Function description

Maintenance Firmware

netX 90

Hilscher Gesellschaft für Systemautomation mbH
www.hilscher.com

DOC190802FD01EN | Revision 1 | English | 2020-02 | Preliminary | Public

Table of contents

1 Introduction

1.1 About this document

1.1.1 Description of the contents

This document describes the functions of the maintenance firmware for the netX 90 SoC. It is intended for:

- Developers of firmware for netX 90-based devices
- OEM customers and firmware developers who want to use maintenance firmware in netX 90-based devices

1.1.2 List of revisions

Index	Date	Author	Revision
1	2020-02-05	MKE	Preliminary version created

Table 1: List of revisions

1.2 Further relevant documentation

Besides this functional description, the following documents are also relevant to the user of netX 90-based devices:

Title	Contents	Document ID
Getting started: netX Studio CDT – netX 90 development	Getting started guide for netX 90 SoC development with netX Studio CDT (for software developers)	DOC170504GSxxEN
Production guide netX 90	Describes the software architecture of the netX 90 SoC and explains how to program necessary software components into your device during end-of-line production.	DOC190101PGxxEN
Firmware Container netX 90	Describes the structure of the firmware update container.	DOC190803ANxxEN
CIFX API – Programming reference guide	Describes the CIFX application programming interface.	DOC121201PRxxEN
netX Dual-Port Memory – Packet-based services (netX 90/4000/4100)	Describes the non-cyclic packet-based DPM services of the netX 90.	DOC190301APIxxEN
Hilscher status and error codes – Firmware and driver	Lists the status and error codes of Hilscher firmware and drivers with brief descriptions.	DOC100802APIxxEN

Table 2: Additional documentation

1.3 Abbreviations

Abbreviation	Meaning
APP CPU	Application CPU on the netX 90 handling the customer's application.
APP FW	Application firmware
CLI Flasher	Command Line Interface Flasher tool
COM FW	Communication firmware
COM CPU	Communication CPU on the netX 90 handling communication (protocol stack) and basic management tasks of the SoC.
DPM	Parallel Dual-Port Memory of the netX 90 (host interface)
FDL	Flash Device Label
HW config	Hardware configuration file
MFW	Maintenance firmware
ROM code	Hard-coded software residing in the Read-only memory of the netX. Handles the netX 90 boot process and "loads" firmware into RAM (if not executed in Flash).
SPM	Serial Dual-Port Memory of the netX 90 (host interface)

Table 3: Abbreviations

2 Brief description

The maintenance firmware (MFW) is a dedicated/standalone firmware for handling the update of “regular” communication firmware (*.nxi and *.nxe) and application firmware (*.nai and *.nae) on the netX 90 SoC. Hilscher provides the maintenance firmware as ready-to-use binary file; it cannot be created or edited by the user or the OEM of the netX device.

It is programmed (“flashed”) to the internal Flash of the netX device (typically at offset 0x61000 in INTFLASH1) during its “end-of-line” production along with the other software components required by the netX.

The functions of the maintenance firmware are:

- to serve as “basic” or “recovery” firmware, which is started if the “regular” firmware is not available or fails to start
- to manage the download of new firmware and the update/replacement of existing firmware
- to install *.nxi, *.nxe, *.nai, *.nae firmware files to their defined destinations.
- to download and install other files (like e.g. configuration files) if the netX 90 is equipped with external SQI Flash with a file system (firmware use case C)

When the maintenance firmware is started, it runs in the COM CPU of the netX, allowing no other firmware to run in parallel, not even in the APP CPU.

Note:
The maintenance firmware cannot update itself. Updates of the maintenance firmware after device production – i.e. when the netX 90 device is “in the field” – are possible only by flashing a new version with the **Flasher** tool of **netX Studio CDT** or with the **Command Line Interface Flasher** (CLI Flasher).

The maintenance firmware can be recognized by its *.mxf file extension. There are two different versions:

- MFW_netX90_flash.mxf for firmware use cases A and B. In these uses cases, the update area is located in “raw” Flash in either INTFLASH1 (use case A) or external SQI Flash (use case B). The maximum size of this MFW file is 84 KByte.
- MFW_netX90_filesystem.mxf for firmware use case C. In this use case, the update area is located in the file system of the external SQI Flash. The maximum size of this MFW file is 124 KByte.

Information on how to create a Flash file system for the netX can be found in the *Production guide netX 90* (DOC190101PGxxEN), section *Creating file system in external SQI Flash for use case C*.

3 Dependencies

Firmware use cases

The maintenance firmware depends on the so-called “firmware use case”. There are three firmware use cases:

- **Use case A**: Small footprint slave device with firmware update area in “raw” INTFLASH1 (without Flash file system). Max. size of the COM FW is 500 KByte.
- **Use case B**: Small footprint slave device with firmware update area in “raw” SQI Flash (without Flash file system). Max. size of the COM FW is 880 KByte.
- **Use case C**: Full-featured loadable firmware with firmware update area in file system of SQI Flash. Max. size of the COM FW is 880 KByte.

For more detailed descriptions of the firmware use cases, see *Production guide netX 90* (DOC190101PGxxEN), section *Firmware use cases*.

Firmware use cases A and B require the MFW type:
`MFW_netX90_flash.mxf`

Firmware use case C requires MFW type:
`MFW_netX90_filesystem.mxf`

Flash layout definition in Flash Device Label

The “firmware update area” is the area in the “raw” Flash respectively in the file system directory (use case C with external SQI Flash) where a new firmware or the “update container” (a.k.a. FWUPDATE container, which is a zip archive containing one or more files) is stored after it has been downloaded. From there, the new firmware is going to be installed to its destination by the MFW during the update process.
The locations of these storage areas depend on the firmware use case. Each location is defined in the “Flash Layout Table” of the Flash Device Label (FDL). Each firmware use case requires a different FDL containing the right Flash layout definitions for its purpose.
The MFW reads the FDL before starting the update process in order to retrieve information about the storage location of the firmware or FWUPDATE container, and also to retrieve information about the destination where the new firmware shall be installed.

The FDL is programmed (flashed) into the netX device during its end-of-line production.

Hardware configuration file for maintenance firmware

Because the ROM Loader cannot use the “regular” hardware configuration file (`*.hwc`) while executing the maintenance firmware, a separate configuration file is needed to configure the device and to provide necessary hardware information for the maintenance firmware during start-up. This *hardware configuration file for maintenance firmware* can be recognized by its `*.mwc` file extension.
It basically contains the same configuration data as the “regular” `*.hwc` hardware configuration file, but it is stored in a different location in Flash, right before the maintenance firmware. It contains a cookie that allows the ROM Loader to identify the maintenance firmware as “alternative software”.

This *Hardware configuration for maintenance firmware* file is created by the OEM in **netX Studio CDT**. After having configured the hardware, the `*.mwc` file is produced automatically alongside the “regular” `*.hwc` hardware configuration file when the OEM starts to build the hardware configuration binaries. Both files will then be programmed (flashed) into the netX device during its end-of-line production.

4 How to activate the maintenance firmware

4.1 Overview

The maintenance firmware remains inactive during “normal” operation of the netX 90. The maintenance firmware will be started only if the device is reset in “alternative boot mode”. The ROM code inside the netX enters this alternative boot mode on the following events:

- After a software reset if the previously running communication firmware has enabled the mode by placing an “alternative boot mode cookie” in INTRAM beforehand (reset modes UPDATESTART or BOOTSTART initiated either via the CIFX API function `xSysdeviceResetEx` or via the DPM packet service `HIL_FIRMWARE_RESET_REQ`)
- After a reset if the `RUN` pin of the netX has been set accordingly beforehand (i.e. “pulled down” to GND)
- After a reset if no valid communication firmware image can be found or if it fails to start.

Note:
Besides “standard boot mode” and “alternative boot mode”, the ROM code also supports the so-called “console mode” that allows downloading (“flashing”) files or images to the internal Flash memory of the netX via various interfaces. If necessary, the console mode can be used to flash a new maintenance firmware to the netX (because the maintenance firmware cannot update itself).
The console mode will be automatically entered if neither valid communication firmware nor maintenance firmware are available for booting. The netX can also be forced into console mode by a software reset via mailbox packet service from the application side or by pulling the `RDY` pin of the netX to `GND` during device reset.
For more detailed information about the console mode, see section *Console mode* and section *Programming interface options* in the *Production guide netX 90* (DOC190101PGxxEN).

The following flow chart shows the boot sequence and the conditions that trigger a certain boot mode:

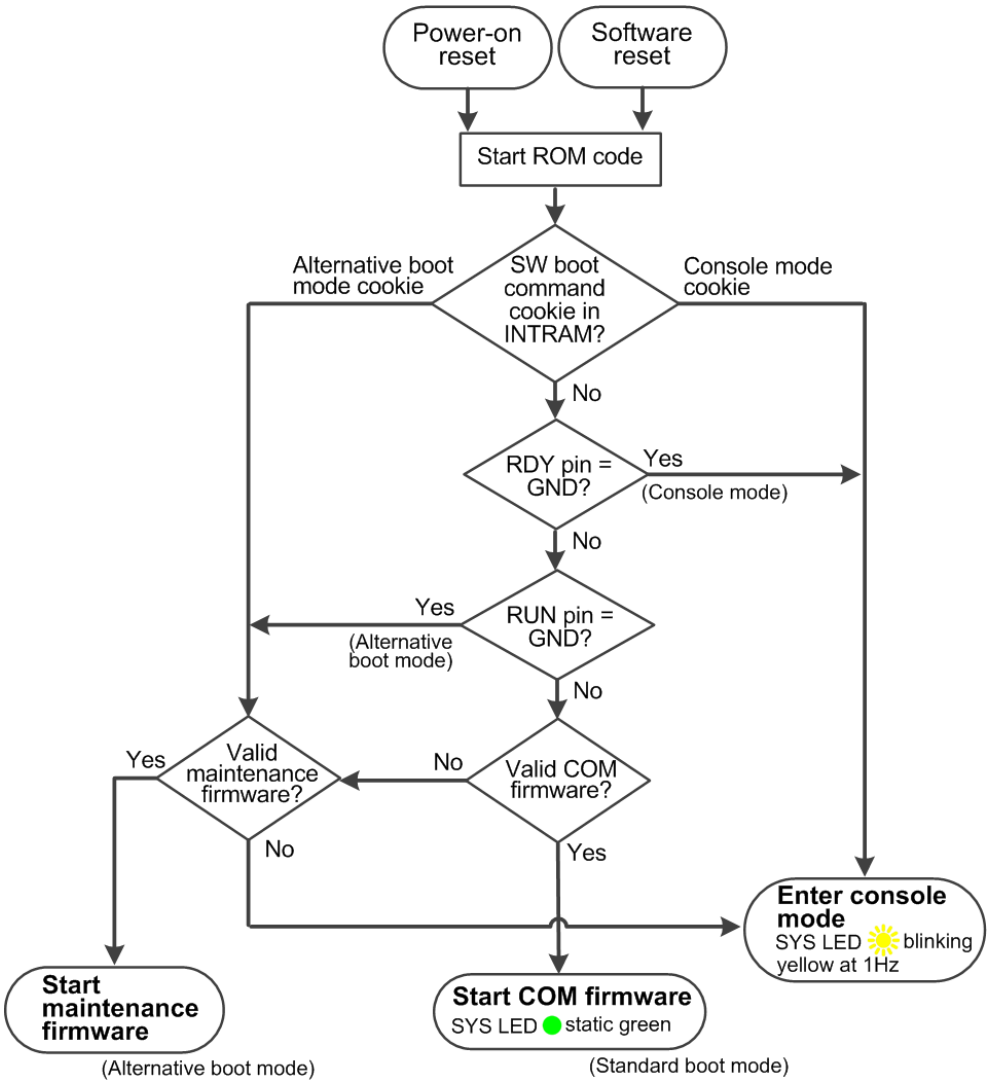

Figure 1: Boot sequence flow chart

4.2 Activation by software reset command

When the communication firmware is running, you can use the following methods to trigger a software reset to initiate alternative boot mode and start the maintenance firmware:

- By CIFX API function call `xSysdeviceResetEx` from the application, with reset modes `UPDATESTART` or `BOOTSTART` (refer to *CIFX API Programming reference guide*, DOC121201PRxxEN)
- By a packet command `HIL_FIRMWARE_RESET_REQ` via the system mailbox, with reset modes `UPDATESTART` or `BOOTSTART` (refer to *Packet API – netX Dual-Port Memory – Packet-based services (netX 90/4000/4100)*, DOC190301APIxxEN)

The behavior of the maintenance firmware after the reset can be controlled by specifying the reset mode:

- Reset mode `UPDATESTART`:
  The maintenance firmware will automatically try to install the firmware specified in the `Reset Parameter` (which are bit fields included in the `xSysdeviceResetEx` function call, respectively in the `HIL_FIRMWARE_RESET_REQ` packet).
  If the `Reset Parameter` does not specify a certain firmware, the maintenance firmware will by default try to install the first firmware it finds in the `FWUPDATE` area of the internal Flash (use cases A and B), respectively in the `VAR0` directory of the SQI Flash file system (use case C). If no valid firmware is available, the maintenance firmware indicates an error state (SYS LED is set to steady yellow) and changes into “idle mode”. The idle mode allows the user to download a valid new firmware or update container to the device, or to issue another reset request.
- Reset mode `BOOTSTART`:
  The maintenance firmware directly enters idle mode after booting and, for the time being, does not attempt to install a new firmware yet (even if a valid firmware file is available in the designated Flash areas). The SYS LED indicates the `BOOTSTART` idle mode by alternating between yellow and green at 4 Hz. The idle mode allows the user to download a valid new firmware or update container to the device, or to issue another reset request.

4.3 Activation in case of invalid COM FW after reset

The ROM Loader will automatically execute the maintenance firmware if it cannot find a valid communication firmware in “standard boot mode” after reset. The MFW thus serves as “backup” firmware if the “regular” firmware is missing or corrupted.
The MFW checks the Firmware Status Register (`only_porn`), which is a register inside the netX storing information about reset modes and parameters. Because the data in this register is volatile/non-remanent, it will be empty after a power-on reset. In this case – being unable to find instructions in this register – the MFW will by default try to install the first firmware it finds in the firmware update area (use cases A and B), respectively in the `VAR0` directory of the SQI Flash file system area (use case C).

If the MFW cannot find a firmware file or a `FWUPDATE.zip` container, it enters “idle mode” and indicates error state by setting the SYS LED to static yellow and writing `ERR_HIL_NOT_AVAILABLE` (`0xC0001152`) code to the system error field (`ulSystemError`) in the system status block of the DPM.
The idle mode allows the user to download a valid new firmware or update container to the device, or to issue another reset request.

The MFW also checks for a special cookie indicating whether a firmware update had taken place before the last software reset (a power-on-reset would have cleared the volatile cookie). If the MFW finds this cookie after a software reset, it will not start another update process, but will enter “idle mode”, sets the SYS LED to static yellow and issues an `ERR_HIL_UPDATE_ERROR` message. This is to prevent the installation of the same (possibly defective) firmware residing in the FWUPDATE area in an infinite loop, and to indicate potential problems with the firmware or the Flash.

4.4 Activation by RUN pin pulled to GND during reset

If the netX is started with its RUN pin connected to GND, the ROM code initiates the “alternative boot mode”; i.e. it directly executes the maintenance firmware instead of trying to start the communication firmware. The user can thus “manually” force the netX into starting the maintenance firmware.

Note:
Consult the hardware documentation of your netX 90 device for information on the availability of e.g. a jumper that allows you to pull the RUN pin to GND in order to enter the alternative boot mode.
If you are using the **NXHX 90-JTAG** board, see section *S400 – Slide switches for console mode and alternative boot mode* in the device description *NXHX 90-JTAG Development board* (DOC170202HWxxEN) for information on this.

In this case, the MFW by default will also try to install the first firmware it finds in the firmware update area (use cases A and B), respectively in the VAR0 directory of the FWUPDATE.zip container in the SQI Flash file system (use case C).

If the MFW cannot find a firmware or a ZIP file, it enters “idle mode” and indicates error state by setting the SYS LED to static yellow and writing code ERR_HIL_NOT_AVAILABLE (0xC0001152) to the system error (ulSystemError) field in the system status block of the DPM.
The idle mode allows the user to download a valid new firmware to the device, or to issue another reset request.

5 Firmware update procedure

5.1 Overview

The main function of the MFW is to update the firmware of netX 90 devices “in the field”.
The following firmware files can be updated:

- `*.nxi` communication firmware (a.k.a LFW) for the COM CPU
- `*.nai` application firmware for the APP CPU (in case of a netX “stand-alone-chip” device)
- `*.nxe` communication firmware extension for the COM CPU
- `*.nae` application firmware extension for the APP CPU

5.2 Firmware download

Except for rare cases in which the netX device is delivered with alternative firmware versions already stored in the FWUPDATE area of its Flash device, the update process requires a firmware download before the actual installation process can be started.

The download of the “new” firmware version to the netX can be managed either by the maintenance firmware or by the communication firmware (except for firmware use case B, in which the COM firmware has no access to the FWUPDATE area in SQI Flash).
On receiving the download request via CIFX API (`xSysdeviceDownload` function call), the running firmware will automatically download and store the new firmware to the designated FWUPDATE area (which is defined in the FDL of the device).

Firmware use cases A and B (download to raw Flash)

In use cases A and B, the FWUPDATE area is located in “raw” (without file system) internal Flash (use case A) or in raw external SQI Flash (use case B).
Only one file can reside in this area at a time, so each new download will overwrite every currently residing file.
Accepted file types are `*.nxi` (communication firmware) or `*.nai` (application firmware) or `FWUPDATE.zip` (firmware update container).
Files and ZIP container must comply with the 8.3 file name convention.
Using an `FWUPDATE.zip` container is required if the size of the new firmware file exceeds the size of the FWUPDATE area (380 KByte). In this case, the firmware file must be zipped before download.
Using an `FWUPDATE.zip` container is also required if `*.nxe` or `*.nae` firmware extension files shall be used on the netX. In this case, the firmware file must be zipped together with the corresponding extension file into a single container before download.

Firmware use case C (download to file system)

In use case C, all software components that are intended for the update process (i.e. `*.nxi` or `*.nai` or `FWUPDATE.zip`) are stored in the `/FWUPDATE` directory of the file system in the external SQI Flash. Files and ZIP container must comply with the 8.3 file name convention.

If one of these files is received via download request on `PORT_0` (`ulChannelNo` set to `0`), the file is automatically transferred to the `/FWUPDATE` directory (it thus will not be visible on `/PORT_0`). Other files residing in the `/FWUPDATE` directory will be deleted before the download of the new file starts.

Use case C allows you to store multiple firmware variants within a single `FWUPDATE.zip` container. The variant to be installed can be specified in the reset parameter for boot mode `UPDATESTART` in the `xSysdeviceResetEx` CIFX API function call or in the `HIL_FIRMWARE_RESET_REQ` packet.

For more information about the firmware update container, see application note *Firmware Container netX 90*, DOC190803ANxxEN.

Note:
The contents of the firmware update areas in raw Flash (use cases A and B) and in the SQI file system (use case C) will not be deleted during or after firmware installation (only a new download process will delete them).
This ensures that the same firmware can be installed again without additional download process if – for some reason – the formerly installed “original” firmware has been destroyed.

5.3 Firmware installation

Start sequence

The actual installation of the “new” firmware that has been downloaded to the FWUPDATE area/directory can be started by a software reset command featuring the `UPDATESTART` reset mode. If you have downloaded an `FWUPDATE.zip` container containing more than one firmware variant (only possible in firmware use case C), you can specify the variant to be installed in the `Reset Parameter` of the command.
The `UPDATESTART` reset mode command can be issued via CIFX API either to the running communication firmware or to the maintenance firmware (running in “idle mode”) by:

- CIFX API function call `xSysdeviceResetEx` from the application (for details, refer to *cifX API Programming reference guide*, DOC121201PRxxEN)
  or
- packet command `HIL_FIRMWARE_RESET_REQ` via the system mailbox (for details, refer to *Packet API – netX Dual-Port Memory – Packet-based services (netX 90/4000/4100)*, DOC190301APIxxEN)

The command causes the firmware to place an “Alternative boot mode cookie” in INTRAM and to write the reset mode and the reset parameter to the “Firmware status register” before reset. After reset, the ROM code finds the “Alternative boot mode cookie” in INTRAM and starts the maintenance firmware with the purpose of installing the new firmware.

Preparation sequence

The MFW clears the “alternative boot mode cookie” in INTRAM (in order to enable standard boot mode [i.e. starting of the COM FW] for the next reset) and checks for a certain cookie in the “Firmware status register” (`only_porn`), indicating whether an installation process had already taken place before the last reset. If it finds this cookie, the MFW will not start the installation process, but changes into “idle mode” instead, issues an `ERR_HIL_UPDATE_ERROR` error and sets the SYS LED to steady yellow (indicating error state). This is to prevent the installation of the same (possibly defective) firmware in an infinite loop. If there is no such cookie, the MFW checks the `Reset Mode` and the `Reset Parameter` in the “Firmware status register”. In case of reset mode `UPDATESTART`, it starts to prepare the installation of the firmware variant specified in the `Reset Parameter`. If the `Reset Parameter` does not specify a variant other than `0`, the MFW starts to prepare the installation of the default firmware, i.e. of the firmware it finds in the `VAR0` file system (use case C) respectively in the `FWUPDATE` area in “raw” Flash.

During the preparation, the MFW performs the following checks:

- it checks boot header and file header
- it checks compatibility of hardware and firmware
- it checks availability of the destination areas for the firmware
- it checks whether new the firmware is identical to the currently installed firmware (which would render an update unnecessary)
- it checks for `*.nxe` or `*.nae` firmware extension files and, if found, checks for the availability of the corresponding `*.nxi` or `*.nai` files

Installation sequence

The installation is performed sequentially from first to last block. If installation fails (if an incomplete firmware file is written to the Flash), the ROM loader will not start this invalid firmware after the next reset.

The installation process is repeated for each firmware file found during the preparation sequence (e.g. `*.nxe` extension file). If no more firmware files can be found, the MFW starts the installation of configuration files or other files.

Note:
The installation of configuration or regular files is only supported on a file system (use case C) using the `FWUPDATE.zip` container.
The mandatory layout of the directory of the container is described in the application note *Firmware Container netX 90*, DOC190803ANxxEN.
The configuration files will be extracted from the `FWUPDATE.zip` container to the corresponding directories in the file system. If a file with the same name already exists on the file system, its size and CRC are compared to the corresponding values of the file in the container. It will be installed only if size or CRC checksums are different. Firmware files (`*.nxi`, `*.nxe`, `*.nai` and `*.nae`) will not be extracted to the file system (even if they were placed into `PORT_X` directories).
Note also that configuration files are installed only for the COM CPU.

Verification sequence

After installation, the MFW reads back all installed firmware files and checks them for validity. It checks:

- all installed files (`*.nxi`, `*.nxe`, `*.nai` and `*.nae`)
- boot header and file header (including CRCs)
- MD5 checksum (by reading the complete file and calculating it)

Finishing sequence

If the verification has been successful, the MFW places the “installation done cookie” into the “Firmware status register” (`only_porn`) and initiates a software reset in “standard boot mode” (i.e. without “alternative boot mode cookie” in INTRAM), whereby the newly installed firmware will be started by the ROM loader.

If the verification has failed, the MFW sets an error code in the `ulSystemError` field in the system status block of the DPM, sets the SYS LED to static yellow and enters “idle mode”. During “idle mode” the “Marshaller” and the file packet handling are enabled again (which were disabled during UPDATESTART mode), in order to allow the download of new firmware.

Note:
The contents of the firmware update areas in raw Flash (use cases A and B) and in the SQI file system (use case C) will not be deleted during or after firmware installation (only a new download process will delete them).
This ensures that the same firmware can be installed again without additional download process if – for some reason – the formerly installed “original” firmware has been destroyed.

Flow chart of firmware installation process

The following flow chart shows a summary of the behavior of the MFW after starting:

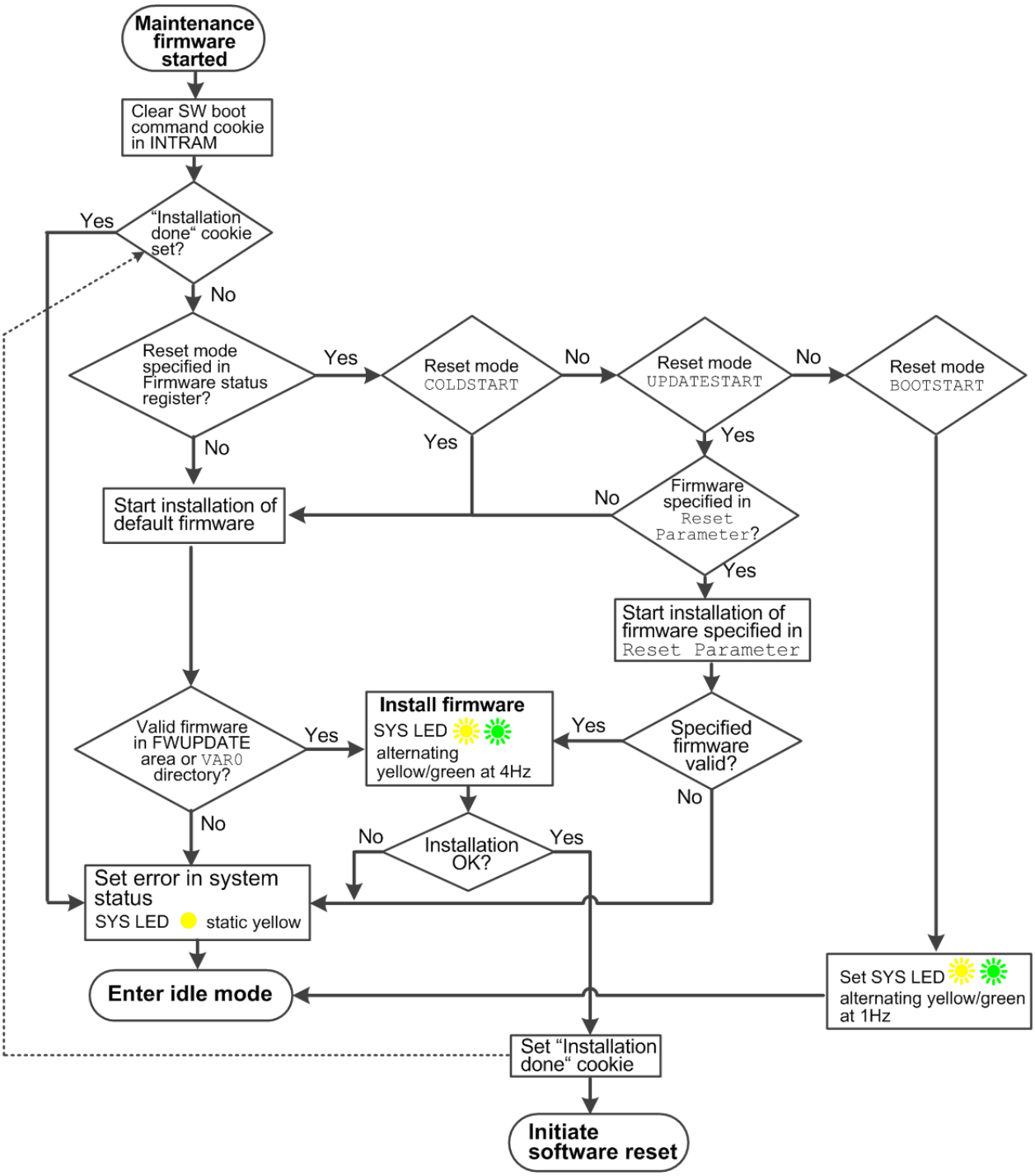

Figure 2: Maintenance firmware after startup

6 How to communicate with the maintenance firmware

6.1 Supported interfaces

Maintenance and communication firmware can be accessed via the following netX 90 interfaces:

- Serial/UART interface (if implemented on the device by the OEM and if `UART Diagnostic Interface` tag in communication firmware was enabled by the developer)
- Dual Port Memory (DPM)
  - If netX is used as “companion chip”: via external DPM from the host CPU
  - If netX is used as “stand-alone-chip”: via internal DPM (iDPM) from the APP CPU

Note:
Access via Ethernet interface is not supported.

6.2 Supported methods and requests

The following methods and requests can be used to communicate with the maintenance firmware:

Function calls via CIFX API

The most important function calls are:

- file download request (in order to download new firmware to the netX): `xSysdeviceDownload`
- firmware reset request (software reset in order to start update process): `xSysdeviceResetEx`

For details, see document *CIFX API – Programming reference guide* (DOC121201PRxxEN).

Packet commands via system device mailbox

The most important packets are:

- Firmware reset request (software reset in order to start update process): `HIL_FIRMWARE_RESET_REQ`
- Error log request (to read error log of the system channel): `HIL_SYSTEM_ERRORLOG_REQ`

For details, see document *Packet API – netX Dual-Port Memory – Packet-based services (netX 90/4000/4100)* (DOC190301APIxxEN).

The maintenance firmware also supports the following file-related packets:

Service	Request	Firmware use cases A and B ("raw" Flash)	Firmware use case C (file system in SQI Flash)	For details, see Packet API manual
File download	HIL_FILE_DOWNLOAD_REQ	Download to firmware update area. Supported file types: *.nxi, *.nai and FWUPDATE.zip Overwrites current file without warning.	Download to file system. Supported file types: *.nxi, *.nai, FWUPDATE.zip and non-firmware files. **Note**: If downloaded on PORT_0, *.nxi, *.nai and FWUPDATE.zip files are automatically moved to the dedicated FWUPDATE directory. Old files are removed first.	Section *Download / Uploading Files*
File download data	HIL_FILE_DOWNLOAD_DATA_REQ			
File download abort	HIL_FILE_DOWNLOAD_ABORT_REQ			
File upload	HIL_FILE_UPLOAD_REQ	Upload of currently installed firmware only. Supported file types: *.nxi, *.nai, *.nxe, and *.nae	All file types supported.	Section *Uploading files from netX*
File upload data	HIL_FILE_UPLOAD_DATA_REQ			
File upload abort	HIL_FILE_UPLOAD_ABORT_REQ			
Format the default partition	HIL_FORMAT_REQ	Not supported.	Formats the default file system volume and creates file system.	Section *Format the Default Partition*
Read MD5 checksum from file	HIL_FILE_GET_MD5_REQ	Supported for currently installed firmware files. Supported file types: *.nxi, *.nai, *.nxe, and *.nae	All file types supported.	Section *Read MD5 File Checksum*
Read MD5 checksum from file header	HIL_FILE_GET_HEADER_MD5_REQ	Supported for currently installed firmware files. Supported file types: *.nxi, *.nai, *.nxe, and *.nae	Supported for files with Hilscher File Header V3	Section *Read MD5 File Checksum from File Header*
Delete file	HIL_FILE_DELETE_REQ	Not supported.	All files in file system except currently installed firmware files (*.nxi, *.nai, *.nxe, and *.nae).	Section *Delete a File*
Rename file	HIL_FILE_RENAME_REQ	Not supported.	All files in file system except currently installed firmware files (*.nxi, *.nai, *.nxe, and *.nae).	Section *Rename a File*
List directory	HIL_DIR_LIST_REQ	Supported for currently installed firmware files (*.nxi, *.nai, *.nxe, and *.nae). **Note**: Installed firmware files are shown on PORT_0 only. Default name FIRMWARE.* will be returned.	All files in file system. **Note**: Installed firmware files are shown on PORT_0 only.	Section *List Directories and Files from File System*

Table 4: File-related packets supported by the firmware

6.3 Supported tools

If you are not using your own tools or mechanisms in your application (based on the CIFX API) to perform a firmware update on the netX, you can use the **netHOST Device Test Application** for downloading and updating firmware on the netX via serial/UART interface or DPM.

7 How to update firmware with the netHOST application

This section provides step-by-step instructions on how to use the **netHOST Device Test Application** to:

1. Download a new firmware file from your PC to the netX via serial/UART interface (in case of a “stand-alone-chip” solution) or via Dual-Port Memory (e.g. PCI interface in case of a CIFX PC card)
2. Install the downloaded firmware by resetting the device and starting the maintenance firmware in UPDATESTART mode

Requirements

- In case of a “stand-alone-chip” solution: Your netX 90 device is equipped with a serial/UART interface and is connected to your PC via this serial/UART interface.
- In case you are using a CIFX PC Card: You have installed the CIFX PC Card and the cifX Device Driver on your PC.
- netHOST Device Test Application (version 1.4 or higher)

Note:
The netHOST application is available on the **NXDIAG Product CD** (netX Diagnostic and Remote Access xxxx-xx-x.zip), which can be downloaded from Hilscher under https://kb.hilscher.com/x/VI0YAg. After download, unpack the ZIP file, then open folder `Windows Executable` > `netHost` > `x64`. Double-click the `netHost.exe` to start the application. Note that the netHost application is “portable” and does not need to be installed on your PC.

- The `UART Diagnostic Interface` tag of the communication firmware was enabled by the developer.
- Communication firmware is running (SYS LED shows steady green) or netX is in “alternative boot mode” and maintenance firmware is running.
- New firmware
  - communication firmware file (`*.nxi`) or
  - application firmware file (`*.nai`) or
  - FWUPDATE.zip container

Note:
Files and ZIP container must comply with the 8.3 file name convention.

Step-by-step instructions

1. Establish connection from netHOST to netX device.

   ➢ Open the **netHOST Device Test Application**.

   ➢ In the **Device** menu, select the driver:
   If you are connected via serial/UART interface, choose **Select netX Driver**.
   If you are connected via Dual-Port Memory (e.g. PCI interface of a CIFX PC card), choose **Select CIFX Driver**.

   ➢ In the **Device** menu, choose **Open**.

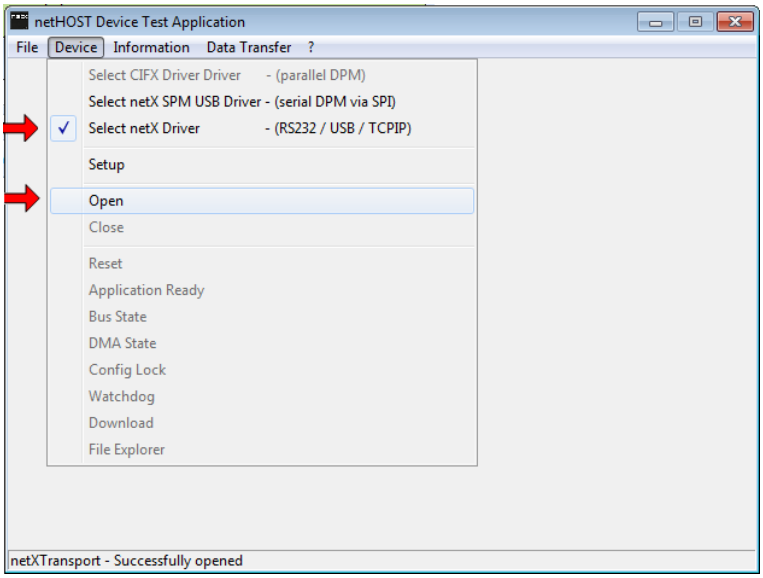

Figure 3: Open connection to netX in netHOST

   ➢ After a few seconds, the **Channel Selection** dialog opens:

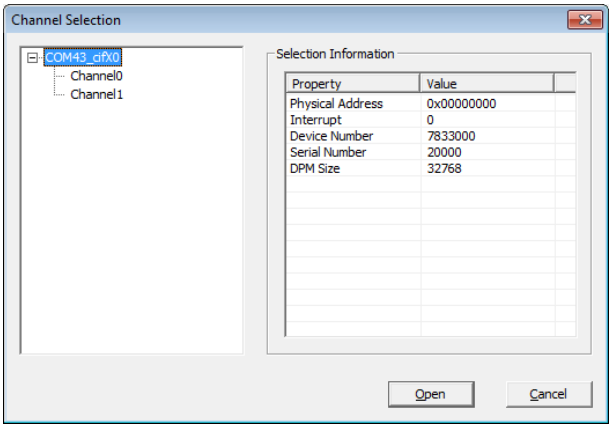

Figure 4: Channel Selection in netHOST

   ➢ Select **COMxx_cifX0** root entry (which is the “system channel”), then click **Open** button.

   ➢ The **Channel Selection** dialog closes.

2. Download new firmware to the netX device.
   - ➢ In the **Device** menu, select **Download**.
   - ➢ The **Download Test** dialog window opens.

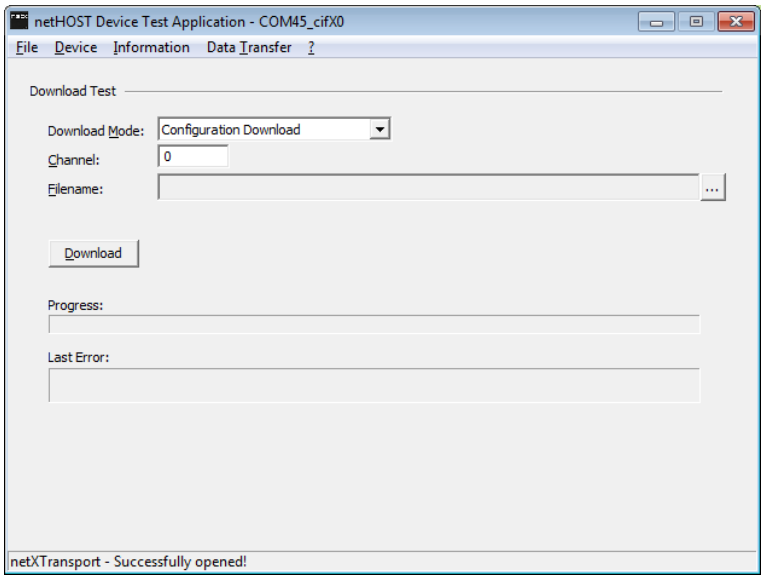

Figure 5: Download window

   - ➢ In the **Download Mode** drop-down list, select **Firmware Download**.
   - ➢ In the **Channel** field, keep the preset `0` value.
   - ➢ Select ... button to open the File selection dialog.

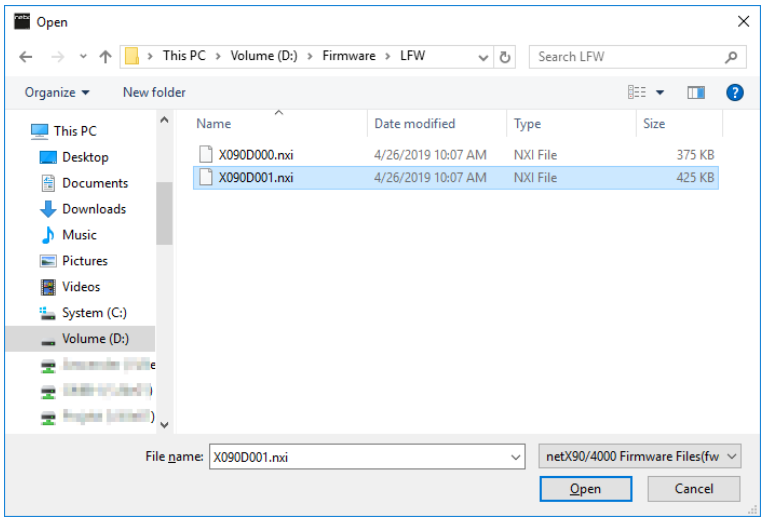

Figure 6: File selection dialog

   - ➢ In the drop-down menu in the right corner of the footer, select **netX90/4000 Firmware Files** or **All Files (*.*)** option.
   - ➢ Navigate to the directory where you have stored the firmware and select the file that you want to download. This must be a
     `fwupdate.zip` firmware update container or
     `*.nxi` communication firmware file or
     `*.nai` application firmware file.

Note:
Firmware extension files for the communication or the application side (`*.nxe` or `*.nae` files) can be downloaded if they are stored in a `fwupdate.zip` container (together with the corresponding `*.nxi` or `*.nai` file).

➢ Click **Open** button.

➢ The File selection dialog window closes. Back in the **Download** screen, the path to the selected file or container is shown in the **Filename** field:

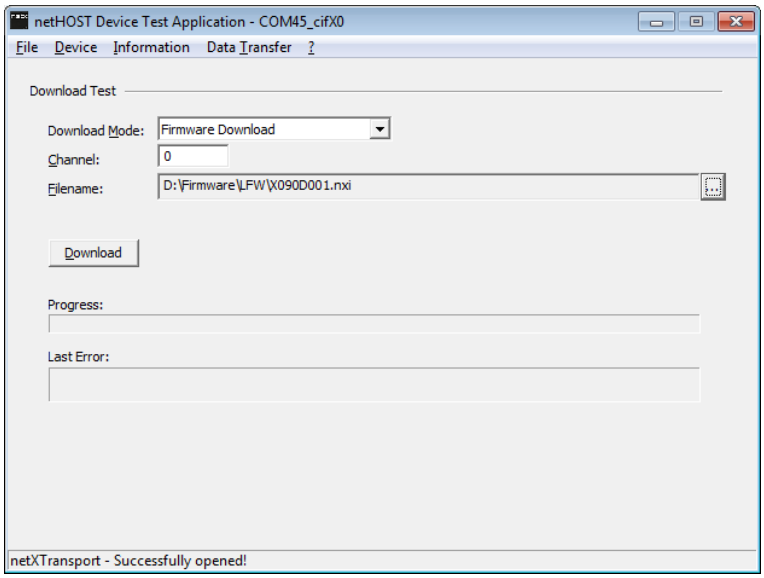

Figure 7: Ready for download

➢ Click **Download** button.

➢ While the firmware is being downloaded, a progress bar is displayed.

Note:
A full progress bar indicates the completion of the download (there will be no extra message box popping up in order to inform you about the completion of the download).

3. Reset device and install new firmware.

   ➢ After downloading has been completed, open **Device** menu and select **Reset**.

   ➢ The **Device Reset Test** dialog window opens:

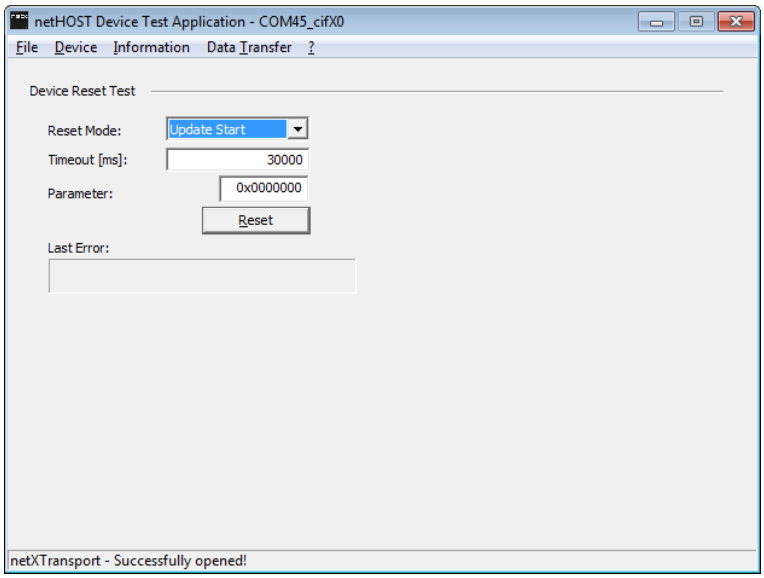

Figure 8: Reset dialog

Mode/Parameter	Value	Meaning
Reset Mode	System Start	Resets the netX and boots the installed “regular” communication firmware.
	Boot Start	Resets the netX and boots the maintenance firmware. The maintenance firmware will remain in “idle mode” after booting and will not automatically attempt to install a new firmware (even if available).
	Update Start	Resets the netX and boots the maintenance firmware. The maintenance firmware will then automatically attempt to install the first firmware file or container stored in the FWUPDATE area of the Flash (use cases A and B), respectively the firmware variant stored in the SQI file system directory (use case C) indicated in the reset parameter of the “update start“ mode (see **Parameter** below). If no specific firmware variant has been specified in the reset parameter, the MFW will try to install the default firmware (`VAR0`). If the specified firmware variant or the default firmware are invalid, the MFW will enter “Idle mode”, set the error in system status and set the SYS LED to steady yellow.
Timeout [ms]	Default: `30000`	Timeout in ms to wait for reset to complete (including firmware update process) before a timeout message is shown in the **Last Error** field.

Mode/Parameter	Value	Meaning			
Parameter	Hex value with seven digits. Only the last two digits are evaluated: 0x00000**YX**	The last digit [**X**] specifies which firmware variant of the update container shall be installed (`0`...`F`) in Reset Mode `Update Start`. The second last digit [**Y**] controls the deletion of the remanent data (`0` = do not delete; `1` = delete) in Reset Modes `Boot Start` and `Update Start`.			
		Reset Mode	**Digit**	**Value/Meaning**	
		System Start	**X**	Parameter is not evaluated (default `0x0000000`)	
			Y		
		Boot Start	**X**	Digit is not evaluated (default `0`)	
			Y	Specifies the deletion of remanent data:	
				Y = `0`	Remanent data will not be deleted
				Y = `1`	Complete remanent data area will be deleted during the startup of the MFW
		Update Start	**X**	Specifies the installation of a certain firmware variant:	
				X = `0`	Default firmware variant `VAR0` will be installed
				X = `1`	Firmware variant `VAR1` will be installed
				…	…
				X = `F`	Firmware variant `VAR15` will be installed
			Y	Specifies the deletion of remanent data:	
				Y = `0`	Remanent data will not be deleted
				Y = `1`	Complete remanent data area will be deleted after successful firmware update
Last Error	Displays error codes, status codes and brief description. See document *Hilscher status and error codes – Firmware and driver*, DOC100802APIxxEN.				

Table 5: Modes, parameters and values in Device Reset dialog window

- In the **Reset Mode** drop-down list, select **Update Start**.
- Keep the preset **Timeout** default value.
- In the **Parameter** field, you can specify a certain firmware variant to be installed after reset (only in firmware use case C and only if you have downloaded a `FWUPDATE.zip` firmware update container containing more than one variant):
  `0x0000000` = default/variant 0 `(VAR0)`
  `0x0000001` = variant 1 `(VAR1`
  `0x0000002` = variant 2 `(VAR2)`
  `...`
  `0x0000000F` = variant 15 `(VAR15)`
- Click **Reset** button.
- ⇨ The maintenance firmware is started and installs the new firmware. This is indicated by the SYS LED alternating between yellow and green at 4 Hz.
  After the installation process is finished (this takes a few seconds), the MFW automatically resets the netX again and causes the ROM loader to start the new firmware. A steady green SYS LED indicates that the new firmware is running properly.

Note:

The contents of the firmware update areas in raw Flash (use cases A and B) and in the SQI file system (use case C) will not be deleted during or after firmware installation (only a new download process will delete them).
This ensures that the same firmware can be installed again without additional download process if – for some reason – the formerly installed “original” firmware has been destroyed.

8 Flowchart

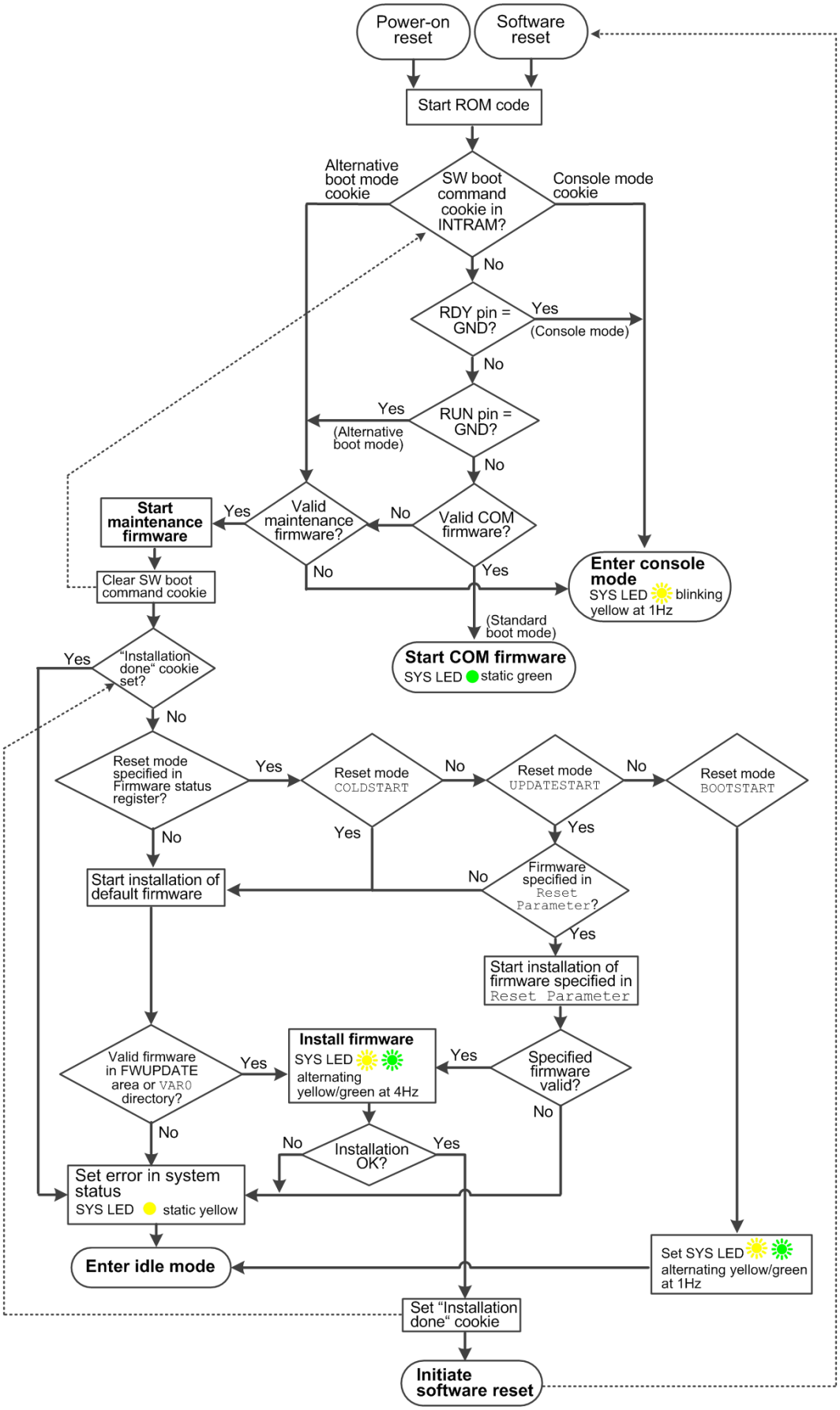

Figure 9: MFW flowchart complete

9 Legal notes

Copyright

Important notes

Utmost care was/is given in the preparation of the documentation at hand consisting of a user's manual, operating manual and any other document type and accompanying texts. However, errors cannot be ruled out. Therefore, we cannot assume any guarantee or legal responsibility for erroneous information or liability of any kind. You are hereby made aware that descriptions found in the user's manual, the accompanying texts and the documentation neither represent a guarantee nor any indication on proper use as stipulated in the agreement or a promised attribute. It cannot be ruled out that the user's manual, the accompanying texts and the documentation do not completely match the described attributes, standards or any other data for the delivered product. A warranty or guarantee with respect to the correctness or accuracy of the information is not assumed.

We reserve the right to modify our products and the specifications for such as well as the corresponding documentation in the form of a user's manual, operating manual and/or any other document types and accompanying texts at any time and without notice without being required to notify of said modification. Changes shall be taken into account in future manuals and do not represent an obligation of any kind, in particular there shall be no right to have delivered documents revised. The manual delivered with the product shall apply.

Under no circumstances shall Hilscher Gesellschaft für Systemautomation mbH be liable for direct, indirect, ancillary or subsequent damage, or for any loss of income, which may arise after use of the information contained herein.

Liability disclaimer

The hardware and/or software was created and tested by Hilscher Gesellschaft für Systemautomation mbH with utmost care and is made available as is. No warranty can be assumed for the performance or flawlessness of the hardware and/or software under all application conditions and scenarios and the work results achieved by the user when using the hardware and/or software. Liability for any damage that may have occurred as a result of using the hardware and/or software or the corresponding documents shall be limited to an event involving willful intent or a grossly negligent violation of a fundamental contractual obligation. However, the right to assert damages due to a violation of a fundamental contractual obligation shall be limited to contract-typical foreseeable damage.

It is hereby expressly agreed upon in particular that any use or utilization of the hardware and/or software in connection with

- Flight control systems in aviation and aerospace;
- Nuclear fission processes in nuclear power plants;
- Medical devices used for life support and
- Vehicle control systems used in passenger transport

shall be excluded. Use of the hardware and/or software in any of the following areas is strictly prohibited:

- For military purposes or in weaponry;
- For designing, engineering, maintaining or operating nuclear systems;
- In flight safety systems, aviation and flight telecommunications systems;
- In life-support systems;
- In systems in which any malfunction in the hardware and/or software may result in physical injuries or fatalities.

You are hereby made aware that the hardware and/or software was not created for use in hazardous environments, which require fail-safe control mechanisms. Use of the hardware and/or software in this kind of environment shall be at your own risk; any liability for damage or loss due to impermissible use shall be excluded.

Warranty

Hilscher Gesellschaft für Systemautomation mbH hereby guarantees that the software shall run without errors in accordance with the requirements listed in the specifications and that there were no defects on the date of acceptance. The warranty period shall be 12 months commencing as of the date of acceptance or purchase (with express declaration or implied, by customer's conclusive behavior, e.g. putting into operation permanently).

The warranty obligation for equipment (hardware) we produce is 36 months, calculated as of the date of delivery ex works. The aforementioned provisions shall not apply if longer warranty periods are mandatory by law pursuant to Section 438 (1.2) BGB, Section 479 (1) BGB and Section 634a (1) BGB [Bürgerliches Gesetzbuch; German Civil Code] If, despite of all due care taken, the delivered product should have a defect, which already existed at the time of the transfer of risk, it shall be at our discretion to either repair the product or to deliver a replacement product, subject to timely notification of defect.

The warranty obligation shall not apply if the notification of defect is not asserted promptly, if the purchaser or third party has tampered with the products, if the defect is the result of natural wear, was caused by unfavorable operating conditions or is due to violations against our operating regulations or against rules of good electrical engineering practice, or if our request to return the defective object is not promptly complied with.

Costs of support, maintenance, customization and product care

Please be advised that any subsequent improvement shall only be free of charge if a defect is found. Any form of technical support, maintenance and customization is not a warranty service, but instead shall be charged extra.

Additional guarantees

Although the hardware and software was developed and tested in-depth with greatest care, Hilscher Gesellschaft für Systemautomation mbH shall not assume any guarantee for the suitability thereof for any purpose that was not confirmed in writing. No guarantee can be granted whereby the hardware and software satisfies your requirements, or the use of the hardware and/or software is uninterruptable or the hardware and/or software is fault-free.

It cannot be guaranteed that patents and/or ownership privileges have not been infringed upon or violated or that the products are free from third-party influence. No additional guarantees or promises shall be made as to whether the product is market current, free from deficiency in title, or can be integrated or is usable for specific purposes, unless such guarantees or promises are required under existing law and cannot be restricted.

Confidentiality

The customer hereby expressly acknowledges that this document contains trade secrets, information protected by copyright and other patent and ownership privileges as well as any related rights of Hilscher Gesellschaft für Systemautomation mbH. The customer agrees to treat as confidential all of the information made available to customer by Hilscher Gesellschaft für Systemautomation mbH and rights, which were disclosed by Hilscher Gesellschaft für Systemautomation mbH and that were made accessible as well as the terms and conditions of this agreement itself.

The parties hereby agree to one another that the information that each party receives from the other party respectively is and shall remain the intellectual property of said other party, unless provided for otherwise in a contractual agreement.

The customer must not allow any third party to become knowledgeable of this expertise and shall only provide knowledge thereof to authorized users as appropriate and necessary. Companies associated with the customer shall not be deemed third parties. The customer must obligate authorized users to confidentiality. The customer should only use the confidential information in connection with the performances specified in this agreement.

The customer must not use this confidential information to his own advantage or for his own purposes or rather to the advantage or for the purpose of a third party, nor must it be used for commercial purposes and this confidential information must only be used to the extent provided for in this agreement or otherwise to the extent as expressly authorized by the disclosing party in written form. The customer has the right, subject to the obligation to confidentiality, to disclose the terms and conditions of this agreement directly to his legal and financial consultants as would be required for the customer's normal business operation.

Export provisions

The delivered product (including technical data) is subject to the legal export and/or import laws as well as any associated regulations of various countries, especially such laws applicable in Germany and in the United States. The products / hardware / software must not be exported into such countries for which export is prohibited under US American export control laws and its supplementary provisions. You hereby agree to strictly follow the regulations and to yourself be responsible for observing them. You are hereby made aware that you may be required to obtain governmental approval to export, reexport or import the product.

List of figures

List of tables

Contacts

HEADQUARTERS

Germany

Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-mail: info@hilscher.com

Support

Phone: +49 (0) 6190 9907-99
E-mail: de.support@hilscher.com

SUBSIDIARIES

China

Hilscher Systemautomation (Shanghai) Co. Ltd.
200010 Shanghai
Phone: +86 (0) 21-6355-5161
E-mail: info@hilscher.cn

Support

Phone: +86 (0) 21-6355-5161
E-mail: cn.support@hilscher.com

France

Hilscher France S.a.r.l.
69500 Bron
Phone: +33 (0) 4 72 37 98 40
E-mail: info@hilscher.fr

Support

Phone: +33 (0) 4 72 37 98 40
E-mail: fr.support@hilscher.com

India

Hilscher India Pvt. Ltd.
Pune, Delhi, Mumbai
Phone: +91 8888 750 777
E-mail: info@hilscher.in

Italy

Hilscher Italia S.r.l.
20090 Vimodrone (MI)
Phone: +39 02 25007068
E-mail: info@hilscher.it

Support

Phone: +39 02 25007068
E-mail: it.support@hilscher.com

Japan

Hilscher Japan KK
Tokyo, 160-0022
Phone: +81 (0) 3-5362-0521
E-mail: info@hilscher.jp

Support

Phone: +81 (0) 3-5362-0521
E-mail: jp.support@hilscher.com

Korea

Hilscher Korea Inc.
Seongnam, Gyeonggi, 463-400
Phone: +82 (0) 31-789-3715
E-mail: info@hilscher.kr

Switzerland

Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-mail: info@hilscher.ch

Support

Phone: +49 (0) 6190 9907-99
E-mail: ch.support@hilscher.com

USA

Hilscher North America, Inc.
Lisle, IL 60532
Phone: +1 630-505-5301
E-mail: info@hilscher.us

Support

Phone: +1 630-505-5301
E-mail: us.support@hilscher.com