Getting started with STSW-L98GD8 software package for EVL-L98GD8

Introduction

This document describes the Graphical User Interface (GUI) that allows to initialize and control the EVL-L98GD8 evaluation board. It's detailed on how to change the L98GD8 device parameters through the SPI protocol, manage the device inputs with parallel GPIO or SPI interface, enable the PWM feature on the device outputs. The EVL-L98GD8 GUI has been developed using LabVIEW and it uses, as Microcontroller interface, the SPC56M-DIS Discovery+ evaluation board.

A section in this document is dedicated to describe how to install and configure the required LabVIEW 2016 Runtime environment, and the needed drivers. Another section is used to specify how to program the proper control firmware onto the microcontroller board, which is part of the STSW-L98GD8 software package.

1 Get software

To setup and control the EVL-L98GD8 evaluation board through the dedicated Graphical User Interface, a software environment must be acquired, and specifically:

1. The STSW-L98GD8 software package, which includes the GUI application and the control firmware to flash into the SPC56M-DIS Discovery+ MCU board.
2. A software ecosystem composed by:
   - LabVIEW 2016 Runtime engine
   - NI-VISA 16.00 Runtime drivers
   - FTDI Chip drivers

1.1 Graphical User Interface and Control Firmware

Visit the dedicated page for STSW-L98GD8 on https://www.st.com: from the first tab **Overview** you can access a button to get the latest release of the software package as a zip file. It includes the firmware package to control the EVL-L98GD8 evaluation board, and a folder with the GUI application to be launched on a desktop PC to communicate with the L98GD8 device.

Select the tab **Documentation** to retrieve the available documentation about the software package, and the tab **Tools & Software** to find the reference to the EVL-L98GD8 ST web page.

1.2 Software ecosystem

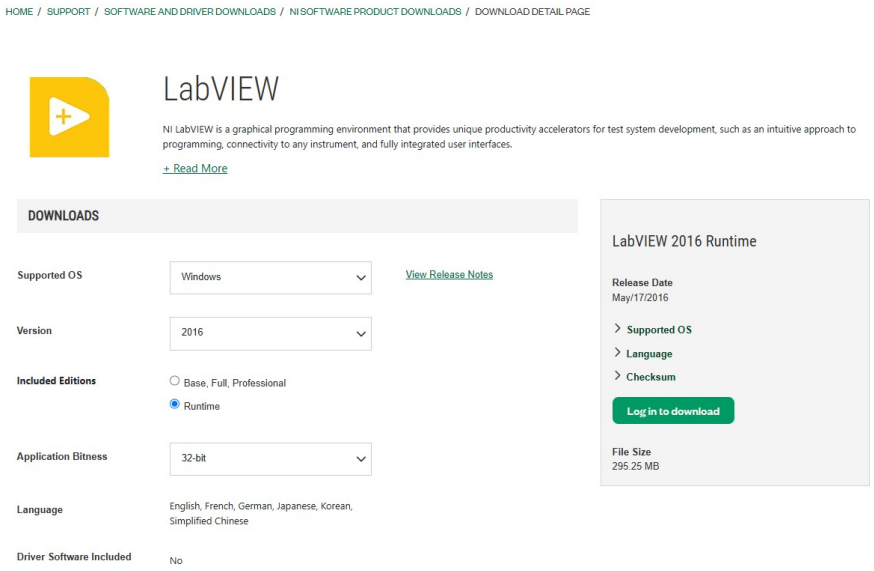
The Graphical User Interface has been developed with LabVIEW from National Instruments. To launch the GUI application, a licensed version of LabVIEW is not required; installing the LabVIEW runtime engine and related drivers, available for free on National Instruments web portal, is sufficient.

All of the software needed to build the proper ecosystem is listed below:

LabVIEW 2016 Runtime

The LVRTE2016_f7Patchstd.zip software package is available at
https://www.ni.com/en/support/downloads/software-products/download.labview.html#310816

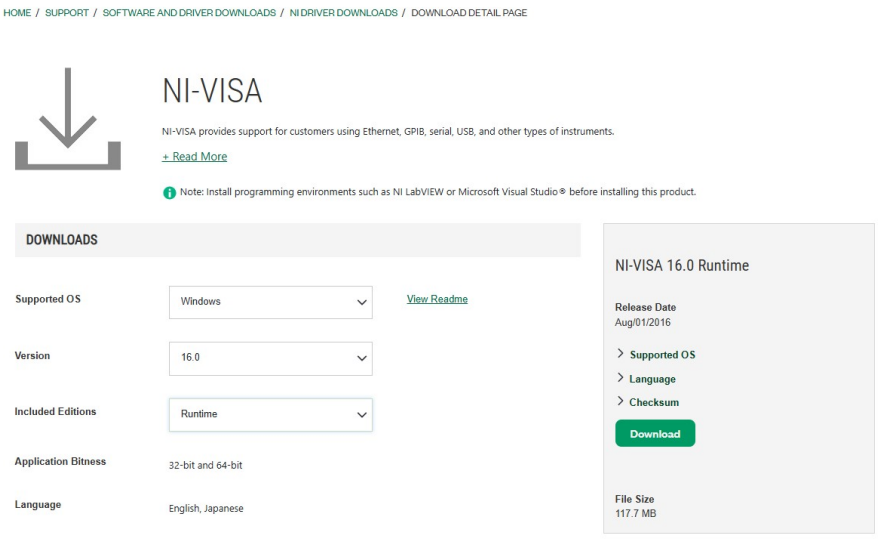
Figure 1. LabVIEW 2016 Runtime



NI-VISA 16.00 Runtime

The NIVISA1600runtime.zip software package is available at
https://www.ni.com/en/support/downloads/drivers/download.ni-visa.html#306112

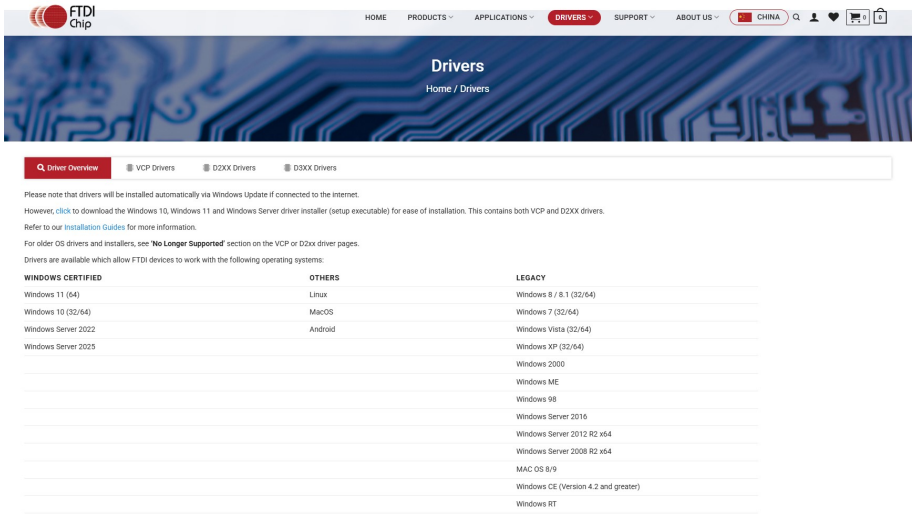
Figure 2. NI-VISA 16.00 Runtime



FTDI Chip Driver

The CDM2123620_Setup.zip software package is available at
https://ftdichip.com/drivers/
Windows 11 version available here: https://ftdichip.com/wp-content/uploads/2025/03/CDM2123620_Setup.zip

Figure 3. FTDI Chip driver



2 Software installation

The software ecosystem needed to setup a PC workstation to run the Graphical User Interface provided in the STSW-L98GD8 software package is now described.

2.1 Firmware

The Graphical User Interface running on the PC workstation needs a dedicated control firmware to communicate with the L98GD8 device present in the EVL-L98GD8 evaluation board. The proper control firmware is included in the STSW-L98GD8 software package and, if not already done, it must be flashed into the SPC56M-DIS Discovery+ evaluation board. The detailed procedure belowed is to be followed to flash the control firmware onto the microcontroller board.

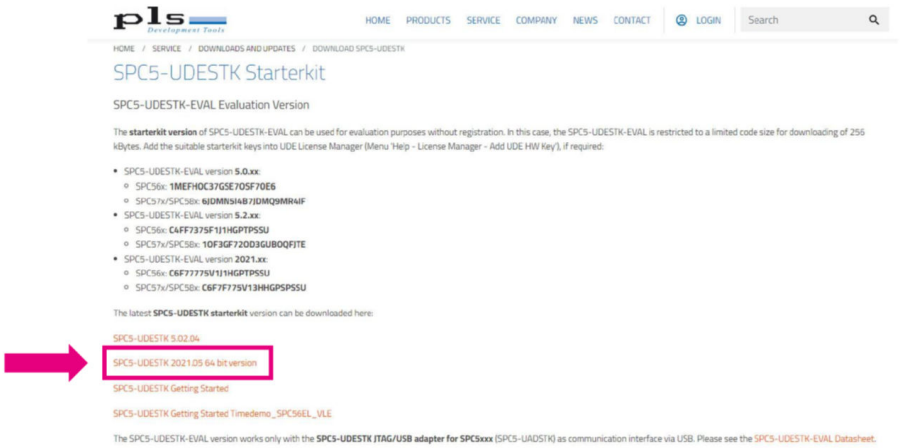
To perform the firmware programming operation, an additional component hardware is needed: the SPC5-UDESTK. This tool is a USB/JTAG interface to enable the debugging and programming of SPC5x MCUs. To use this hardware component, the related software ecosystem must be preliminarily installed.

Figure 4. SPC5-UDESTK USB/JTAG interface

The software and drivers needed to manage the SPC5-UDESTK USB/JTAG interface can be downloaded from the following link: https://www.pls-mc.com/service/downloads/download-spc5-udestk/.

Install the UDE tool **SPC5-UDESTK 2021.05 64-bit version** as reported here below:

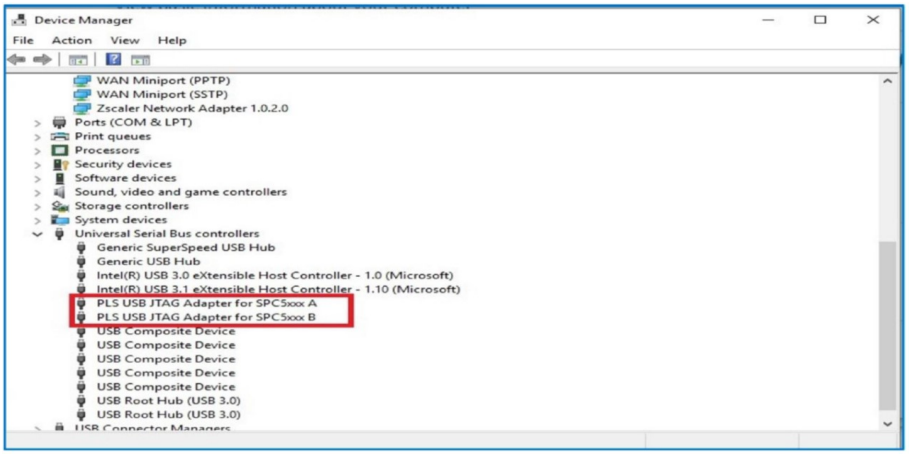
Figure 5. PLS UDE Starterkit 2021



When software and driver installation is finished, insert the SPC5-UDESTK USB/JTAG interface into the JTAG connector available in the SPC56M-DIS Discovery+ evaluation board and, using a USB A to Micro USB cable, connect the USB/JTAG interface to an available USB port in your PC workstation.

When opening the Device Manager in Windows OS, these two PLS controllers must have been recognized if everything went well.

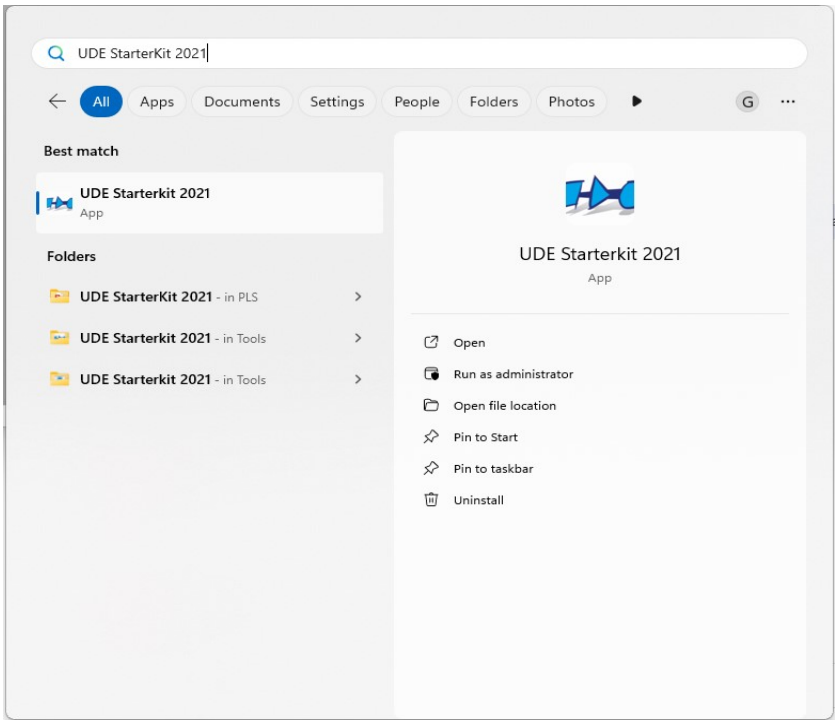
Figure 6. Universal Serial Bus controllers



The steps requested to program the control firmware onto the microcontroller board are detailed below.

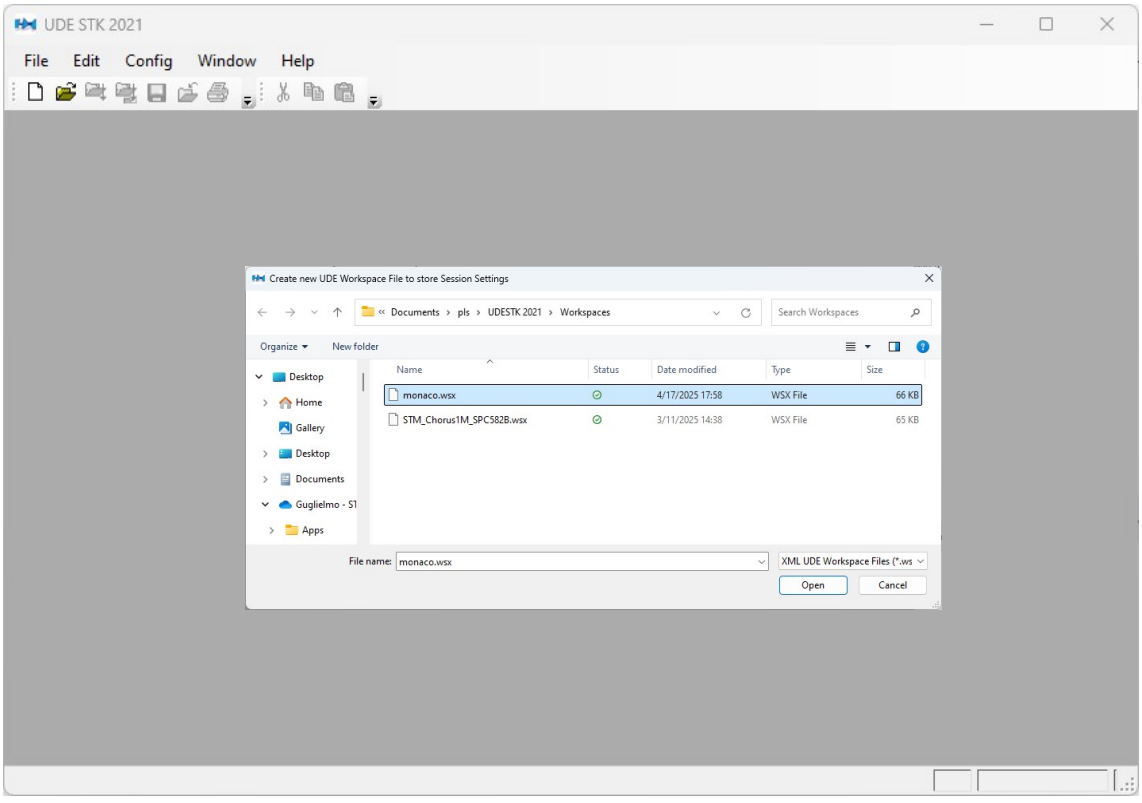
1. Run **UDE Starterkit 2021** application

Figure 7. UDE Starterkit 2021



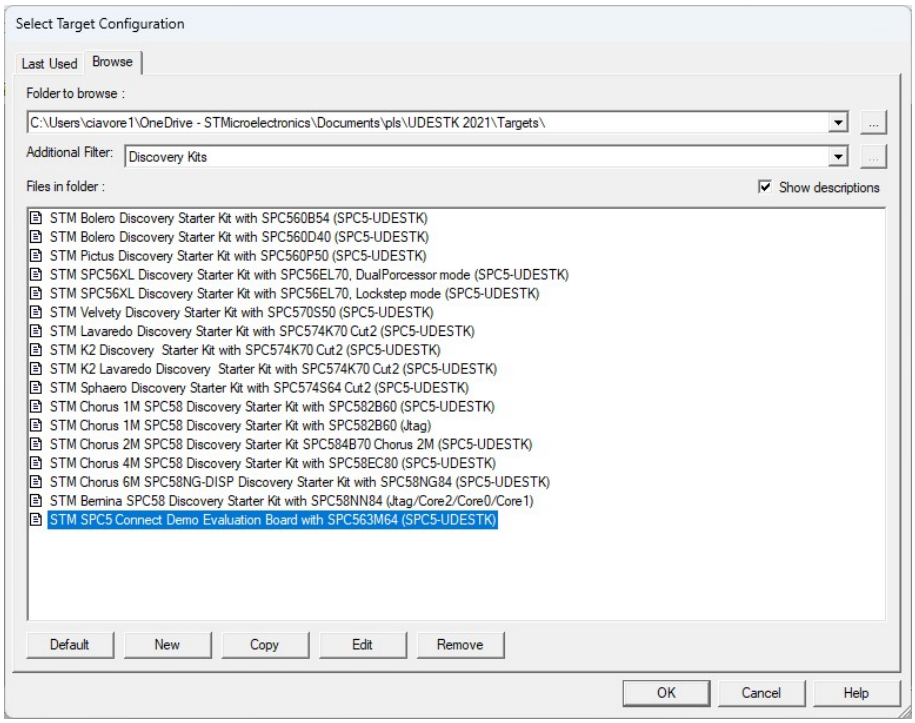
2. Select from the **File** menu, choose the item **New Workspace** and enter a name for the workspace. If a workplace with that name already exits, it will be overwritten (in this example the new workspace is named monaco.wsx)

Figure 8. New workspace creation



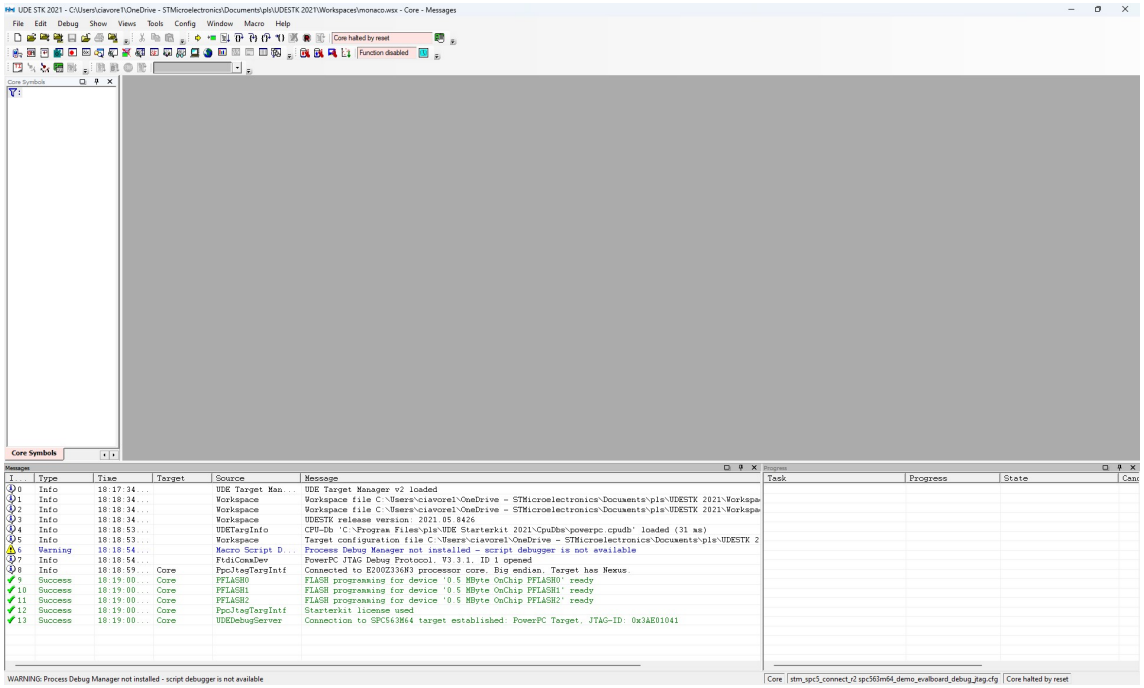
3. **Select Target Configuration** window appears: select the item named **STM SPC5 Connect Demo Evaluation Board with SPC563M64 (SPC5-UDESTK)** and press **OK**

Figure 9. Select Target Configuration dialog window



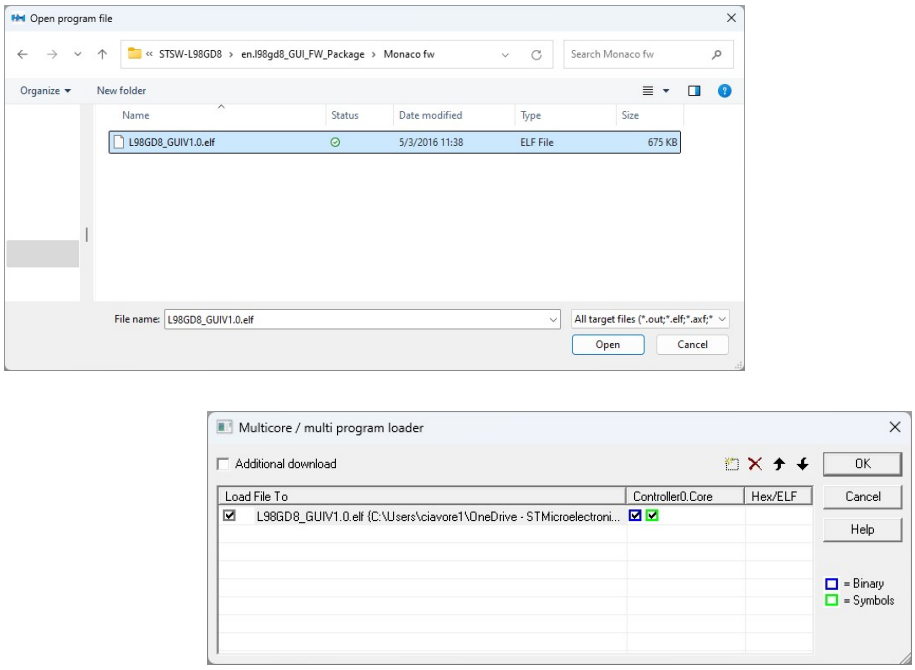
4. In the **UDE STK 2021** window, check the **Messages** window for a line indicating that a successful connection to the target has been established

Figure 10. Connection established



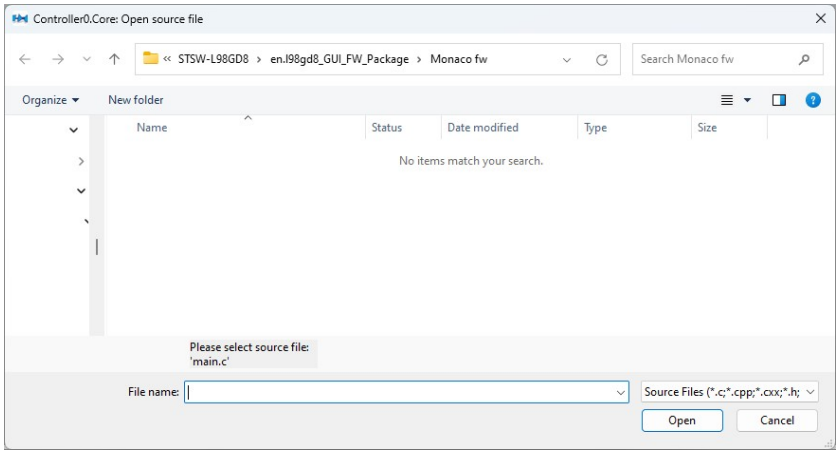
5. Select **Load Program** from **File** menu, browse your hard drive to select the provided control firmware (**.elf** file) and then press **OK** in the **Multicore / multi program loader** dialog window

Figure 11. Load Program dialog windows



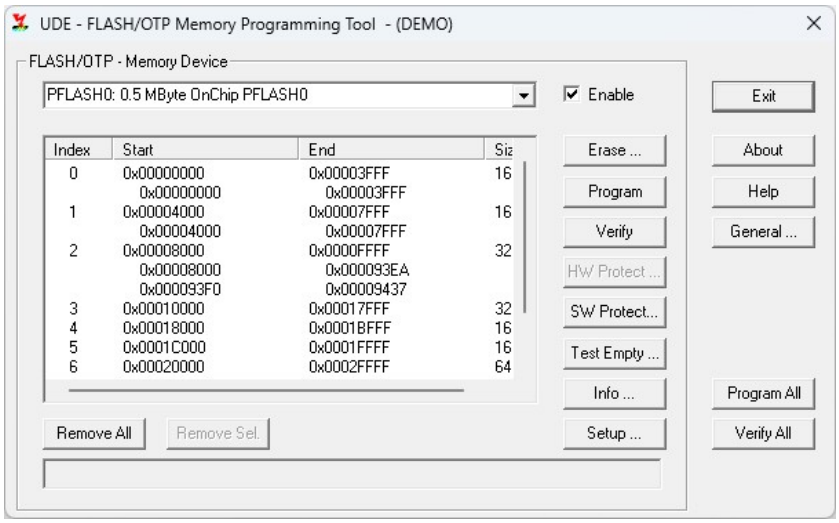
6. Press **Cancel** for any request about Controller0.Core source files (**.c** or **.h**)

Figure 12. Additional source files request



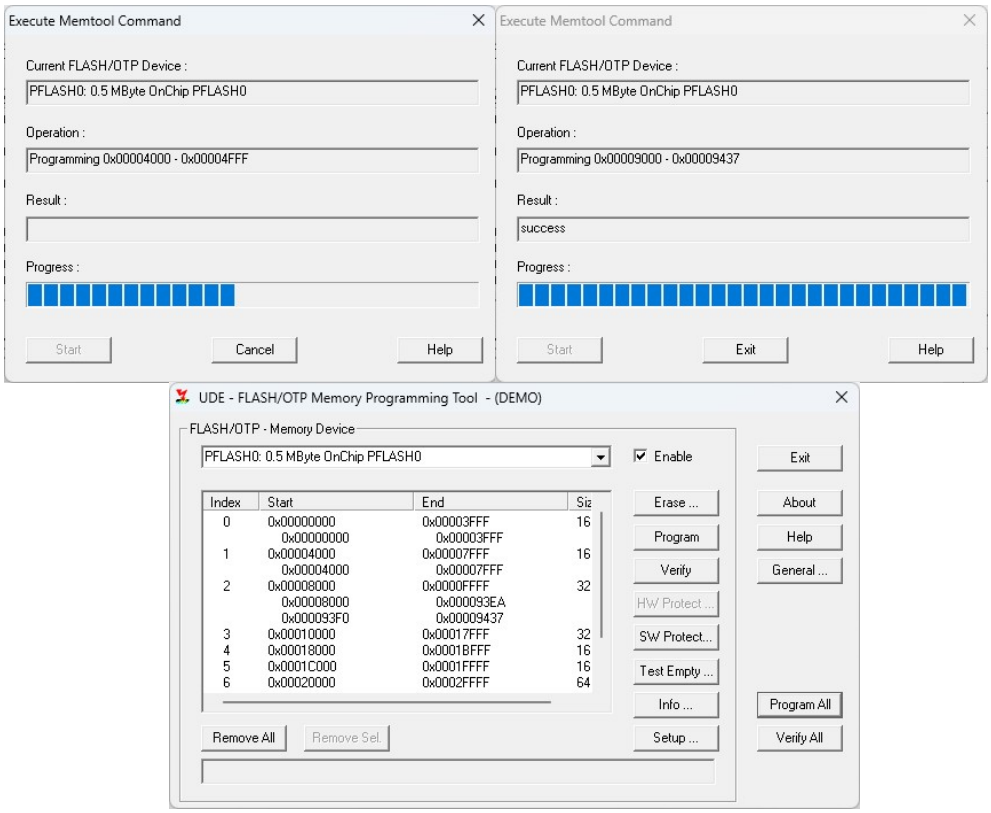
7. The **UDE - FLASH/OTP Memory Programming Tool - (DEMO)** window now appears. Press **Program All**

Figure 13. FLASH/OTP Memory Programming Tool dialog window



8. Follow the execution of programming operation through the **Execute Memtool Command** dialog window. At the end of the operation, check the **Result:** field reporting **success**, and then press **Exit**

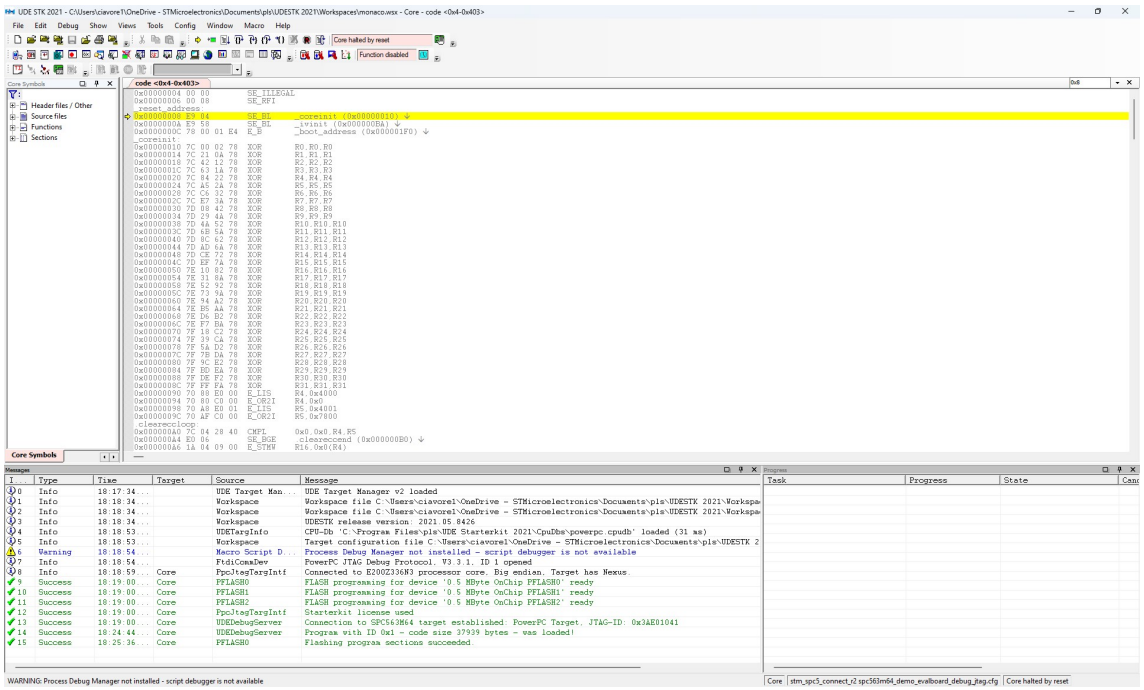
Figure 14. FLASH/OTP Memory Programming in progress



9. Press **Exit** also in the **UDE - FLASH/OTP Memory Programming Tool - (DEMO)** dialog window

10. The **UDE STK 2021** main window now will report a message about the successful flash programming operation

Figure 15. FLASH/OTP Memory Programming completed



11. To complete the firmware programming, simply disconnect the SPC5 target board and reset it using the reset push-button on the microcontroller board

Figure 16. Target disconnection

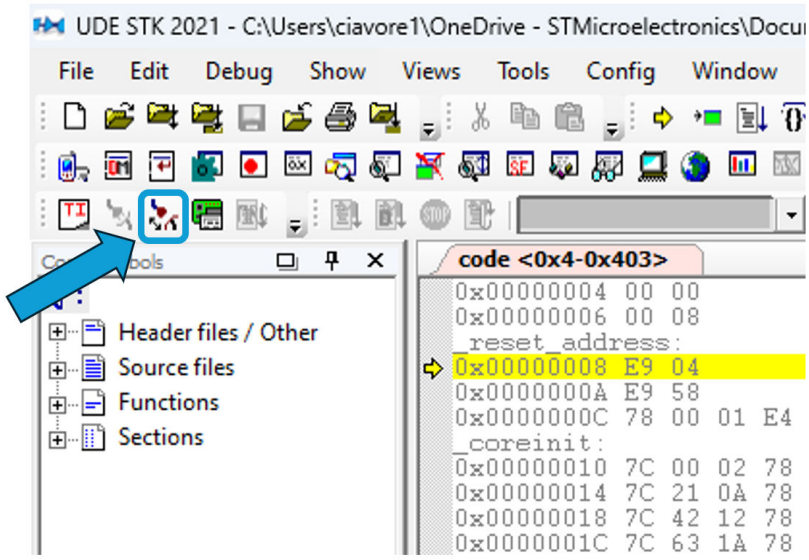
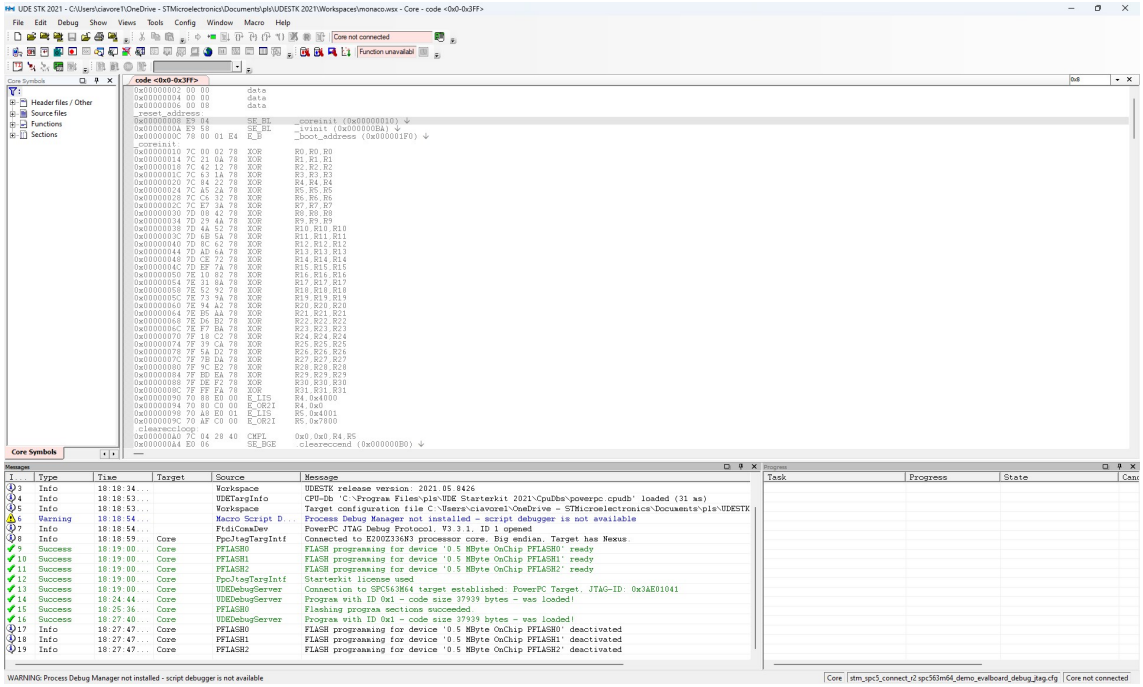


Figure 17. Target disconnected



2.2 Graphical User Interface

The software package is provided in the form of a zip archive. This archive includes the GUI application and the control firmware to be downloaded onto the microcontroller board.

The Graphical User Interface application is provided as a portable Windows application that can be immediately used, no installation process needed, if the proper LabVIEW ecosystem has been installed in the PC workstation. The only action requested is to extract from the STSW-L98GD8 software package in the provided archive, and save it on the PC hard drive.

After locating the **PC GUIx.y** folder in the extracted archive, to start the GUI application, you just need to run the **L98GD8_vx.y.exe** executable

Note: x.y is a generic version number, reported only as an example

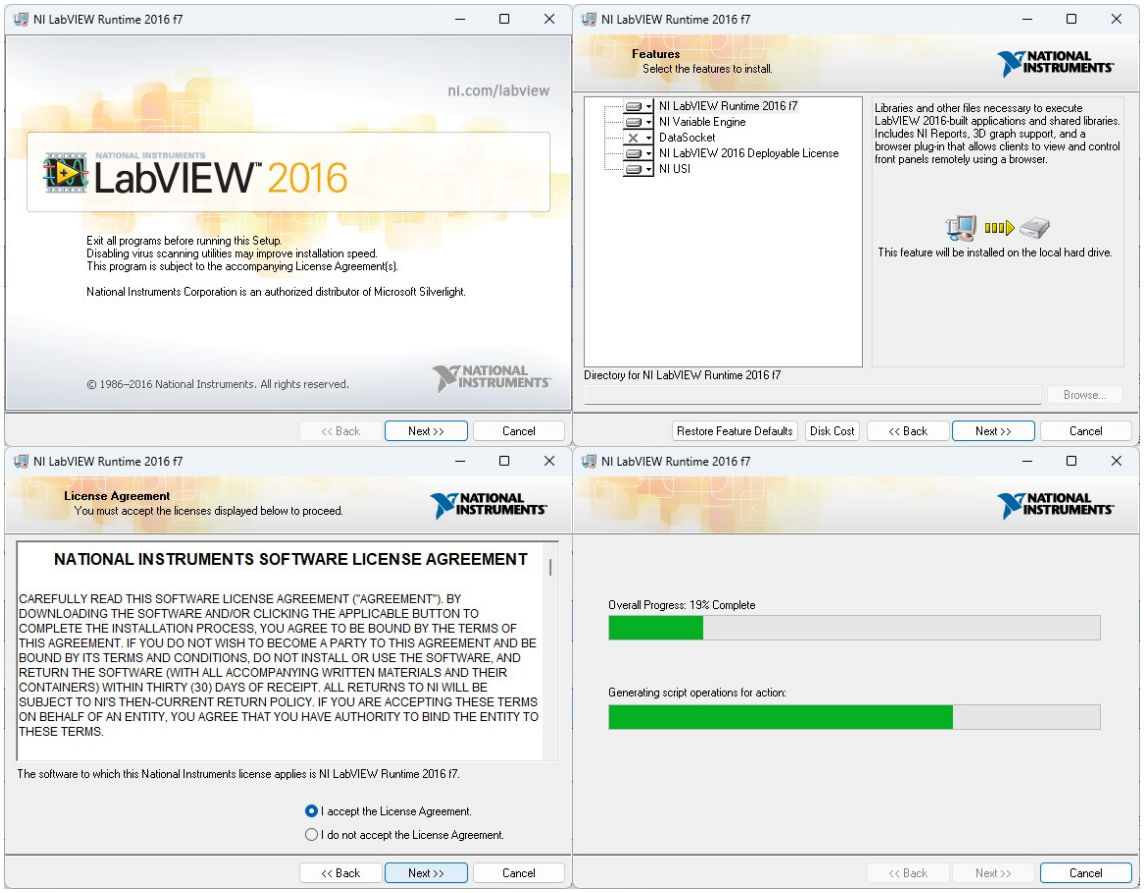
2.3 Software ecosystem

To setup your desktop PC to run the Graphical User Interface included in the STSW-L98GD8 software package, some software installations are required.

LabVIEW 2016 Runtime (32-bit)

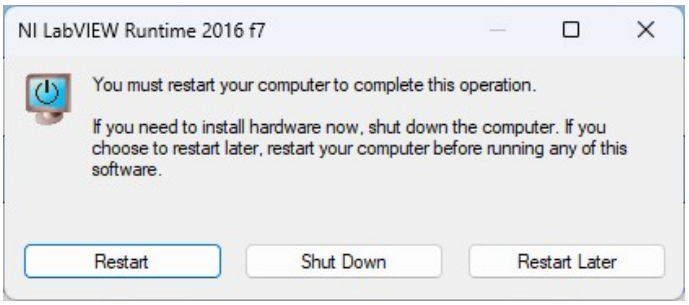
Run the LVRTE2016_f7Patchstd/setup.exe utility and follow the installation wizard:

Figure 18. LabVIEW 2016 Runtime



After installation is completed, a restart message is presented

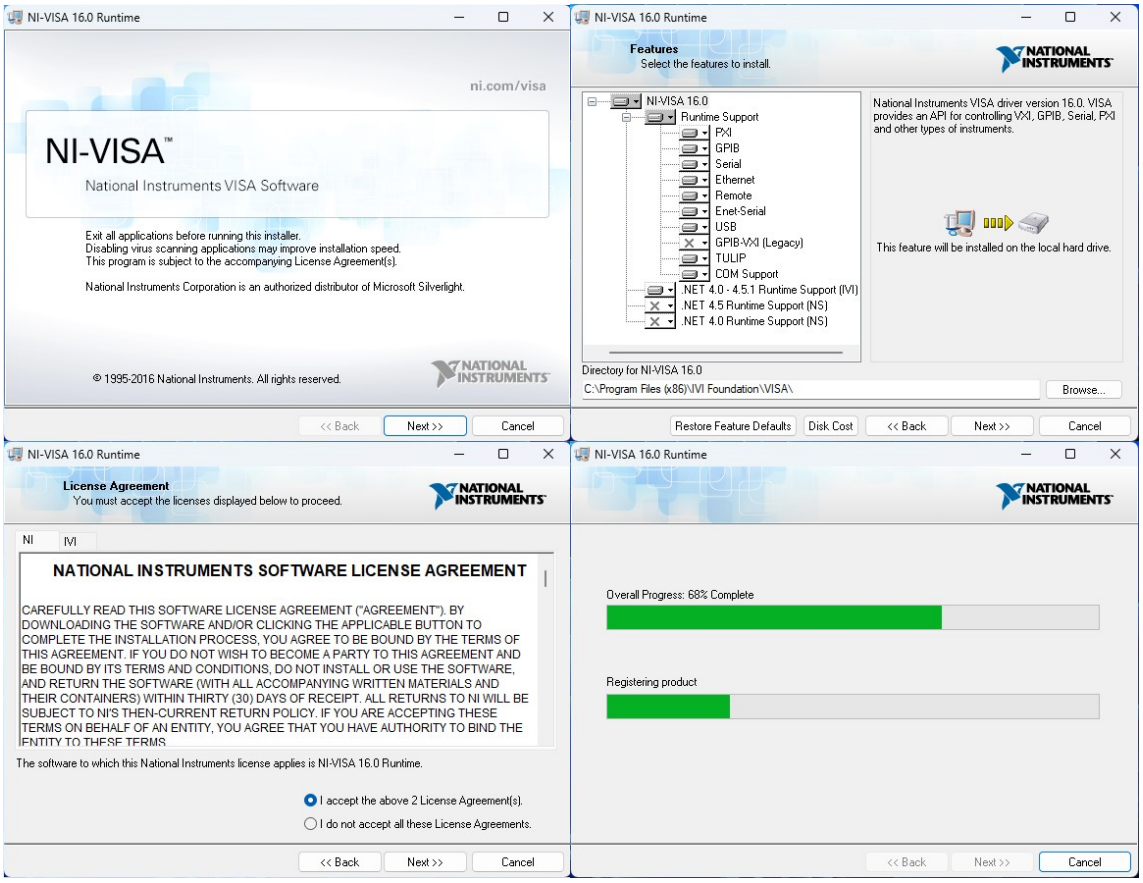
Figure 19. LabVIEW 2016 Runtime Installation completion



NI-VISA 16.00 Runtime

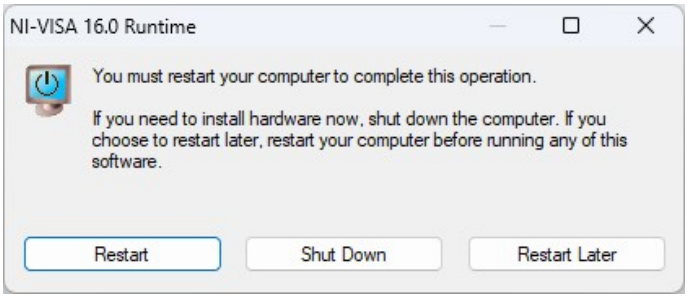
Run the NIVISA1600runtime/setup.exe utility and follow the installation wizard:

Figure 20. NI-VISA 16.00 Runtime



After installation is completed, a restart message is presented

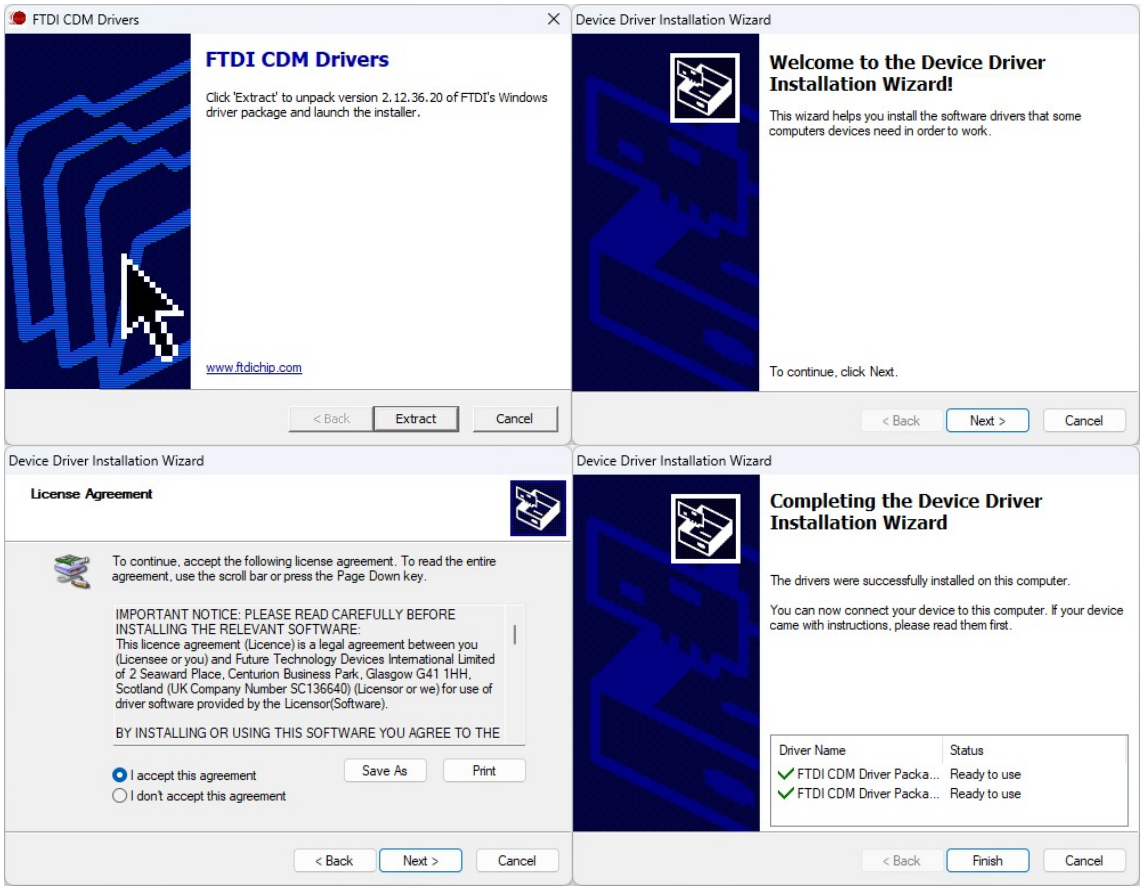
Figure 21. NI-VISA 16.00 Runtime Installation completion



FTDI Chip driver

Run the CDM2123620_Setup/CDM2123620_Setup.exe utility and follow the installation wizard:

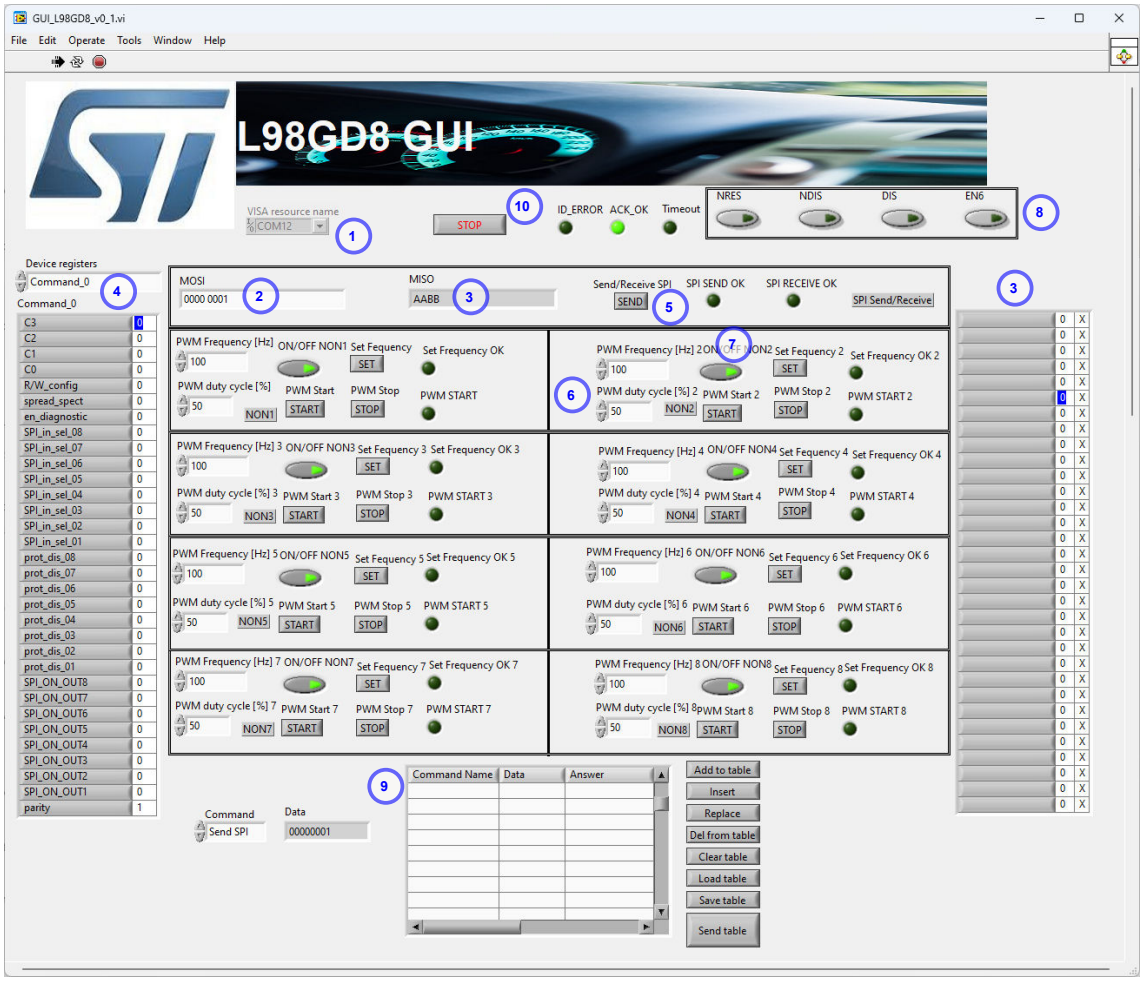
Figure 22. FTDI Chip driver



3 Graphical User Interface description

The STSW-L98GD8 GUI and its several fields are described below:

Figure 23. GUI interface description



1. **COM Port Setup**: this field allows to select the proper microcontroller board COM port.
2. **MOSI**: it allows to select the specific device register and define its required value.
3. **MISO**: this field is used to read the SDO register of the device; the response is also decoded, with detailed information for each bit.
4. **Device registers**; the drop-down list allows to select the desired command to setup for the device. After selecting a specific command, it's also possible to change value to the desired command registers.
5. **Send/Receive SPI**: pushing the **SEND** button is possible
   - to send SPI command present in MOSI field
   - to send SPI command after manually editing the MOSI field
   - to read the device answer in MISO field

   The two LEDs, **SPI SEND OK** and **SPI RECEIVE OK**, are used to check the SPI communication status (Green led means communication is OK).
6. **PWM/Steady state output control through SPI command**: it's possible to control the state of the eight output lines by setting the corresponding values of the eight input pins via the SPI interface. This approach requires a preliminary setup for the device through proper setting of internal command register bits. This approach will be detailed later.
   Additionally, a PWM feature is available and to use this feature, it is requested to setup the Frequency and the Duty cycle of the eight PWM functions (NON1…NON8) related to the eight output lines. By pushing the **START** button of a specific output section, the PWM function starts for the related output; by pushing the **STOP** button, the PWM for the related output is stopped. Before starting a PWM function for a specific output, the related **PWM Frequency** must be confirmed by pushing the corresponding **SET** button.
7. **PWM/Steady state output control through parallel command**: it's possible to control the state of the eight output lines by setting the corresponding values of the eight input pins via the direct parallel interface (GPIO mode). This approach requires a preliminary setup for the device through proper setting of internal command register bits. This approach will be detailed later.
   Additionally, a PWM feature is available and to use this feature, it is requested to setup the Frequency and the Duty cycle of the eight PWM functions (NON1…NON8) related to the eight output lines. By pushing the **START** button of a specific output section, the PWM function starts for the related output; by pushing the **STOP** button, the PWM for the related output is stopped. Before starting a PWM function for a specific output, the related **PWM Frequency** must be confirmed by pushing the corresponding **SET** button.
   By pushing the **ON/OFF** button, it is also possible to manually control the steady state of the corresponding output.
8. **NRES, NDIS, DIS and EN6 GPIO control**: with these buttons, it is possible to control the status of the related signals. To let L98GD8 properly work, **NRES** should be asserted, **NDIS** should be asserted, **DIS** should not be asserted, and **EN6** should be asserted only if you need to use OUT6 output.
9. **Command Table**: this table can be used to schedule a set of commands to be sent to L98GD8 in terms of SPI read/write. It's possible to setup a batch of up to 100 lines of commands, to save it and to reload it when needed. The commands will be deployed to the microcontroller and executed in real time; after the execution, the results will be sent back to the GUI.
10. **Stop**: this button closes the selected communication port with the device and exits the application.

4 Running procedure

Launch the application and then follow these steps:

1. Configure the COM port (FTDI) to communicate with the microcontroller board.
2. Press **OK**. If the microcontroller board is recognized, the **ACK_OK** LED becomes green.
3. Press (activate) buttons **NRES** and **NDIS**.
4. To setup the device to correctly communicate with the GUI, Communication Check (CC) feature must be disabled: send the following SPI command

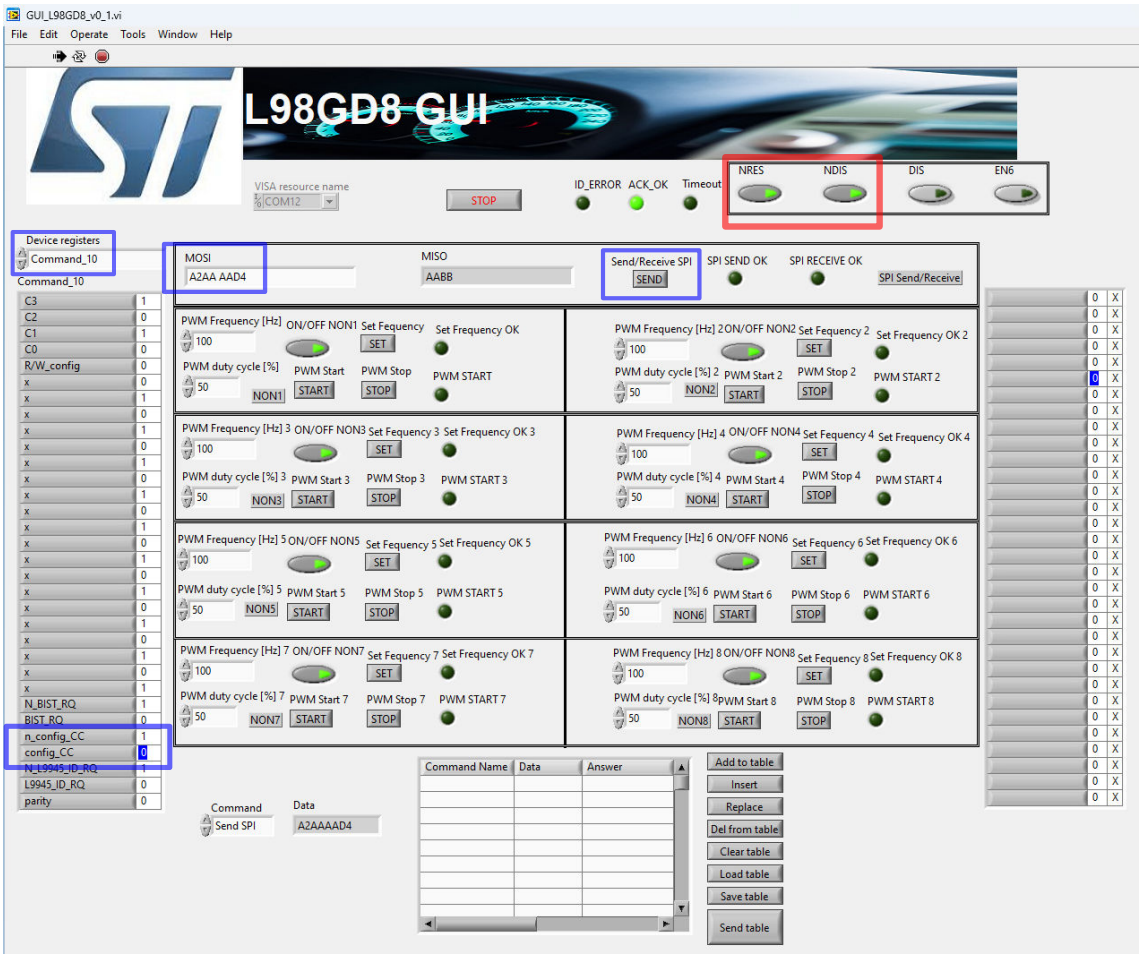
```
A2AA AAD4
```

This corresponds to set in Command_10 register the following bits:

```
n_config_cc = 1
config_CC = 0
```

leaving the other register bits at their default value, and then press **SEND** button.

Figure 24. Communication Check disable



4.1 Usage example

As an example of using the Graphical User Interface to control the EVL-L98GD8 board embedding the L98GD8 device, the procedure to turn on the output line 1 of the device in both modes, direct parallel (GPIO) and SPI, is reported below.

1. Properly setup the Output 1 and enable it: as reported in L98GD8 datasheet, this output stage is an NMOS High-side architecture. So select Command_1 and set the following command register bits

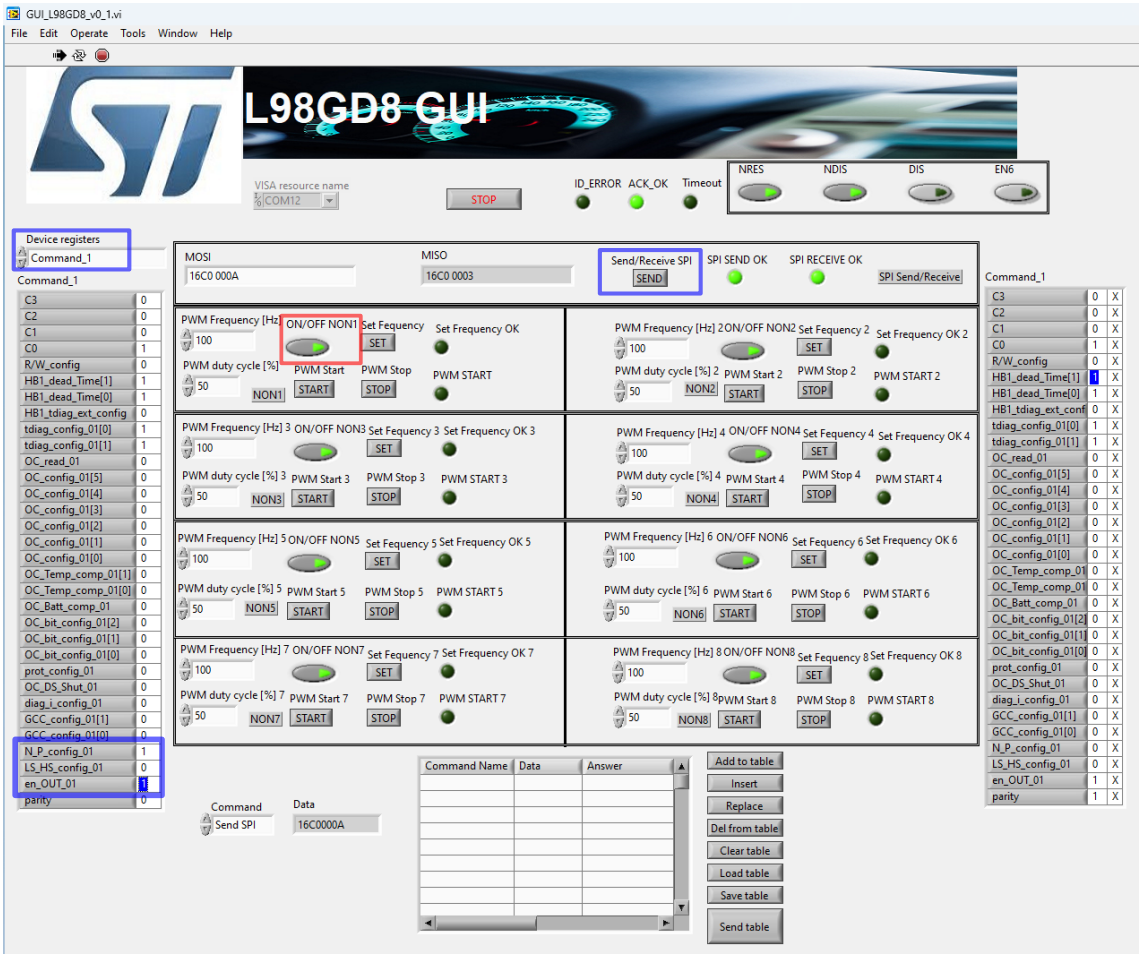
```
N_P_config_01 = 1
LS_HS_config_01 = 0
en_OUT_01 = 1
```

   and send the resulting SPI command

```
16C0 000A
```

2. Press **ON/OFF NON1** to turn on output 1 in steady state mode through the direct parallel (GPIO) interface

Figure 25. Out 1 ON through direct parallel (GPIO) interface



The same action can be performed using the SPI interface. The steps to execute in this case are the following:

1. Properly setup the Output 1 and enable it: as reported in L98GD8 datasheet, this output stage is an NMOS High-side architecture. So select Command_1 and set the following command register bits

```
N_P_config_01 = 1
LS_HS_config_01 = 0
en_OUT_01 = 1
```

   and send the resulting SPI command

```
16C0 000A
```

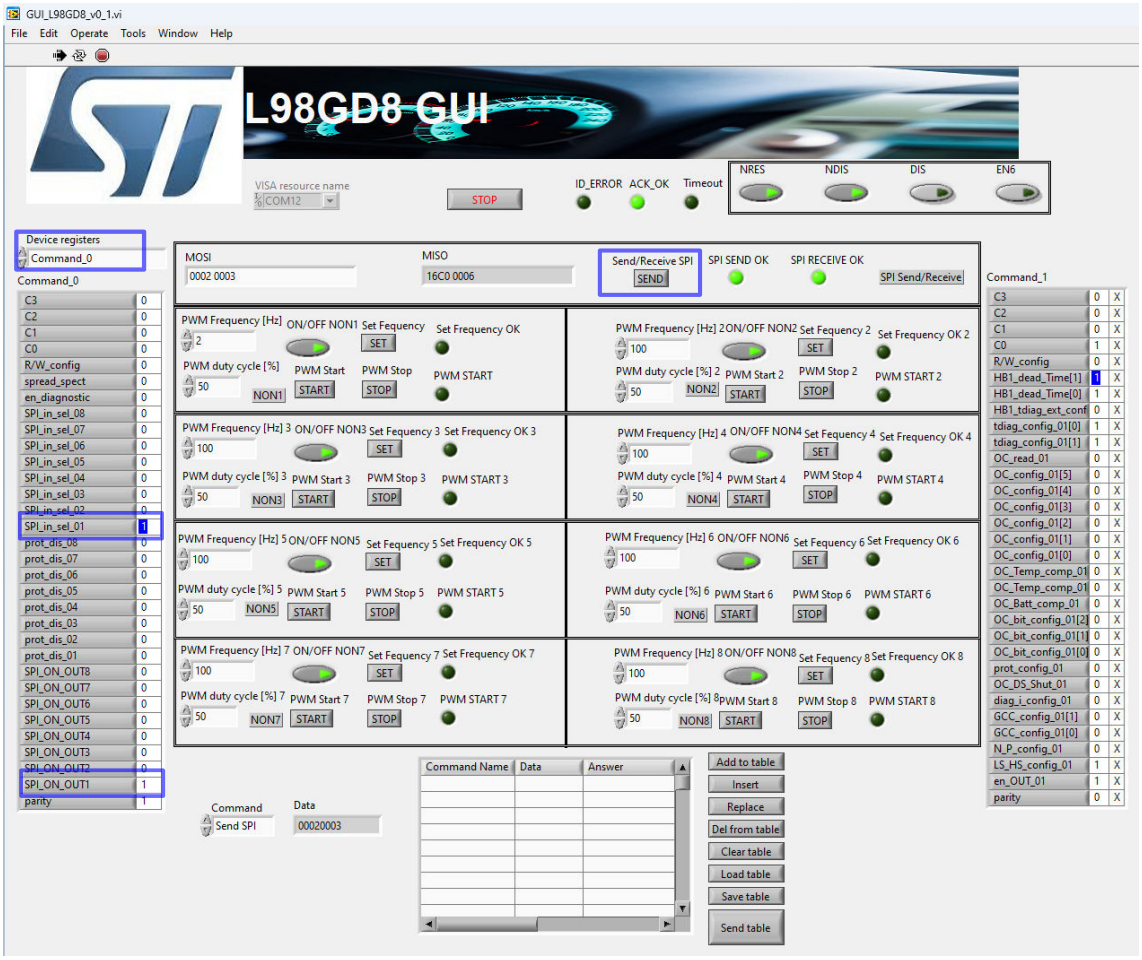
2. Enable SPI input 1 and turn ON output line 1. This is obtained by selecting Command_0 and setting the following command register bits

```
SPI_in_sel_01 = 1
SPI_ON_OUT1 = 1
```

   and send the resulting SPI command

```
0002 0003
```

Figure 26. Out 1 ON through SPI interface



In a similar way, it's possible to enable the PWM feature on output line 1. The following steps are needed to achieve the desired result:

1. Properly setup the Output 1 and enable it: as reported in L98GD8 datasheet, this output stage is an NMOS High-side architecture. So select Command_1 and set the following command register bits

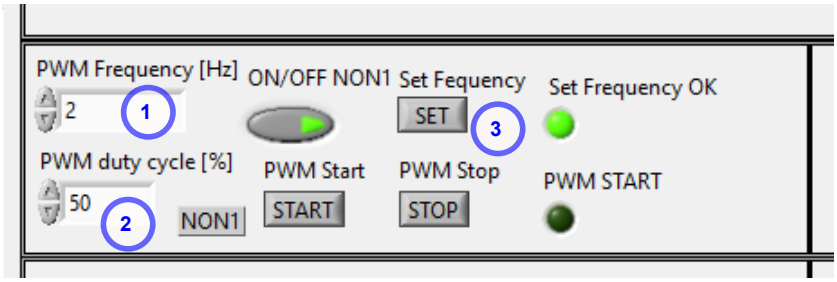
```
N_P_config_01 = 1
LS_HS_config_01 = 0
en_OUT_01 = 1
```

   and send the resulting SPI command

```
16C0 000A
```

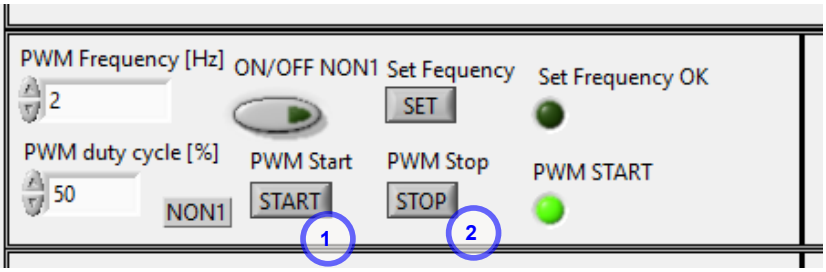
2. Press **ON/OFF NON1** to turn on output 1 in steady state mode through the direct parallel (GPIO) interface
3. Selet the desired Frequency and Duty Cycle for the PWM feature on Out 1
4. Press **SET** to activate the selected values

Figure 27. PWM setup for Out 1



5. Press now **START** to activate the PWM feature on Out 1. Press **STOP** to terminate this feature.

Figure 28. PWM ON and OFF for Out 1



Revision history

Table 1. **Document revision history**

Date	Revision	Changes
24-Jul-2025	1	Initial release.

Contents

List of figures

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

The purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of the purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.