

# Helvar Driver Configurator

## USER GUIDE

**This user guide provides instructions on how to configure Helvar LED driver parameters and provides guidance for the use of the Helvar Driver Configurator program.**

### PARAMETER SETTING WITH HELVAR DRIVER CONFIGURATOR

The Helvar Driver Configurator is configuration tool created for controlling the configurable parameters of the LED driver. Once device data is read, all functions supported by the connected driver will appear on the "Device Parameters" tab. The configuration tool includes text fields where parameters can be modified: by typing into information fields or pressing buttons to select/enable/disable functionalities. An "Write" button in the "Device Parameters" tab uploads the modified parameters into the driver and verifies that configuration has succeeded.

The program is used to control and program manually or automatically the connected drivers. Also driver address copying and physical driver identification features are available. By "Create" button, user is able to create virtual configuration for specified driver without having connection to the LED driver, making the NFC operation easier as the driver is not mains powered in the configuration / programming operation.

The program is used for customizing the LED driver for a specific need. Configuration options vary between different LED driver types depending which features and functions they support.

# CONNECTION SETUPS

## DALI

### ELECTRICAL CONNECTION

*Safety warning: remember to turn off the electricity before disconnecting or connecting devices. Only a trained person should be doing this!*

The connection between the programming PC and the driver can be established for example with the following setup:

Devices needed:

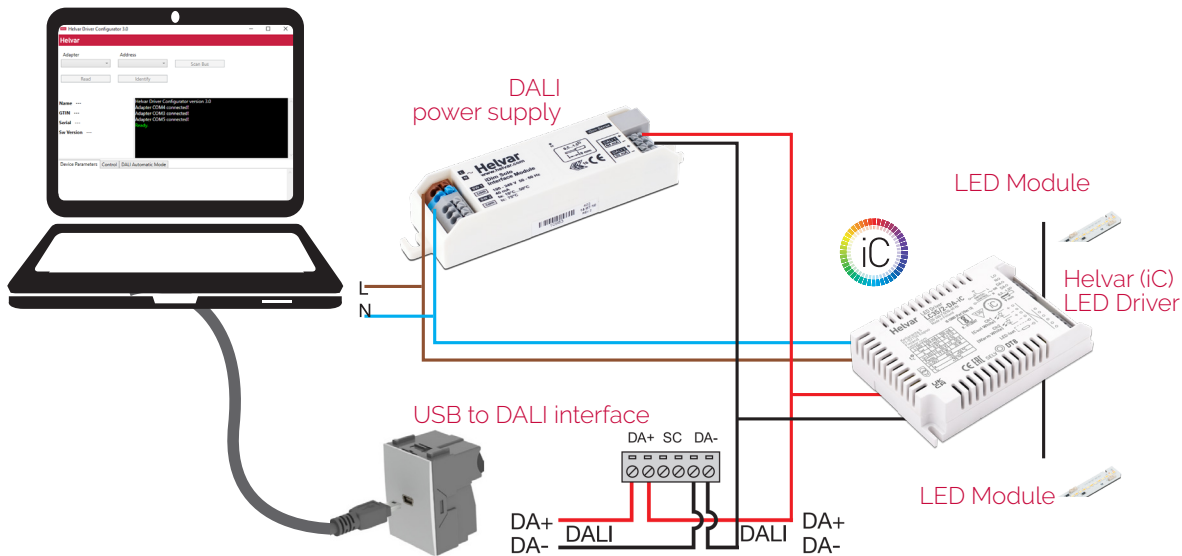
- DALI power supply (ie. Helvar 401, 402 or Helvar iDim Solo 403; iDim Solo 403 is used in this example.)
- Helvar USB to DALI Interface Unit 510 or Helvar DIGIDIM 505 Serial interface.

For installation of the software, run the installation file. Supported operating system is Microsoft Windows 10, Windows 7 and compatible.

Preliminary setup: The USB to DALI Interface unit is connected to the DALI power supply, and the driver under configuration with DALI bus. Please ensure correct polarity of the USB to DALI Interface and DALI power supply! Helvar LED driver DALI bus connections are unipolar and require no special attention.

The DALI power supply and the driver under configuration must also be provided with mains power.

Example setup to use when using Helvar DALI Driver Configurator:

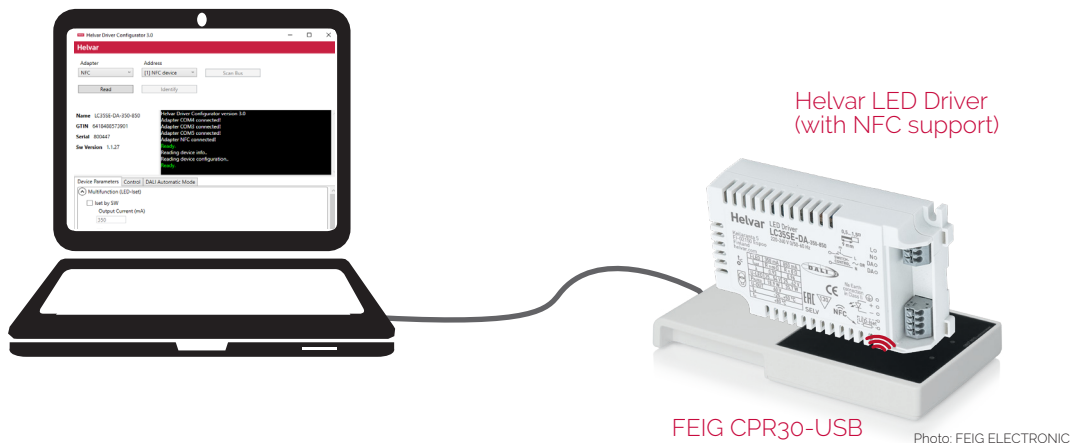


### NEAR-FIELD COMMUNICATION (NFC) CONNECTION

The connection between PC and the LED driver via NFC (Near-Field Communication) can be established with MD-SIG standardized CPR30-USB and ISC.MR102-USB NFC readers.

**Preliminary setup:** The NFC reader device drivers is installed and the reader is connected to the laptop before starting the software. For best wireless connectivity between driver and NFC reader, the NFC antenna (marked with **NFC** logo) is placed in middle of the black NFC contact area.

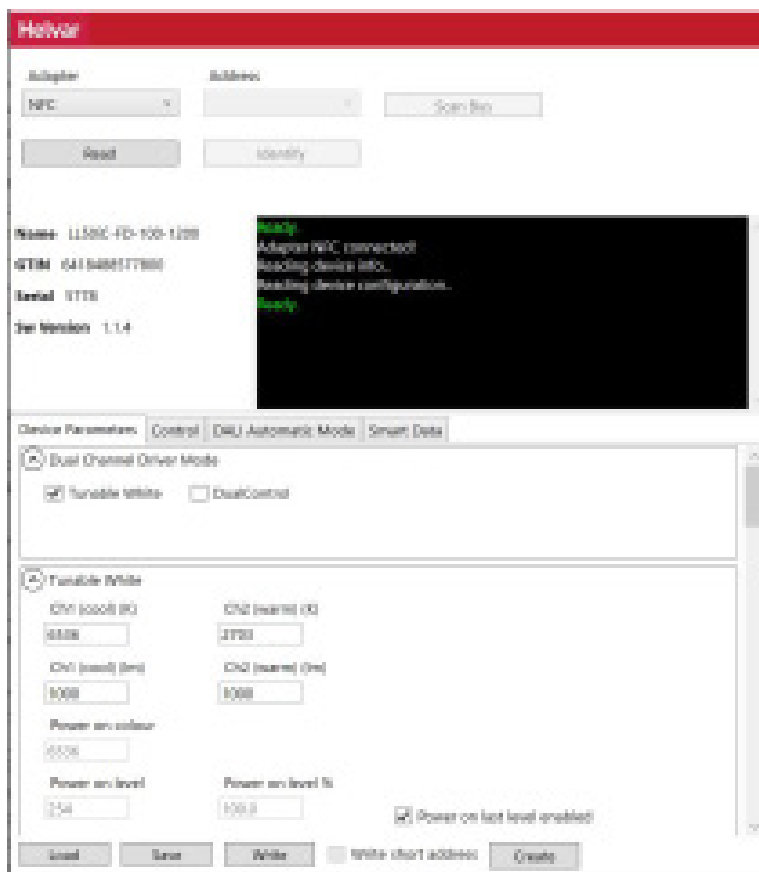
Example setup to use when using Helvar DALI Driver Configurator via NFC:



# CONFIGURATOR FEATURES

## MAIN WINDOW

The main part of the Configurator window is common for all sub tabs showing selected adapter selector, address selector, scan bus, read and identify buttons.



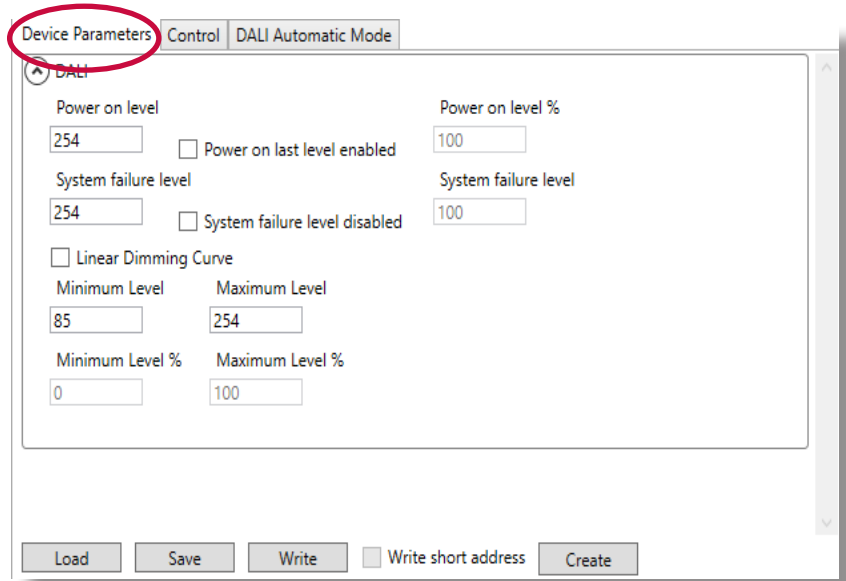
## MAIN WINDOW BUTTON FUNCTIONALITIES

Adapter selector	Select USB / COM port for DALI communication or NFC for near field communication. USB selection always refers to a connected Helvar 510 USB to DALI converter, while COM port might be any serial device. NFC selection refers to the connected NFC reader device. The user should know the actual COM port for their serial to DALI converter.
Address selector	Select the DALI address of the device. Use "Scan bus" to populate the list (initially the list is empty).
Scan Bus	Scans through DALI short address space to find devices connected to the selected DALI adapter. Updates the address list with found short addresses and device names.
Read	Reads the current configuration from the selected device. The configuration can then be edited in "Device Parameters" tab below.
Identify	The selected LED driver changes between MIN and MAX values, so it's possible to physically identify which device is currently being configured, as the light levels of connected LED modules change. Use this feature together with address selector to program a desired driver in an installation.

# CONFIGURATOR FEATURES

## DEVICE PARAMETERS

After reading the configuration from a selected device using "Read" button or after creating configuration with "Create", the "Device Parameters" tab displays the features of the device which can be configured.

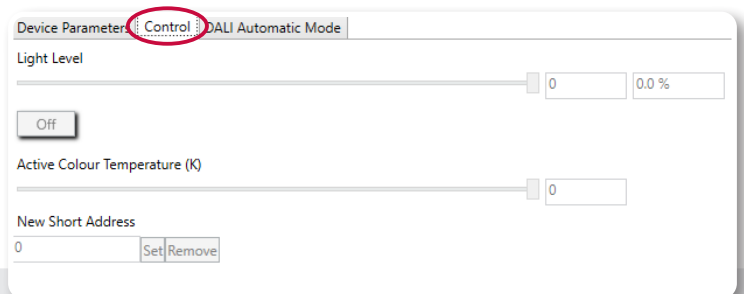


## GENERAL BUTTONS

<p><b>Load</b></p> <p>Load configuration from file. The parameters supported by the device type will be shown in "Device Parameters" tab.</p>	<p><b>Write</b></p> <p>Upload the current parameters to the device selected by "Adapter" and "Address" selectors.</p>
<p><b>Save</b></p> <p>Save configuration to a file.</p>	<p><b>Create</b></p> <p>Create virtual configuration file for specified LED driver without actual connection to the driver.</p>

## CONTROL

Driver operation can be tested in control window, where light level and colour temperature can be changed. Driver addressing and address removal are also possible. CONTROL features are not supported over NFC.



Light level	Real time control to set the light level of the device.
Active colour temperature	Real time control to set the colour temperature for Tunable White devices.
New short address	Select a new short address (1-64) for the device. The new short address is written using "Set" button.
Remove short address	Removes the short address from the device. Unaddressed devices can be configured when they are the only one on the DALI bus, in this situation Helvar Driver Configurator uses a special "broadcast" address.

# CONFIGURATOR FEATURES

## AUTOMATIC MODE

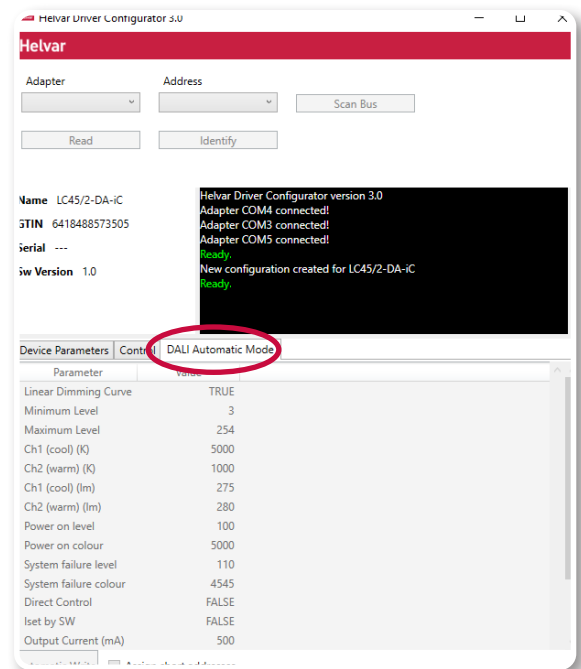
Automatic upload is a feature for serial configuration of LED drivers. When activated, the DALI bus is polled constantly to find devices compatible with current configuration and the settings are transferred to the device. The current configuration can be seen as a list in this view. This function allows multiple drivers to be programmed one after one, for example in luminaire manufacturing environment.

The settings for the Automatic Programming Mode are set in "Device Parameters". The parameter containing file can also be loaded in the "Device Parameters" and used in the Automatic Programming Mode. The parameter settings to be uploaded are listed in the "Automatic Mode" window. With "Assing short address" a new unique short addresses are assigned to devices that lacks them or have conflicting addresses. Already assigned unique short addresses are left as they are.

## AUTOMATIC UPLOAD

*Safety warning: remember to turn off the electricity before disconnecting or connecting devices. Only a trained person should be doing this!*

When waiting and scanning the bus, the green color indicator is shown - the user is free to connect new devices to the DALI bus. During the upload the colour indicator is red



## NOTE!

- Do not turn off LED drivers while the upload process is not finished.
- Status box shows information about the state of the automatic uploading process.
- When all your devices are configured, click "Automatic mode" button again to exit.
- The "Write log" option shall be activated before the configuration process starts, otherwise the data will not be collected to the log file.
- Either multiple drivers with no address, or one driver with an address can be programmed.
- This feature is designed for DALI line setup via USB or COM port. For NFC continous programming, use Helvar NFC Production Programmer tool, which is designed for optimal NFC programming.

## SMART DATA



LED drivers with Helvar Smart Data features can log and report information about operation and diagnostics. Data read from LED driver memory is presented under Smart Data tab of Helvar Driver Configurator.

Name	Value	Unit
Light Source Start Counter Resettable	1	-
Light Source Current	349.0	mA
Light Source Overall Failure Condition	0	-
Light Source Voltage	91.1	V
Light Source Start Counter	1	-
Light Source On Time Resettable	00:00:31	h:m:s
Light Source On Time	00:00:31	h:m:s
Light Source Overall Failure Condition Counter	0	-
Rated Median Useful Life Of Luminaire	255000	h
Internal Control Gear Reference Temperature	195	°C
Rated Median Useful Light Source Starts	6553500	-
Control Gear Start Counter	1	-
Control Gear Overall Failure Condition	0	-
Control Gear Operating Time	00:00:38	h:m:s
Control Gear Output Current Percent	100	%
External Supply Voltage	229.6	Vrms
Control Gear Temperature	26	°C
Control Gear Overall Failure Condition Counter	0	-
Active Power	36.5	W
Load Side Power	31.8	W
Active Energy	0.4	Wh
Load Side Energy	0.3	Wh

# CONFIGURATOR FEATURES

## "COMMAND PROMPT" MODE

Helvar Driver Configurator can also be used to transfer configurations without human interaction and without opening the normal operating window. This is mostly useful when LED drivers are configured as a part of an automated (luminaire testing) process. When preferred settings for the device in the graphical user interface has been created, save them to a file. This file can then be transferred to the device using command prompt arguments.

*Note: Either multiple drivers with no address, or one driver with an address can be programmed.*

*Note: The application must be run from the install directory*

## SYNTAX

`HelvarDriverConfigurator.exe <filename> <flags>` *NOTE: file names with blank spaces are not supported*

### Flags that can be used:

- p <port>** Adapter port to be used. USB for Helvar 510, or COM1 .. COM99 for serial.  
Default: USB
- a <address>** Short address of the device
- b** Use broadcast address. Can be used if there is only one device on the bus, even when using a device with no short address set.
- s** Scan DALI bus for compatible devices and upload the configuration to all of them
- x** Program missing or conflicting short addresses. This is used with the "scan" option in case the user wants to set new short addresses to devices without them (or have conflicting addresses). This is basically the same functionality as "assign short addresses" in User Interface mode. Already programmed unique short addresses are left as is.

*Note: flags -a, -b, -s all concern the device address so they are mutually exclusive*

**Example:** `HelvarDriverConfigurator exampleconfiguration.json -p USB -s -x`

This command transfers configuration file "exampleconfiguration.json" using adapter port USB, scans the DALI bus for compatible devices, and programs short addresses automatically.

## RETURN VALUES

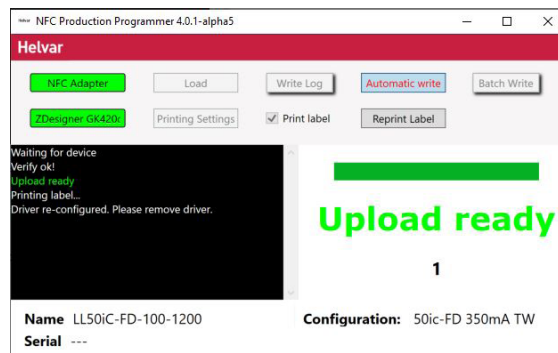
Helvar Driver Configurator exits with an error code giving information about the operation. If Helvar Driver Configurator is used as a part of an automated system, this return value can be handled in the script or application it is called from.

- 0** SUCCESS
- 1** ILLEGAL ARGUMENT
- 2** FILE NOT FOUND
- 3** ILLEGAL CONFIGURATION FILE
- 4** MULTIPLE RESPONSE
- 5** NO RESPONSE
- 6** DEVICE NOT COMPATIBLE
- 7** PORT NOT AVAILABLE
- 8** VERIFY FAILED
- 9** UNKNOWN ERROR
- 10** DALI BUS EMPTY

# NFC PRODUCTION PROGRAMMER

NFC Production Programmer is a tool designed for efficient and easy-to-use, continuous LED Driver configuration via NFC. It is designed to help the production line with easy-to-read graphical user interface and features which are designed in co-operation with luminaire manufacturers.

NFC Production Programmer can be used to configure LED drivers one by one or multiple LED drivers at ones in bulks. To configure multiple LED drivers in bulks FEIG XXX.XXXX.XXXX reader XXX.XXX.XXX antenna are required.



## CONFIGURATION PROCESS

1. Load correct configuration file created with Helvar Driver Configurator
2. Press "Write Log" button and name log-file in case logging of the process is needed
3. Select "Print Label" tick-box in case labels are needed for each configured LED driver
4. Select "Automatic Write" or "Batch Write" depending is the batch size known or not
5. Place LED driver or multiple LED drivers on the reader and follow the instructions on the screen

NFC Adapter	"NFC Adapter" box indicates, when the NFC reader is successfully connected and ready to use. The NFC reader must be connected to PC and the reader device drivers successfully installed before launching the NPP software. In case the NFC reader is not found / connected, the box remains gray and indicates with "No Adapter" the lack of NFC reader connection.
Load	Operator is able to load the pre-configured (with Helvar Driver Configurator) JSON driver parameter configuration file as setup for NPP. This must be done before starting the process.
Automatic Write	For continuous configuration of the LED drivers with loaded configuration settings. The program includes counter with increasing value so operator can easily follow the number of configured products. NPP features identical configuration detection, preventing incorrect increase of count number.
Batch Write	Same as automatic write, including batch size as well. The counter indicates the number of drivers configured and batch size (inserted by operator). If multiple LED drivers are configured as a bulk "Enable programming in bulk" and write amount of LED drivers in bulk to "Bulk size" view.
Write Log	When activated, the NPP writes log information of the configured devices to excel file. The operator is able to name and define the destination folder of the log file. The log file includes all set parameters as well as the result of configuration (FAIL / PASS) of each device. By default, the log file will be named as "HDC_{YearMonthDateHourSecond}".
Printing Setting	"Printing Settings" can be used to configure label printer. Installed label printer is selected from the "Printer Selection" drop-down menu and supported label sizes can be selected from the "Label dimensions (width, height) drop-down menu. Under "Label Settings" two freetext fields "Work order number" and "Driver batch" can be used to add additional information on the printed label. With "Go" button test label can be printed. NFC Production Programmer remembers these settings after program is shut down and restarted.
Print Label	When "Print label" tick-box is selected NFC Production Programmer prints automatically single label every time LED driver has been configured.
Reprint Label	"Reprint Label" prints again previous label in case first label is i.e. damaged.
Name / Configuration / Serial	Name is the name of the LED driver and configuration indicates the name of the configuration file. Serial number represents the running production number of the driver.



# NFC PRODUCTION PROGRAMMER

## SUPPORTED NFC READERS AND ANTENNAS

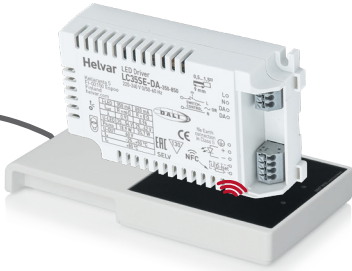
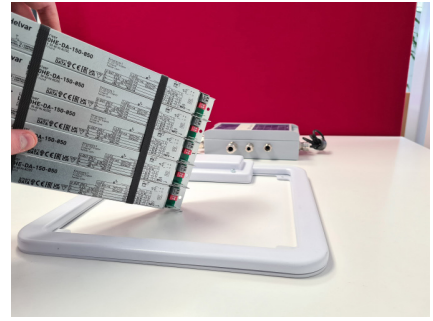


Photo: FEIG ELECTRONIC

FEIG CPR30-USB NFC Reader



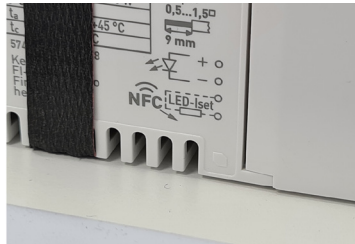
ID ISC.LR1002 HF Long Range NFC Reader



+  
FEIG ID ISC.ANT310/31 HF Antenna

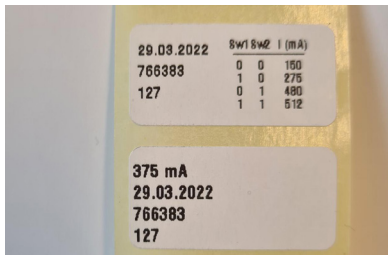
## GUIDELINES FOR PROGRAMMING MULTIPLE LED DRIVERS

- When programming multiple plastic-cased LED drivers, place a wrapped bundle of LED drivers within the HF antenna. Make sure NFC marking on the LED driver label is facing down, as in the picture on the right.



- When programming multiple linear metallic LED drivers, place a wrapped bundle of linear drivers within the HF antenna so that NFC antennas of the LED drivers are in the centre of the antenna. Hold the drivers in an angle so that NFC antennas of the LED drivers are facing the table. When NPP UI shows that all drivers are found, keep the drivers in that position until configuration is finished.

## SUPPORTED LABEL STICKER INFORMATION



Date	Automatic date when the driver is programmed.
Output current	The programmed output current of the LED driver will be printed on the label if the output current is set in the Helvar Driver Configurator.
Smart Switch output currents	In drivers where Smart Switch is supported, a table of the programmed output currents selectable with the DIP switch will be printed. It may be the default table from the factory, or then user-defined custom setting adjusted with Helvar Smart Switch feature.
Work order number	Optional free text field to be filled by the operator.
Driver batch	Optional free text field to be filled by the operator.

## SUPPORTED LABEL PRINTERS

Zebra GK/GX Series



# NFC PRODUCTION PROGRAMMER

## NOTE!

- The Helvar Driver Configurator and NFC Production Programmer must not be open at same time within same PC . The configuration gateway is active only in one program at time.
- Configuration file (JSON file) must be created with Helvar Driver Configurator before starting the process.
- The NFC reader must be connected to the PC before starting NFC Production Programmer.
- The label printer must be connected to the PC before starting NFC Production Programmer.
- The "Write log" option shall be activated before the configuration process starts, otherwise the data will not be collected to the log file.

## "COMMAND PROMPT" MODE FOR NFC CONFIGURATION

*NFC Production Programmer is capable of performing NFC configuration in Command Prompt mode without use of graphical UI.*

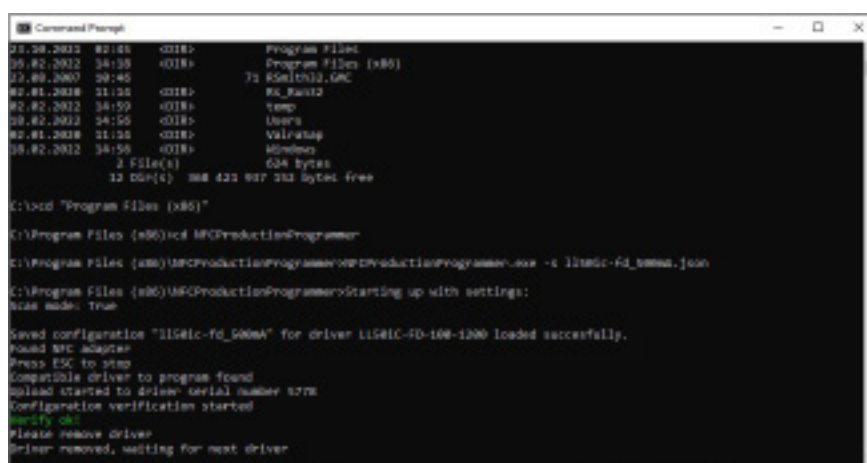
## SYNTAX

`NFCProductionProgrammer.exe <flags> <filename>`     *NOTE: file names with blank spaces are not supported*

Flags that can be used:

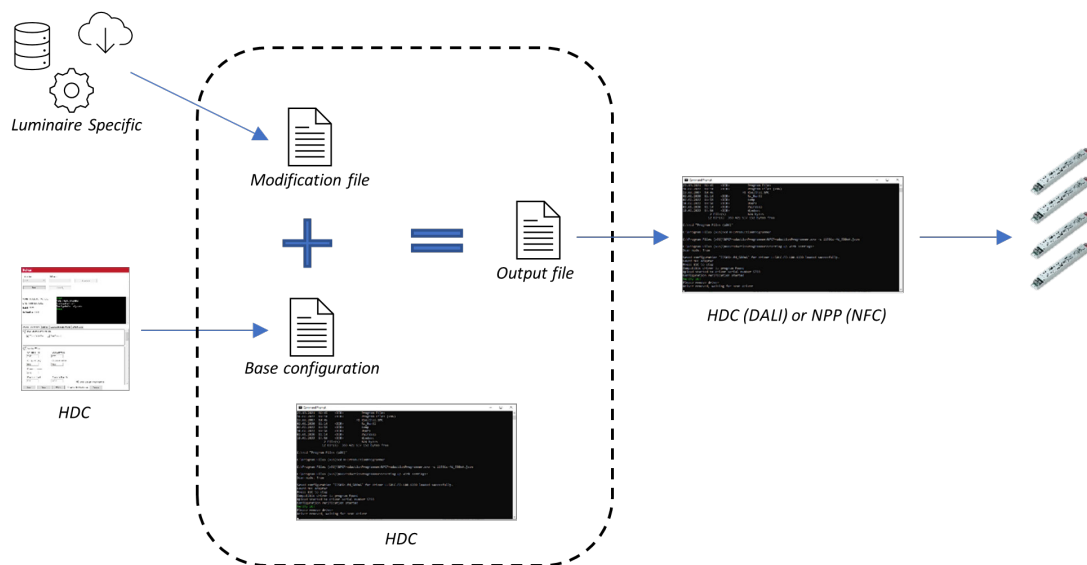
-s                    Configuration of several LED drivers. Without "-s" software won't wait for following LED drivers.

Example:     `NFCProductionProgrammer.exe -s exampleconfiguration.json`



# CONFIGURATION MODIFICATION MODE

Helvar Driver Configurator supports modifying driver configuration files with no user input, solely based on input files. This feature is only available when running HDC or NPP with no graphical user interface in Command Prompt Mode. Configuration Modification Mode enables user to export configuration parameters directly from ERP systems, cloud, or different databases.



## FILES

### Base configuration

The Base configuration is a standard configuration file created with HDC tool for specific Helvar LED-driver type which includes all possible configuration parameters.

### Modification file

The Modification file defines the changes to be applied to the Base configuration. Modification file is created by the user, and it includes only those luminaire specific parameters which customers extracts from their own database. The changes are mapped via the labels used in the normal UI of HDC. Rest of the configuration parameters will stay the same as they are in Base configuration.

### Output file

The Output file is the driver configuration which is written to LED-Driver. Output file has the Base configuration and Modification file parameters combined. Combining these 2 parameter files is necessary to make sure that parameters imported from external database are correct and compatible with HDC and NPP tools after software updates of these tools.

The name of the Output file may be left as the default or defined explicitly. The default name for the Output file is generated by appending the Modification file name to the Base configuration filename, separated by a '+' symbol. For example, with a Base configuration named 'driver.json' and Modification file name 'modification1.json' the resulting default Output filename will be 'driver+modification1.json'.

## EXECUTABLE ARGUMENTS

The relevant arguments for configuration modification mode define the filenames of the files in the Files section.

- The Base configuration file path is always given as the first argument
- --modifyWith or -m: This required argument must be followed with the Modification file path.
- --modifyOutputFile or -o: This optional argument must be followed with the output file path.

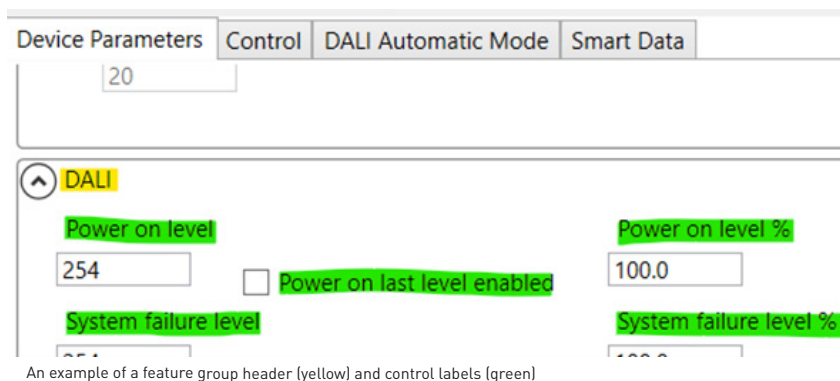
Example:

```
HelvarDriverConfigurator.exe base.json --modifyWith mod.json --modifyOutputFile base_with_mod_applied.json
```

# CONFIGURATION MODIFICATION MODE

## MODIFICATION FILE FORMAT VERSION 1

The Modification file has the fields 'version' and 'changes'. Version (int) should be 1. Changes defines an array of change objects. A Change object has the fields 'group', 'control' and 'action'. Group should be a string defined to match a UI feature group header, and Control should be a string defined to match a UI control label. Examples of these are shown in figure 1. Action are action objects defined in the following section.



### Action Objects

Actions are defined as a combination of a verb and optional parameters

Defined verbs:

- SetFloat
- SetInteger
- SetTableInteger
- SelectDropdown
- ClickCheckbox

### Set Float Info

Parameters:

- 'float\_value': a JSON number, which will be deserialized to a floating-point number.

Is compatible with both integer and float controls (with and without . in the number). If a float value is given for an integer control the value will be rounded to the nearest integer. .5 will be rounded to the nearest even number.

### Set Integer Info

Parameters

- 'integer\_value': a JSON number, which will be deserialized as an integer value.

Similarly compatible with both types of number control.

### Set Table Integer

Used for setting values in dipswitch table controls. Has a more complex definition, since multiple parameters are defined as multiple rows for one control in the UI.

Parameters:

- 'table\_integer\_changes': A JSON object with one required string 'type' matching either "2\_ds" or "3\_ds" matching the number of dipswitches displayed in the UI. Two optional fields "2\_ds\_changes" and "3\_ds\_changes" are for defining the desired array of changes. Only the one matching the type will be parsed. They contain the following definitions
  - 2\_ds\_changes: array of: {'sw1': uint, 'sw2': uint, 'current': uint}. sw1 and sw2 must be either 0 or 1, matching the values displayed in the UI.
  - 3\_ds\_changes: array of: {'sw1': uint, 'sw2': uint, 'sw3': uint, 'current': uint}. Same as above.

### Select Dropdown

Parameters

- 'dropdown\_decimal\_id\_value': a JSON number, deserialized as a floating-point number that matches an option visible in the desired dropdown.

# CONFIGURATION MODIFICATION MODE

## Click Checkbox

Parameters: None

Will perform the change that clicking on the UI checkbox would have done. This means an unchecked checkbox will be checked, and a checked checkbox will be unchecked if it is allowed. An error will be given when attempting to uncheck a checkbox that does not allow unchecking.

## ON APPLYING CHANGES

Changes can only be applied if they can be applied in the graphical user interface. If a control is disabled, indicated by being grey, it will not be possible to apply changes to it.

## APPENDIX

LL80HE-DA-150-850 Modification json-file example:

- Output Current: 650mA
- CLO: Enabled
- CLO Level: 75%
- CLO life time: 60000h
- DALI Power on last level: Enabled
- Linear dimming curve: Enabled
- Minimum level: 90

```
{
  "version": 1,
  "changes": [
    { "group": "Output Current", "control": "Override DIP switch currents", "action": { "verb": "ClickCheckbox" } },
    { "group": "Output Current", "control": "Output current (mA)", "action": { "verb": "SetInteger", "integer_value": 650 } },
    { "group": "Constant Light Output", "control": "CLO Enabled", "action": { "verb": "ClickCheckbox" } },
    { "group": "Constant Light Output", "control": "CLO level (%)", "action": { "verb": "SetInteger", "integer_value": 75 } },
    { "group": "Constant Light Output", "control": "CLO life time (x1000h)", "action": { "verb": "SetInteger", "integer_value": 60 } },
  ],
  { "group": "DALI", "control": "Power on last level enabled", "action": { "verb": "ClickCheckbox" } },
  { "group": "DALI", "control": "Linear Dimming Curve", "action": { "verb": "ClickCheckbox" } },
  { "group": "DALI", "control": "Minimum Level", "action": { "verb": "SetInteger", "integer_value": 90 } },
}
}
```

## TRIPLE DIP SWITCH -example:

```
{
  "version": 1,
  "changes": [
    {
      "group": "Output Current",
      "control": "DIP switch combination output current values",
      "action": {
        "verb": "SetTableInteger",
        "table_integer_changes": {
          "type": "ds_3",
          "ds_3_changes": [
            { "sw1": 0, "sw2": 1, "sw3": 0, "current": 321 }
          ]
        }
      }
    }
  ]
}
```

## CONFIGURATION MODIFICATION MODE

### SET INTEGER -example:

```
{
  "version": 1,
  "changes": [
    {
      "group": "DC Emergency",
      "control": "Level (%)",
      "action": {
        "verb": "SetInteger",
        "integer_value": 51
      }
    }
  ]
}
```

### SET FLOAT -example:

```
{
  "version": 1,
  "changes": [
    {
      "group": "DC Emergency",
      "control": "Level (%)",
      "action": {
        "verb": "SetFloat",
        "float_value": 51.0
      }
    }
  ]
}
```

### SELECT DROP DOWN -example:

```
{
  "version": 1,
  "changes": [
    {
      "group": "Corridor Control",
      "control": "1. Fade in time (s)",
      "action": {
        "verb": "SelectDropdown",
        "dropdown_decimal_id_value": 45.3
      }
    }
  ]
}
```

### DUAL DIP-SWITCH -example:

```
{
  "version": 1,
  "changes": [
    {
      "group": "Output Current",
      "control": "DIP switch combination output current values",
      "action": {
        "verb": "SetTableInteger",
        "table_integer_changes": {
          "type": "ds_2",
          "ds_2_changes": [
            { "sw1": 0, "sw2": 1, "current": 333 }
          ]
        }
      }
    }
  ]
}
```

# CONFIGURATION MODIFICATION MODE

## CLICK CHECKBOX -example:

```
{
  "version": 1,
  "changes": [
    {
      "group": "DC Emergency",
      "control": "Disabled",
      "action": {
        "verb": "ClickCheckbox"
      }
    }
  ]
}
```