Colour | Description |
|---|---|
| **GREEN** | The green path in the schematic shows the workflow from loading the file and programming the drivers if everything works correct |
| **BLACK** | The black path in the schematic is the stop path and describes whats to do to stop the old batch an start a new batch |
| **RED** | The red path in the schematic is shows the workflow if the programming of a driver fails. |

**⚠ CAUTION!**

The following graph describes errors that can occur as a result of a wrong installation of the system!



# Programming modes and examples

Green path:

Black path:

| No. | Description |
|---|---|
| **6** | to stop the batch and start a new batch proceed with the black path and step number 1. |

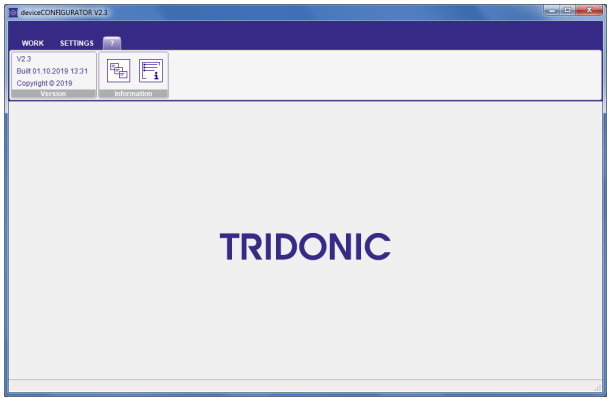
# Programming modes and examples

Red path:

| No. | Description |
|---|---|
| 7 | If the programming is finished with a fail the start button turns to a red colour and no optical feedback is visible! In this case the driver switch off the load without flashing.<br>_ Note: When printing is activated no label is printed in this case!<br><br>Disconnect the driver from the ready2mains programmer and from the load and proceed with red path and step number 8. |
| 8 | Before you proceed with a second try, do the following:<br>_ check the error message in the deviceCONFIGURATOR window<br>_ check the connections from ready2mains programmer to the driver and from the driver to the load<br>_ check if the correct driver is used<br><br>If the connections are bad or the wrong driver is connected proceed with the red path and step number 2.<br>If both is correct:<br>_ check if the correct configuration file is loaded<br><br>If the wrong configuration file is loaded, proceed with the red path and step number 1. |

# Help section

## 7.1. General information




In the help area, it is possible to open the help manual and to check the version of the software.

## 7.2. Version

Check the version of the software.

## 7.3. Information

| Image | Description |
|---|---|
|  | Open the user manual |
|  | Show disclaimer |

# History

| Version | Changes | Date | Comments |
|---|---|---|---|
| 2.3 | **New**<br>_ Consistent two-digit version number<br>_ Add to Logfile: "deviceGENERATOR version" (deviceGENERATOR version at file creation)<br>_ Add to Logfile: "Script downloaded" (Date and time at file creation)<br>**Change**<br>_ Now it is possible to set the function "Reset to default" into the configuration file.<br>**Eliminated defect**<br>_ Message "Negative optical feedback" by programming via ready2mains interface is fixed | 10. 2019 | |
| 1.6.85.13232 | **New**<br>_ Backwords compatibililty improved.<br>**Eliminated defect**<br>_ Deleting the deviceCONFIGURATOR "Settings" has been fixed.<br>_ A parameter file for the basicDIM DGC can be used again. | 06. 2019 | A fallback for unknown functions has been implemented into the deviceCONFIGURATOR. |

# History

| | | | |
|---|---|---|---|
| 1.5.81.16581 | **Maintenance**<br>_ Several smaller bugs solved<br>_ Manual expanded with programming examples | 03. 2019 | |

# History

| Version | Changes | Date | Comments |
|---|---|---|---|
| 1.5.77.16911 | **New**<br>_ Tunable White (DT8) functionality implemented<br>_ Batch settings and Operator ID moved to **WORK** window<br>_ Multiple file upload possible (limited to max 16)<br>**Maintenance**<br>_ Several smaller bugs solved<br>**Eliminated defect**<br>_ deviceANALYSERreader does not start.<br>_ Uninstall removes know all files.<br>_ LED current setting via ready2mains fixed. | 02. 2019 | DT8 is supported as of driver firmware version 1.6 or higher |
| 1.4.75.26433 | **New**<br>_ Added new logging parameter | 01. 2019 | |

# History

| Version | Changes | Date | Comments |
|---|---|---|---|
| 1.4.74.27494 | **Eliminated defect**<br>_ ready2mains file with only one parameter leads to an error. | 12. 2018 | |
| 1.4.73.15743 | **New**<br>_ NFC configuration is now possible | 10. 2018 | |
| 1.3.60.14607 | **New**<br>_ Supports .trgf files from deviceGENERATOR<br>_ NFC Drivers | 06. 2018 | |
| 1.2.59.16076 | Maintenance | 01. 2018 | Including<br>_ scriptGENERATOR V1. 5.107<br>_ DALIBusServer V3.0.0.57 |
| 1.1.5117890 | Maintenance | 06. 2017 | Including<br>_ scriptGENERATOR V1. 3.64<br>_ DALIBusServer V2.9.0.39 |
| 1.0.44.24976 | First Release | 04. 2017 | Including<br>_ scriptGENERATOR V1. 2.52<br>_ DALIBusServer V2.9.0.39 |