

Compact Logic Firmware Manager – User Manual

Documentation version: 27 Sep 2026

Application: Compact Logic Firmware Manager

Platform: Windows / WPF

Purpose: Detecting, analyzing, backing up, and updating Compact Logic devices based on ESP32-S3 and ESP32-P4.

Note on the documentation version

This manual describes the currently implemented functionality of the application: device scanning, COM auto-scan, device history, ROM bootloader and RAM stub detection, firmware catalog, backup catalog, full flash backup, automatic firmware status checking, and firmware updates with selectable flash regions. If the software evolves, this manual should be updated together with the application version.

1. Overview

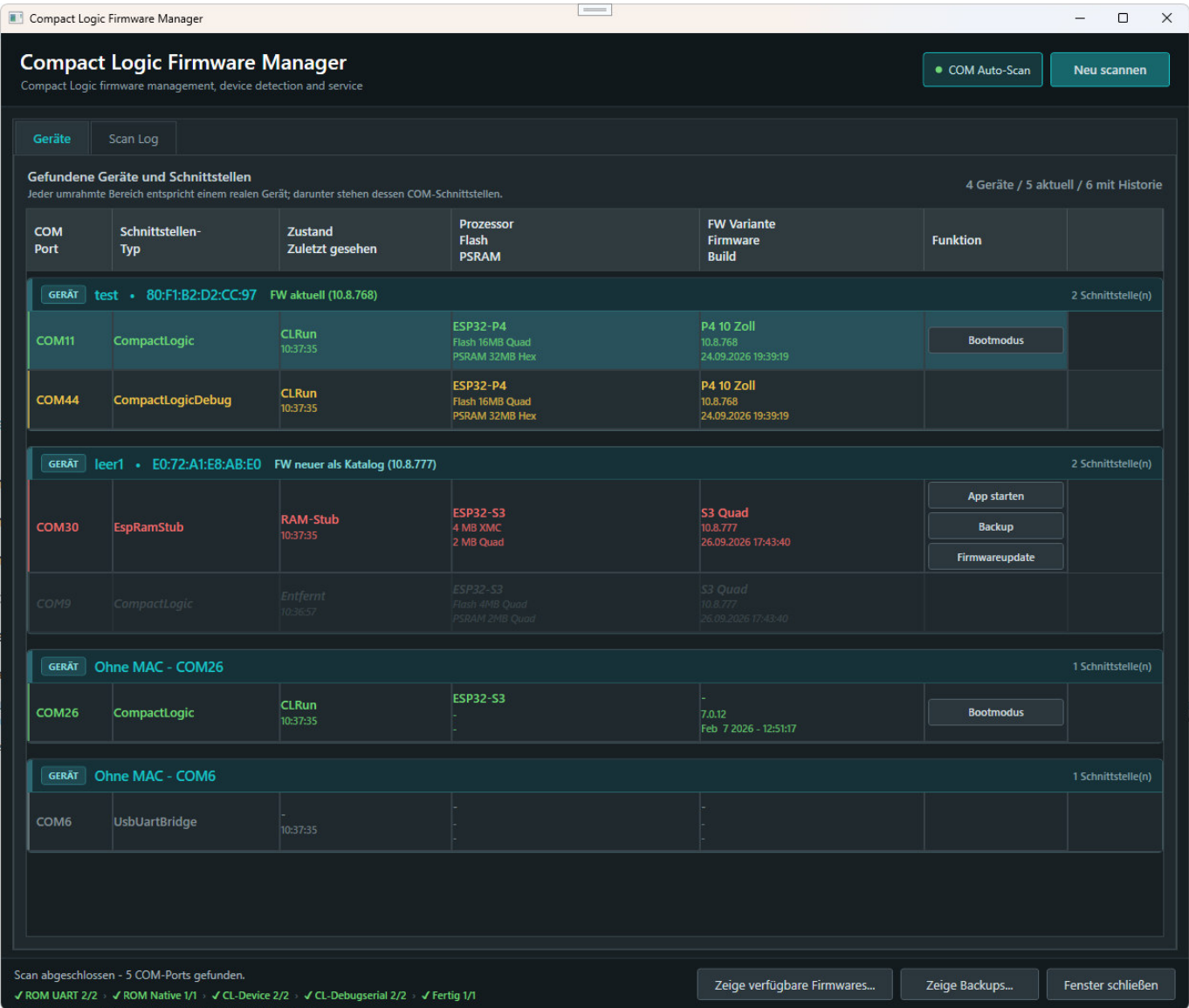
The Compact Logic Firmware Manager is a central service application for Compact Logic devices. The program detects connected devices and their interfaces, displays hardware and firmware information, can switch devices between application mode and boot/service mode, back up the complete flash, and write selected areas of a firmware image to the device.

A key feature of the application is the automatic assignment of multiple COM interfaces to one physical device. Whenever a MAC address is available, all related interfaces are shown in a common device block.

The manager recognizes three main operating states of an ESP device:

State	Meaning	Typical functions
CLRun / CompactLogic	The normal Compact Logic application is running.	Read device data, switch to boot mode
ESP ROM bootloader	The ESP's internal ROM bootloader is active.	Load stub, backup, firmware update, start app
RAM stub	The temporary Espressif flasher stub runs in RAM.	Fast backup, firmware update, further service operations, start app

The RAM stub is **not an error state**. It is the preferred service state for fast flash operations. After a backup or firmware update, the device may intentionally remain in the RAM stub so that further operations can be performed without loading the stub again.



2. Important Terms

2.1 CompactLogic / CLRun

CompactLogic refers to the normal USB/COM interface of the running device firmware. The state is shown in the user interface as CLRun .

2.2 CompactLogicDebug

CompactLogicDebug is an additional debug/service interface if the firmware or hardware provides one. It is displayed separately, but is assigned to the same physical device via the MAC address.

2.3 ESP ROM Bootloader

The ROM bootloader is the bootloader built into the ESP chip. In this state, the manager can identify the chip and load a RAM stub. The state is displayed in red.

2.4 RAM Stub

The RAM stub is a temporary program that the manager loads into the ESP's RAM. It primarily speeds up flash read and write operations. The stub remains in RAM only until a full reset and is also displayed in red.

2.5 CLMETA

CLMETA is a Compact Logic metadata partition. It can contain, among other things:

- Device name
- Factory MAC or device identity
- Chip type
- Hardware/firmware variant
- Installed firmware version
- Firmware date
- UMSR version
- Format and layout version

For newly delivered firmware that has never been started, CLMETA may be empty. On an installed or previously started device, CLMETA is populated by the firmware with device-specific information.

2.6 NVM

The NVM/NVS partition contains non-volatile settings. During a normal firmware update, it is **not** overwritten by default.

2.7 APP

The app partition contains the actual device firmware. For a normal update of a device that has already been programmed, only this partition is usually required.

2.8 SPIFFS

The SPIFFS/storage partition contains file system data. It can be updated together with the app if required, but is not automatically selected during a normal app update.

3. Program Directories and Files

The application works relative to its own program directory.

3.1 Firmware Directory

Firmware images are stored in the subdirectory:

```
firmware
```

`.bin` files directly in this directory are analyzed. Subdirectories are not scanned recursively.

3.2 Backup Directory

Flash backups are stored in the subdirectory:

```
fw_backup
```

The directory is created automatically if required.

3.3 Firmware File Naming Scheme

The following naming scheme is currently supported for release firmware:

```
CLFW_<Target>-<Variant>_<YYYYMMDD>_<HHMMSS>[_Options].bin
```

Examples:

```
CLFW_ESP32-S3-Quad_20260926_081451_with_storage.bin  
CLFW_ESP32-S3-Octal_20260926_081451_with_storage.bin  
CLFW_ESP32-P4-Dsi7_20260926_081423_with_storage.bin  
CLFW_ESP32-P4-Dsi10_20260926_081423_with_storage.bin
```

Supported variants:

Target	File name	Display
ESP32-S3	Quad	S3 Quad
ESP32-S3	Octal	S3 Octal
ESP32-P4	Dsi7	P4 7-inch
ESP32-P4	Dsi10	P4 10-inch

The firmware date shown in the catalog view is primarily derived from this file name. The internal app build time is also shown in the analysis area.

3.4 Backup File Naming Scheme

Backups are automatically given a name according to the following scheme:

```
YYYYMMDD_HHMMSS_MAC_DeviceName.bin
```

Example:

```
20260926_133245_E072A1E8ABE0_leer1.bin
```

Colons are removed from the MAC address. Invalid characters and spaces in the device name are sanitized for the file name.

During a running backup, a temporary file with the `.partial` extension is written first. Only after a completely successful backup is it renamed to the final `.bin` file. If the backup is canceled or fails, the temporary file is deleted whenever possible.

4. Program Startup

Two main operations are performed when the application starts:

1. The firmware catalog from the `firmware` subdirectory is created and cached in memory.
2. A full device scan is then performed.

The firmware catalog is later used to automatically indicate whether newer firmware is available for a detected device.

If the firmware directory does not exist or contains no `.bin` files, the application can still scan devices and create backups. However, the firmware status check and firmware update will then

not be able to offer matching images.

5. Main Window

The main window consists of the header area, device view, scan log, and status area.

5.1 COM Auto-Scan

COM Auto-Scan enables or disables the automatic response to changes in COM interfaces.

When enabled, the application monitors Windows device changes. If a COM port is added or removed, the application briefly waits for a stable USB/COM state and then automatically performs a new scan.

This is particularly useful for actions such as:

- Switching to boot mode
- Starting the application from the bootloader or RAM stub
- USB re-enumeration of an ESP32-S3
- Connecting or disconnecting a device

Auto-scan is disabled or suppressed during a flash backup so that the active serial connection is not disturbed.

5.2 Scan Again

Scan again manually starts a complete device scan at any time, provided no other blocking action is currently running.



6. Device Scan

The device scan consists of several probing phases. The status line shows progress for, among other things:

- ROM UART
- ROM Native
- CL-Device
- CL-Debugserial

- Done

Depending on the connected hardware, not every interface can respond to every probe. For example, a USB-UART adapter may be present even though the ESP does not respond on that interface in its current state.

6.1 Detected Interface Types

Display	Meaning
CompactLogic	Running Compact Logic application
CompactLogicDebug	Compact Logic debug interface
EspRomBootloader	ESP ROM bootloader detected
EspRamStub	Already running RAM flasher stub detected
EspressifNative	Native Espressif interface present, but currently not assigned to an active loader state
UsbUartBridge	USB-UART bridge detected, currently without a matching device response

6.2 Device Row Colors

Color	Meaning
Green	Running Compact Logic application
Yellow	Compact Logic debug interface
Red	ROM bootloader or RAM stub / service state
Gray / transparent	Historical interface that is currently no longer present

6.3 State Indicators

Possible values in the **State** column include:

State	Meaning
CLRun	Compact Logic application is running
Bootloader	ROM bootloader active
RAM-Stub	Flasher stub active in RAM
Removed	Interface was previously present but is currently no longer available
In use	COM port is being used by another application
Cannot be opened	Port could not be opened
Reset required	Native ESP interface is known, but the loader does not respond
Error	Error during device probing

The application regularly checks whether currently present ports are occupied by other programs.

7. Device Blocks and Device History

As soon as the MAC address of a device is known, all associated interfaces are displayed in a common device block.

Example:

```
test • 80:F1:B2:D2:CC:97
```

Below the header, for example, both the normal USB application interface and a debug UART interface may be shown.

If no MAC address can be determined, the application creates a separate block, for example:

```
No MAC - COM6
```

During its runtime, the application keeps a history of COM interfaces that have already been seen. Removed ports remain visible in a grayed-out state. This makes it possible to see which interface existed, for example, before switching to boot mode.

Geräte

Scan Log

Gefundene Geräte und Schnittstellen

Jeder umrahmte Bereich entspricht einem realen Gerät; darunter stehen dessen COM-Schnittstellen.

4 Geräte / 5 aktuell / 6 mit Historie

COM Port	Schnittstellen-Typ	Zustand Zuletzt gesehen	Prozessor Flash PSRAM	FW Variante Firmware Build	Funktion	
GERÄT test • 80:F1:B2:D2:CC:97 FW aktuell (10.8.768) 2 Schnittstelle(n)						
COM11	CompactLogic	CLRun 10:37:35	ESP32-P4 Flash 16MB Quad PSRAM 32MB Hex	P4 10 Zoll 10.8.768 24.09.2026 19:39:19	Bootmodus	
COM44	CompactLogicDebug	CLRun 10:37:35	ESP32-P4 Flash 16MB Quad PSRAM 32MB Hex	P4 10 Zoll 10.8.768 24.09.2026 19:39:19		
GERÄT leer1 • E0:72:A1:E8:AB:E0 FW neuer als Katalog (10.8.777) 2 Schnittstelle(n)						
COM30	EspRamStub	RAM-Stub 10:37:35	ESP32-S3 4 MB XMC 2 MB Quad	S3 Quad 10.8.777 26.09.2026 17:43:40	App starten Backup Firmwareupdate	
COM9	CompactLogic	Entfernt 10:36:57	ESP32-S3 Flash 4MB Quad PSRAM 2MB Quad	S3 Quad 10.8.777 26.09.2026 17:43:40		
GERÄT Ohne MAC - COM26 1 Schnittstelle(n)						
COM26	CompactLogic	CLRun 10:37:35	ESP32-S3 -	- 7.0.12 Feb 7 2026 - 12:51:17	Bootmodus	
GERÄT Ohne MAC - COM6 1 Schnittstelle(n)						
COM6	UsbUartBridge	- 10:37:35	-	-		

8. Displayed Device Information

The device view shows the following information for each interface:

COM Port

Windows COM port, for example `COM9` , `COM30` , or `COM44` .

Interface Type

Detected logical role of the interface, for example `CompactLogic` , `EspRamStub` , or `CompactLogicDebug` .

State / Last Seen

Current state and time of the last successful scan.

Processor / Flash / PSRAM

Depending on the available source, the following information may be displayed:

- ESP32-S3 or ESP32-P4

- Flash size and flash manufacturer/mode
- PSRAM size and mode

FW Variant / Firmware / Build

Depending on the device and operating state, this information comes from the Compact Logic API or from CLMETA.

Examples:

```
S3 Quad  
10.8.768  
24.09.2026 19:39:19
```

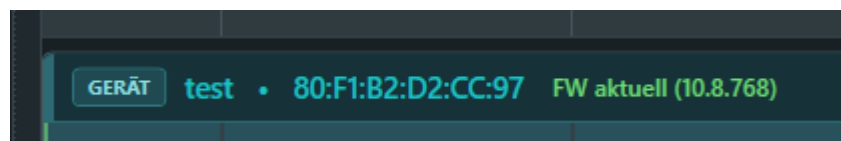
or:

```
P4 10-inch  
10.8.766  
24.09.2026 19:39:19
```

9. Automatic Firmware Status Check

After each device scan, the firmware catalog loaded at program startup is compared with the detected devices.

The status information appears in the device block header after the device name and MAC address.



9.1 Possible Status Messages

Display	Meaning
FW up to date (x.y.z)	Installed version matches the highest matching catalog version
FW update x.y.z available	A higher matching version is available in the catalog
FW newer than catalog (x.y.z)	The device has a higher version than all matching catalog files
Firmware available (x.y.z)	Matching firmware is available for a device without installed firmware
No firmware installed	No installed version and no comparable catalog version are available
No matching firmware in catalog	Processor/variant do not match any firmware file
FW check: variant unknown	A unique variant check is not possible
FW check: version not comparable	Installed or available version does not have a parseable version format
Firmware catalog not available	The catalog could not be loaded

9.2 Assignment by Processor and Variant

For the automatic status check, the firmware must match the target processor and the installed variant.

For the ESP32-S3, `S3 Quad` and `S3 Octal` are distinguished.

For the ESP32-P4, the known installed variant is used for the automatic status check, for example `P4 7-inch` or `P4 10-inch`.

9.3 Version Comparison

Version comparison supports numeric versions with up to four components, for example:

```
10.8.768
10.8.769
10.8.769.1
v10.8.770
```

An optional suffix is also supported. If the numeric components are identical, a version without a suffix is considered a regular release version and is ranked higher than a version with a suffix.

Example:

```
10.8.769 > 10.8.769-test
```

For reliable automatic update checking, the release BIN version number must be included in the ESP image's app descriptor. If CLMETA is still empty, the firmware file version is read from this app descriptor.

10. Function Column

Function buttons are only shown when the corresponding action is meaningful and available in the current state.

10.1 Boot Mode

The **Boot mode** button is offered when the normal Compact Logic application is running.

After clicking it, the application attempts to switch the device into the ESP bootloader. On the ESP32-S3, a Compact Logic command sent to the running firmware is preferred. Depending on the hardware and interface, UART/reset mechanisms may be used as alternatives.

After the state change, the application waits for stable COM interfaces and performs a new scan.

During the transition, COM ports may briefly disappear and reappear with a different port number or device identifier. This is normal, especially for native USB interfaces.

10.2 Start App

The **Start app** button is offered when a device is in the ROM bootloader or RAM stub and known Compact Logic firmware is present.

For an ESP32-S3 in the RAM stub, a full reset is triggered after the forced download-boot state has been cleared. The normal firmware then starts and the USB interfaces re-enumerate.

After starting the app, the application performs a new scan.

10.3 Backup

The **Backup** button is available in the ROM bootloader and RAM stub states.

See section [12. Flash Backup](#).

10.4 Firmware Update

The **Firmware update** button is available in the ROM bootloader and RAM stub states.

See section [15. Firmware Update](#).

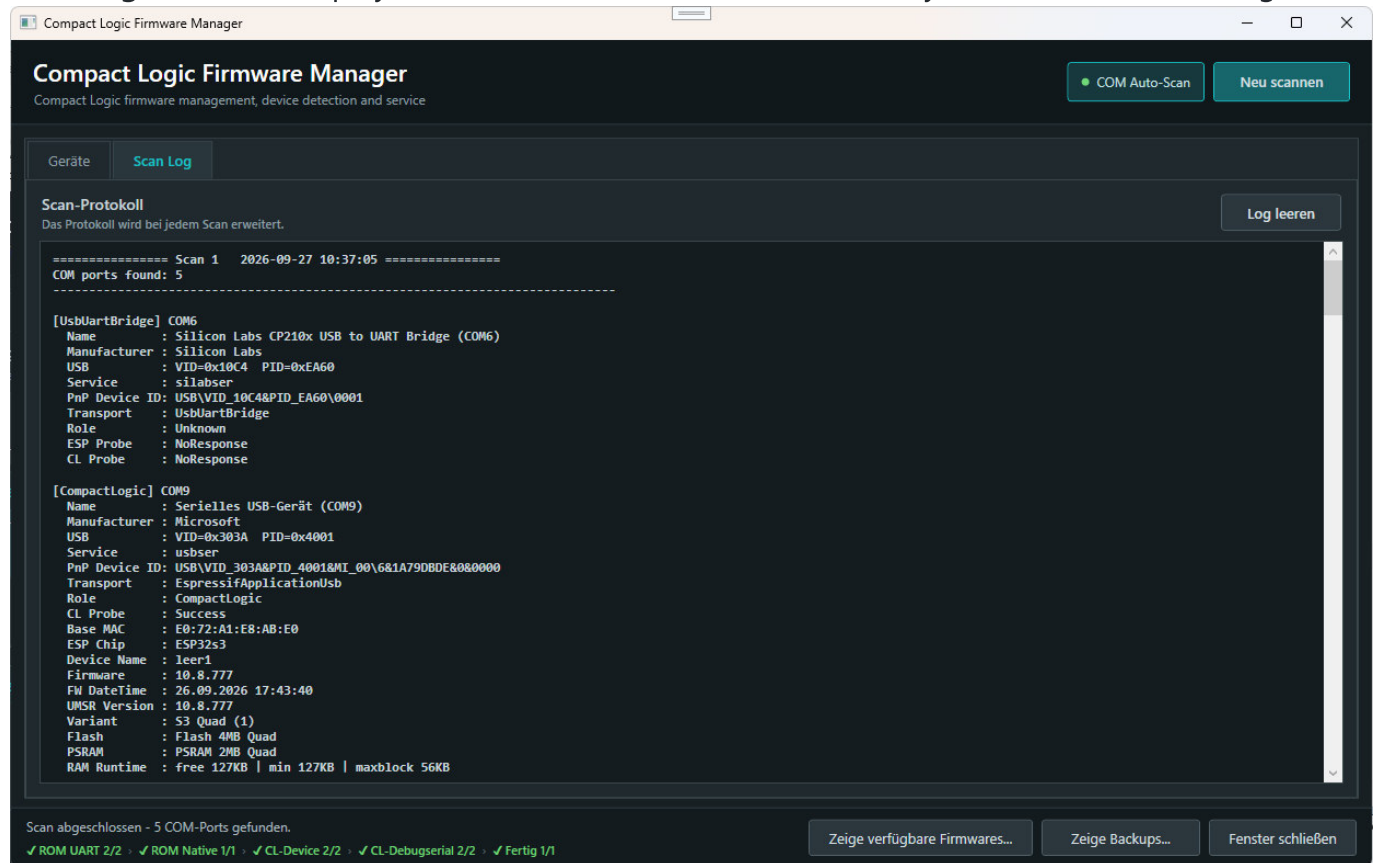
11. Scan Log

The **Scan Log** tab records the results of each device scan.

Depending on the interface, the log contains, among other things:

- COM port
- Windows device name
- Manufacturer
- USB VID/PID
- Windows service/driver
- PnP Device ID
- Detected transport
- Detected role
- ESP probe result
- CL probe result
- MAC address
- Chip type
- Flash/CLMETA result
- Device name
- Firmware version
- Firmware date
- Variant
- Flash and PSRAM information
- Sources of the detected information
- CLMETA format details

Clear log clears the display. This does not affect the device history or the firmware catalog.



12. Flash Backup

12.1 Purpose

The backup function reads the **entire physical flash** of the device from address `0x00000000` up to the reliably detected flash size.

A backup therefore generally contains all regions present in flash, for example:

- Bootloader
- Partition table
- NVM/NVS
- CLMETA
- App
- SPIFFS/storage
- Additional data partitions

12.2 Requirements

A backup can only be started if:

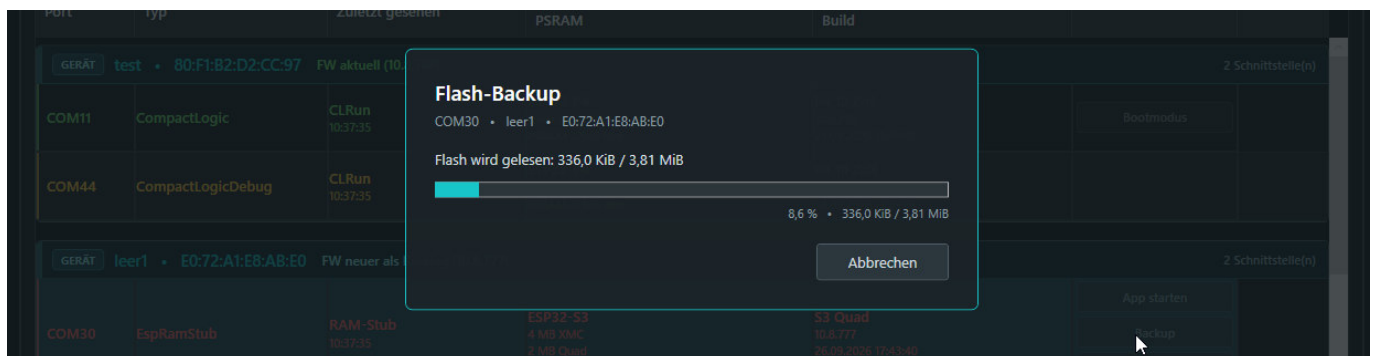
- the device is present,
- it is in the ROM bootloader or RAM stub state,
- the COM port is not occupied by another program,
- the physical flash size can be determined reliably.

The application **does not start a full backup using a guessed flash size.**

12.3 Procedure

When a backup is started:

1. The main user interface is locked for the duration of the backup.
2. If a RAM stub is already running, it is reused directly.
3. If the ROM bootloader is active, the appropriate RAM stub is loaded.
4. The service baud rate is increased to 921600 baud if required.
5. The entire flash is read.
6. The data is initially written to a `.partial` file.
7. After a completely successful read, the file is renamed to `.bin`.
8. The device intentionally remains in the RAM stub afterwards.



12.4 Cancel

Cancel stops the backup. An incomplete `.partial` file is not retained as a valid backup.

The main window cannot be closed while a backup is running. The backup must first finish or be canceled.

12.5 State After the Backup

After a successful backup, the device remains in the **RAM stub**. This is intentional.

This allows further service operations to be performed immediately, for example:

- another backup,
- a firmware update,
- additional flash functions in the future.

Start app exits the stub and starts the normal firmware.

13. Firmware Catalog

The **Show available firmware...** button opens the firmware catalog for the `firmware` subdirectory.

The catalog is already created and cached at program startup. **Reload catalog** can be used to analyze the directory again. The main screen then uses the updated catalog for firmware status checking.

Verfügbare Firmwares

Verfügbare Firmwares

L:\CL_Tools_VS\EspDotNet-main\CL_Flash\bin\Debug\net10.0-windows\firmware

Katalog neu laden

Gerät / Abbild	Datum	Variante	Firmware	Target	Enthaltene Daten	Status	Dateiname
CL-Firmware P4 10 Zoll	26.09.2026 10:34:47	P4 10 Zoll	10.8.768	ESP32-P4	Boot, PartTab, APP, SPIFFS	OK	CLFW_ESP32-P4-Dsi10_20260926_103447_with_stora
CL-Firmware P4 7 Zoll	26.09.2026 10:34:47	P4 7 Zoll	10.8.768	ESP32-P4	Boot, PartTab, APP, SPIFFS	OK	CLFW_ESP32-P4-Dsi7_20260926_103447_with_storag
CL-Firmware S3 Octal	26.09.2026 10:08:55	S3 Octal	10.8.768	ESP32-S3	Boot, PartTab, APP, SPIFFS	OK	CLFW_ESP32-S3-Octal_20260926_100855_with_storax
CL-Firmware S3 Quad	26.09.2026 10:08:55	S3 Quad	10.8.768	ESP32-S3	Boot, PartTab, APP, SPIFFS	OK	CLFW_ESP32-S3-Quad_20260926_100855_with_stora

Analyse / Log des ausgewählten Abbilds
CLFW_ESP32-P4-Dsi7_20260926_103447_with_storage.bin
Quelle: Firmware
Gerät / Abbild: CL-Firmware P4 7 Zoll
Datum: 26.09.2026 10:34:47
Enthalten: Boot, PartTab, APP, SPIFFS
Pfad: L:\CL_Tools_VS\EspDotNet-main\CL_Flash\bin\Debug\net10.0-windows\firmware\CLFW_ESP32-P4-Dsi7_20260926_103447_with_storage.bin
Größe: 5,75 MiB (6.029.312 Byte)
SHA-256: aae5c48b29018dfb66457b1beab54ac2b24114d15954381eb6789d0b1f05d747
Variante: P4 7 Zoll
Varianten-Quelle: Dateiname
Dateiname-Metadaten: Target=ESP32-P4, Variante=Dsi7, Build=26.09.2026 10:34:47, Optionen=with_storage
Projekt: umsrstation_p4_uni
Firmware: 10.8.768
Build: Sep 26 2026 10:35:18
Target: ESP32-P4
Partitionstabelle: 0x00000000 / MDS OK
Bootloader: 0x00002000-0x5B80 (22,88 KiB)

Partitionen:
nvs 0x00000000 - 0x000000FF (24 KiB) leer (0xFF) Typ=data/nvs
clmeta 0x00000000 - 0x000000FF (4 KiB) leer (0xFF) Typ=data/clmeta
factory 0x00010000 - 0x00040000 (4 MiB) Daten vorhanden Typ=app/factory
spiffs 0x000410000 - 0x0005BFFFF (1,69 MiB) Daten vorhanden Typ=data/spiffs

Hinweise:
- CLMETA-Partition ist vorhanden, im Full-Image aber leer.

4 Abbild-Datei(en), 4 grundsätzlich analysierbar.

Fenster schließen

13.1 Firmware Catalog Columns

Column	Content
Device / Image	For firmware, for example <code>CL-Firmware S3 Quad</code>
Date	Release/package timestamp from the standardized firmware file name
Variant	S3 Quad, S3 Octal, P4 7-inch, or P4 10-inch
Firmware	Firmware version from CLMETA or the app descriptor
Target	Target processor, for example ESP32-S3 or ESP32-P4
Contained Data	Compact list of data regions that are actually present
Status	OK, warnings, or analysis errors
File Name	Complete file name

13.2 Firmware Variant

Variant information is determined in the following order of priority:

1. valid CLMETA information, if available,
2. for release firmware, the defined file name,
3. otherwise unknown.

Because CLMETA can still be empty in newly delivered firmware before the first startup, the file name is currently an important fallback.

13.3 Firmware Version of a Firmware File

The firmware version is preferably read from CLMETA if valid information is already present there. If CLMETA is empty, the version is read from the ESP app descriptor of the app partition.

For automatic update checking to work correctly, the release firmware should contain the same version number in the app descriptor that will later be reported through CLMETA or the Compact Logic API.

13.4 Contained Data

The **Contained Data** column can contain, for example:

Boot, PartTab, NVM, CLMETA, APP, SPIFFS, DATA

NVM, CLMETA, APP, SPIFFS, and DATA are only listed as data regions if the corresponding partition actually contains data and not only `0xFF` .

13.5 Analysis Area

The lower analysis area shows detailed information about the selected file, including:

- Source: firmware or backup
- Device/image
- Date
- Contained regions
- Full path
- File size
- SHA-256
- Variant and source of the variant information
- Metadata read from the file name
- Project name
- Firmware version
- Internal build time
- Target
- Partition table address and MD5 status
- Bootloader region
- CLMETA content, if present
- All partitions with address, size, type, and data status
- Notes
- Warnings

14. Backup List

The **Show backups...** button opens the same catalog dialog for the `fw_backup` directory.

For backups, the CLMETA partition is analyzed explicitly. This allows the catalog to display, among other things, the device name, MAC address, variant, and installed firmware version of the backed-up device.

Gespeicherte Backups

L:\CL_Tools_VS\EspDotNet-main\CL_Flash\bin\Debug\net10.0-windows\fw_backup

Katalog neu laden

Gerät / Abbild	Datum	Variante	Firmware	Target	Enthaltene Daten	Status	Dateiname
leer1	27.09.2026 10:42:31	S3 Quad	10.8.777	ESP32-S3	Boot, PartTab, NVM, CLMETA, APP, SPIFFS	OK	20260927_104231_E072A1E8ABE0_leer1.bin
test	26.09.2026 14:51:27	P4 10 Zoll	10.8.768	ESP32-P4	Boot, PartTab, NVM, CLMETA, APP, SPIFFS	OK	20260926_145127_80F1B2D2CC97_test.bin
leer1	26.09.2026 06:16:58	S3 Quad	10.8.768	ESP32-S3	Boot, PartTab, NVM, CLMETA, APP, SPIFFS	OK	20260926_061658_E072A1E8ABE0_leer1.bin
leer1	25.09.2026 19:50:59	S3 Quad	10.8.768	ESP32-S3	Boot, PartTab, NVM, CLMETA, APP, SPIFFS	OK	20260925_195059_E072A1E8ABE0_leer1.bin
leer1	25.09.2026 19:40:23	S3 Quad	10.8.768	ESP32-S3	Boot, PartTab, NVM, CLMETA, APP, SPIFFS	OK	20260925_194023_E072A1E8ABE0_leer1.bin
leer1	25.09.2026 19:03:24	S3 Quad	10.8.768	ESP32-S3	Boot, PartTab, NVM, CLMETA, APP, SPIFFS	OK	20260925_190324_E072A1E8ABE0_leer1.bin
leer1	25.09.2026 17:19:10	S3 Quad	10.8.768	ESP32-S3	Boot, PartTab, NVM, CLMETA, APP, SPIFFS	OK	20260925_171910_E072A1E8ABE0_leer1.bin
Backup	25.09.2026 16:05:11	-	10.8.766	ESP32-P4	Boot, PartTab, NVM, CLMETA, APP, SPIFFS	OK	20260925_160511_80F1B2D2CC97_test.bin
leer1	25.09.2026 15:23:36	S3 Quad	10.8.768	ESP32-S3	Boot, PartTab, NVM, CLMETA, APP, SPIFFS	OK	20260925_152336_E072A1E8ABE0_leer1.bin
leer1	25.09.2026 13:38:47	S3 Quad	10.8.768	ESP32-S3	Boot, PartTab, NVM, CLMETA, APP, SPIFFS	OK	20260925_133847_E072A1E8ABE0_leer1.bin

Analyse / Log des ausgewählten Abbilds

20260927_104231_E072A1E8ABE0_leer1.bin
Quelle: Backup
Gerät / Abbild: leer1
Datum: 27.09.2026 10:42:31
Enthalten: Boot, PartTab, NVM, CLMETA, APP, SPIFFS
Pfad: L:\CL_Tools_VS\EspDotNet-main\CL_Flash\bin\Debug\net10.0-windows\fw_backup\20260927_104231_E072A1E8ABE0_leer1.bin
Größe: 3,81 MiB (3.997.096 Byte)
SHA-256: 3c2480ede05b93d87e55c173ce94221a8578cf8b5267bada739572028e7b92d
Variante: S3 Quad
Varianten-Quelle: CLMETA
Projekt: unsrotation
Firmware: 10.8.777
Build: 26.09.2026 17:43:40
Target: ESP32-S3
Partitionstabelle: 0x00000000 / MDS OK
Bootloader: 0x00000000+0x58c0 (22,19 KiB)

CLMETA:
Device Name: leer1
Factory MAC: E0:72:A1:E8:AB:E0
Chip: ESP32-S3
Variante: S3 Quad
Firmware: 10.8.777
Firmware-Datum: 26.09.2026 17:43:40
UMSR: 10.8.777
Format / Layout: 1 / 1
Identity / Struct: 96 / 804 Byte

Partitionen:
nvm Nvm 0x00000000 - 0x00008000 (24 KiB) Daten vorhanden Typ=data/nvm
clmeta CLMeta 0x00008000 - 0x0000FFFF (4 KiB) Daten vorhanden Typ=data/0x40
factory Application 0x00020000 - 0x00021FFF (2 MiB) Daten vorhanden Typ=app/factory
storage SpiFFS 0x00220000 - 0x0022FFFF (1,69 MiB) Daten vorhanden Typ=data/spiffs

Hinweise:
- NVM/NVS enthält Daten. Bei einem Update sollte diese Partition standardmäßig erhalten bleiben.

10 Abbild-Datei(en), 10 grundsätzlich analysierbar.

Fenster schließen

14.1 Date Source

For backups, the displayed date is preferably taken from the backup file name generated by the application. This means the date corresponds to the time the backup was created.

14.2 Variant

For backups, CLMETA is the authoritative source for variant information. The application does not attempt to guess the hardware variant from an arbitrary backup file name.

14.3 CLMETA Warnings

If a backup contains a CLMETA partition that is empty or invalid, this is explicitly reported as a note or warning in the analysis area.

15. Firmware Update

15.1 Requirements

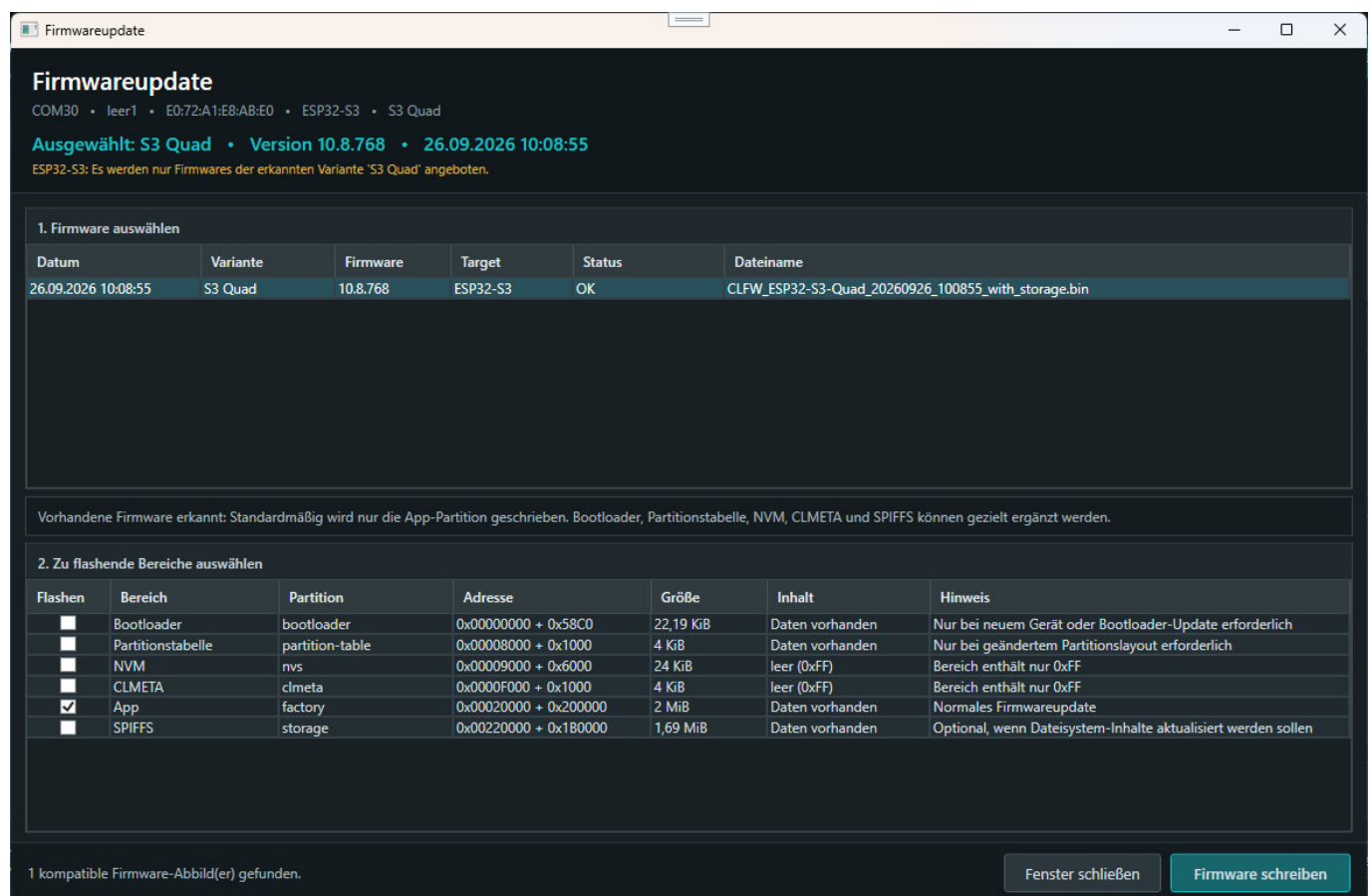
A firmware update is only available if the device is in the ROM bootloader or RAM stub state and the COM port is available.

Recommended procedure before an update:

1. Scan the device normally.
2. If possible, create a current backup.
3. Put the device into boot/service mode.
4. Open **Firmware update**.

15.2 Step 1 – Select Firmware

The upper area of the update window shows the device data and the selectable firmware images.



ESP32-S3

For the ESP32-S3, only firmware images matching the processor and detected S3 variant are offered.

Examples:

- S3 Quad → S3 Quad firmware only
- S3 Octal → S3 Octal firmware only

If the S3 variant is not known, a unique automatic assignment is not possible.

ESP32-P4

For the ESP32-P4, **both display variants** are deliberately offered:

- P4 7-inch / DSI7
- P4 10-inch / DSI10

The user must select the variant that matches the display actually connected to the device.

Multiple Firmware Versions

Multiple firmware files of the same variant may exist in the firmware directory at the same time. This makes it possible, for example, to offer a release version and an experimental test version in parallel.

The user chooses the desired version in the update window.

15.3 Step 2 – Select Flash Regions

After selecting the firmware, a flash plan is created.

The list can include, among other things:

- Bootloader
- Partition table
- NVM
- CLMETA
- App
- SPIFFS
- Additional data regions

For each region, the following are displayed:

- **Flash** checkbox
- Region/role
- Partition name
- Flash address

- Size
- Content status
- Note on typical usage

Flashen	Bereich	Partition	Adresse	Größe	Inhalt	Hinweis
☐	Bootloader	bootloader	0x00000000 + 0x58C0	22,19 KiB	Daten vorhanden	Nur bei neuem Gerät oder Bootloader-Update erforderlich
☐	Partitionstabelle	partition-table	0x00008000 + 0x1000	4 KiB	Daten vorhanden	Nur bei geändertem Partitionslayout erforderlich
☐	NVM	nvs	0x00009000 + 0x6000	24 KiB	leer (0xFF)	Bereich enthält nur 0xFF
☐	CLMETA	clmeta	0x0000F000 + 0x1000	4 KiB	leer (0xFF)	Bereich enthält nur 0xFF
☒	App	factory	0x00020000 + 0x200000	2 MiB	Daten vorhanden	Normales Firmwareupdate
☐	SPIFFS	storage	0x00220000 + 0x180000	1,69 MiB	Daten vorhanden	Optional, wenn Dateisystem-Inhalte aktualisiert werden sollen

15.4 Default Selection for an Empty MCU

If the manager detects an empty flash, all regions that are completely present in the firmware image are preselected.

This enables full initial programming.

15.5 Default Selection for Existing Firmware

On an already programmed device, only the primary app partition is selected by default.

Typical normal case:

- ☐ Bootloader
- ☐ Partition table
- ☐ NVM
- ☐ CLMETA
- ☒ App
- ☐ SPIFFS

The additional regions can be deliberately enabled if required.

15.6 Meaning of the Regions During an Update

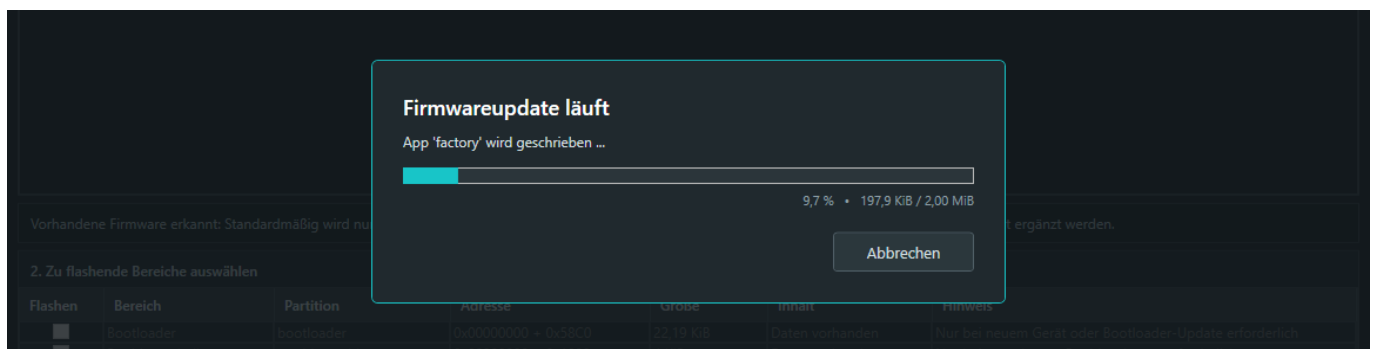
Region	Recommendation
App	Default selection for a normal firmware update
SPIFFS	Also select if file system/web/resource data must be updated
NVM	Normally do not overwrite; may contain device settings
CLMETA	Normally do not overwrite on existing devices; contains device identity and metadata
Partition table	Only required if the partition layout has changed or an empty device is being fully programmed
Bootloader	Only required for an empty device or a deliberate bootloader update
Additional data regions	Only select if their purpose is known

15.7 Write Firmware

Clicking **Write firmware** first displays a safety confirmation. The firmware file and selected regions are shown again there.

After confirmation:

1. The existing RAM stub is reused or loaded if required.
2. The service baud rate is set to 921600 baud.
3. Each selected region is extracted from the full image.
4. The region is written to its original flash address.
5. After each region, an MD5 is calculated over the written flash and compared with the source data.
6. Only after successful verification does the application continue with the next region.
7. When finished, the device remains in the RAM stub.



15.8 Canceling a Firmware Update

A firmware update can be stopped using **Cancel**.

Important: The flash region that was being processed last may then be incomplete. The device should **not** simply be started into the app afterwards. Instead, restart the firmware update and completely rewrite the required regions, or at least the affected region.

15.9 State After a Successful Update

After writing, the device remains in the RAM stub. This allows the following to be performed immediately:

- another flash operation,
- a backup,
- a verification, or
- **Start app**.

The new firmware is not executed until **Start app** is used or a suitable reset is performed.

16. Recommended Workflows

16.1 Normal Firmware Update of an Existing Device

```
Start program
→ device is scanned
→ check firmware status
→ Boot mode
→ optional backup
→ Firmware update
→ select matching firmware
→ keep only App preselected
→ Write firmware
→ Start app
→ verify final scan
```

16.2 Update Including SPIFFS

```
Boot mode
→ Backup
→ Firmware update
→ select firmware
→ select App + SPIFFS
→ Write firmware
→ Start app
```

16.3 Full Programming of an Empty Controller

```
Put controller into ESP bootloader
→ Scan
→ open Firmware update
→ select matching firmware/variant
→ verify all required regions
→ write complete image
→ Start app
→ verify new scan and CLMETA initialization
```

Note for ESP32-S3: The S3 hardware variant must be known for automatic firmware filtering. If it cannot be determined on a completely empty device, the variant must be clarified before full programming.

16.4 Testing Experimental Firmware

Multiple versions of the same variant may exist in the `firmware` directory at the same time.

```
Create backup
→ Firmware update
→ deliberately select the desired experimental version
→ flash App
→ Start app and test
```

The previous backup serves as a safety copy. An automatic “restore backup” function is not described as a separate operation in the current version; a backup is primarily a complete flash image for backup and analysis purposes.

17. Safety Notes

17.1 Back Up Before Every Critical Update

Before major changes, especially to the partition table, NVM, CLMETA, or bootloader, a complete backup should be created.

17.2 Do Not Transfer CLMETA Carelessly

CLMETA can contain a specific device MAC or device-specific identity. A CLMETA partition from another device must not be written to an existing device without verification.

17.3 Preserve NVM

NVM can contain calibration, configuration, or user data. During normal updates, this partition should not be selected.

17.4 Observe the Partition Layout

If the partition layout changes between firmware versions, an app-only update may not be sufficient. In this case, the partition table and, if necessary, additional regions must be updated accordingly.

17.5 Power Supply and USB Connection

Do not intentionally disconnect the power supply or USB connection during a flash operation.

17.6 Close Other Serial Applications

Terminal programs, debuggers, or other applications can occupy a COM port exclusively. The manager displays such a port as `In use` or unavailable.

18. Preparing the Firmware Catalog for Release Files

To allow the manager to reliably assign and compare firmware files, release files should provide the following information.

18.1 Target and Variant

If CLMETA is still empty, the variant can currently be read from the standardized file name.

Recommended names:

```
CLFW_ESP32-S3-Quad_YYYYMMDD_HHMMSS_with_storage.bin
CLFW_ESP32-S3-Octal_YYYYMMDD_HHMMSS_with_storage.bin
CLFW_ESP32-P4-Dsi7_YYYYMMDD_HHMMSS_with_storage.bin
CLFW_ESP32-P4-Dsi10_YYYYMMDD_HHMMSS_with_storage.bin
```

18.2 Version Number

The release firmware version number should be included in the ESP app descriptor (`esp_app_desc_t.version`).

The installed firmware later reports the version through CLMETA or the Compact Logic API. For consistent comparison, both sources should use the same version number.

Example:

```
10.8.768
```

18.3 Release Date

The release date shown in the firmware list comes from the file name:

```
20260926_081451
```

corresponds to:

```
26.09.2026 08:14:51
```

The app's internal compiler/build time may differ and is shown separately in the analysis area.

19. Firmware and Backup Analysis

The catalog analyzes the full image and attempts to detect, among other things, the following structures:

- Bootloader
- Partition table
- Partition table MD5
- App partitions

- ESP app descriptor
- NVM/NVS
- CLMETA
- SPIFFS
- Other data partitions

For each partition, the flash offset, size, type/subtype, and data status are determined, among other things.

A partition may be present in the layout but consist only of `0xFF` in the full image. The catalog therefore distinguishes between “partition present” and “partition contains data”.

20. RAM Stub Behavior

The manager treats the RAM stub as a separate, valid loader state.

An existing stub is detected during a later scan. In addition to the ROM bootloader, the application attempts to synchronize with the stub using the known service baud rate.

Typical display:

```
COM30
EspRamStub
RAM-Stub
```

The scanner can read chip, MAC, and CLMETA data again through the running stub. This keeps backup and firmware update functions available even after a later rescan.

The stub currently typically operates at:

```
921600 baud
```

21. ESP32-S3 Special Considerations

21.1 S3 Quad / S3 Octal

The firmware variants differ particularly because of their different PSRAM configurations. The firmware variant must therefore match the device.

S3 Quad firmware is not offered for a device detected as S3 Octal, and vice versa.

21.2 Native USB and UART

Depending on the hardware, an S3 can provide multiple interfaces, for example a native Espressif USB interface and an additional USB-UART adapter. Not every interface responds in every operating state.

A present `UsbUartBridge` with `NoResponse` therefore does not automatically indicate a hardware fault.

21.3 Starting the App from the RAM Stub

On the S3, a system reset is triggered to safely leave the stub. The native COM port may briefly disappear and the application USB interface may then reappear.

22. ESP32-P4 Special Considerations

For the P4, the display variant is decisive for firmware selection.

The manager currently distinguishes between:

- P4 7-inch / DSI7
- P4 10-inch / DSI10

Both P4 variants are deliberately shown in the firmware update window. The user must select the firmware matching the display connected to the device.

The automatic firmware status check in the main window, however, uses the already detected installed variant.

23. Troubleshooting

23.1 No Devices Found

Check:

- USB cable and power supply
- Windows Device Manager
- Required USB/UART drivers
- Whether the COM port appears in Windows
- Whether the port is opened by another application
- Run a manual scan using **Scan again**

23.2 Port Is Shown as “In Use”

Close other programs that may have opened the same COM port, for example a terminal, IDE monitor, or debugging software.

23.3 Native ESP Interface Present, but “Reset Required”

The interface is present in Windows, but does not respond to the expected loader communication. A manual reset or switching to boot mode again may be required.

23.4 Device Remains in RAM Stub After Backup

This is intentional. **Start app** starts the normal firmware. Backup and firmware update can directly reuse the existing stub.

23.5 No Matching Firmware in the Update Window

Check:

- Is the `.bin` file located in the `firmware` directory?
- Can the file be analyzed?
- Does the processor match?
- For the S3, does the Quad/Octal variant match?
- Does the file name follow the defined naming scheme?
- On an empty S3, is the hardware variant known at all?

23.6 Firmware Status Shows “Variant Unknown”

The device variant could not be determined uniquely from CLMETA or the running device information. The application deliberately does not guess the update status in this case.

23.7 Firmware Status Shows “Version Not Comparable”

The installed or available version information does not match a parseable version format. Check whether the release BIN contains a correct app version.

23.8 Firmware Catalog Shows Warnings

A warning does not necessarily mean that the file is unusable. Typical causes include:

- CLMETA present but empty
- Specific device MAC stored in a firmware CLMETA

- Target information is inconsistent
- Partition table or MD5 is unusual
- Partition present but contains no data

The exact cause is shown in the lower analysis area.

23.9 Backup Cannot Be Started

Check:

- Is the device in the bootloader or RAM stub?
- Is the COM port free?
- Was the flash size detected?
- Was the MAC address detected?

The manager refuses to start a full backup if the flash size is not known reliably.

23.10 Firmware Update Was Canceled

Do not immediately use **Start app**. The partition that was being processed last may be incomplete. Restart the firmware update and fully write the required regions.

24. Status and Control Logic Overview

Device state	Boot mode	Start app	Backup	Firmware update
CompactLogic / CLRun	✓	–	–	–
ESP ROM bootloader	–	✓*	✓	✓
ESP RAM stub	–	✓*	✓	✓
Unknown/unresolved interface	–	–	–	–
Port in use	–	–	–	–

* **Start app** is only offered if the manager assumes that known Compact Logic firmware is present.

25. Recommended Screenshot List for This Manual

The following screenshots would be useful additions to the manual:

No.	Subject	Recommended content
01	Main window	Complete device overview after a scan
02	Header area	COM Auto-Scan and Scan again
03	Device block	One physical device with multiple interfaces and history
04	Firmware status	FW up to date and FW update available after the MAC
05	Scan Log	Detailed information for an app device and a RAM stub
06	Backup	Running backup with progress and Cancel
07	Firmware catalog	List of multiple S3/P4 firmware images plus analysis area
08	Backup list	Selected backup with visible CLMETA details
09	Firmware update step 1	Multiple compatible firmware versions available for selection
10	Firmware update step 2	Partition selection with App preselected
11	Firmware update progress	Flash overlay with progress and Cancel
12	RAM stub state	Main window after backup, EspRamStub in red and Start app/Backup/Firmware update buttons

26. Quick Reference

Normal Update

Scan → Boot mode → Backup → Firmware update → select App → flash App only → Start app

Update with Storage

Scan → Boot mode → Backup → Firmware update → select App + SPIFFS → Flash → Start app

Full Programming

Bootloader → Firmware update → check firmware variant → flash all required regions → Start app

Backup Only

Boot mode/RAM stub → Backup → device remains in RAM stub → optional Start app

Appendix A – Typical Firmware Files

CLFW_ESP32-S3-Quad_20260926_081451_with_storage.bin
CLFW_ESP32-S3-Octal_20260926_081451_with_storage.bin
CLFW_ESP32-P4-Dsi7_20260926_081423_with_storage.bin
CLFW_ESP32-P4-Dsi10_20260926_081423_with_storage.bin

Appendix B – Typical Backup File

20260926_133245_E072A1E8ABE0_leer1.bin
