Report	**OTA**	**V2**
Date:	12 December 2022	
Written by:	René Weimar	
Copy to:	Kumkeo	NLC

1 Introduction

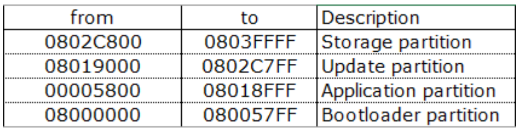
This document describes the memory layout of both coffee machine and grinder, as well as the update procedure via UART / BT using a customized bootloader. Furthermore, the requirements for the bin file holding the new image are defined.

In order to perform updates and to make sure the machine is not bricked; a bootloader is introduced and the memory of the MCU is divided into different partitions.

Note:
At this moment only the grinder embedded software and Windows program are modified for updating.
After successful testing, the coffee machine software will be updated.

2 Memory layout

In order to be able to update the application, the memory of the MCU will be divided into 4 partitions:



from	to	Description
0802C800	0803FFFF	Storage partition
08019000	0802C7FF	Update partition
00005800	08018FFF	Application partition
08000000	080057FF	Bootloader partition

1. Bootloader partition 22K
   Handles the actual update of the application
2. Application partition 78K
   Actual active application
3. Upgrade partition 78K
   Updated application to be installed
4. Storage partition 78K
   Also called RollBack partition. Used to store the current application before updating. In case the new partition is not working, the bootloader roll back this storage partition back to the application partition.

3 Bootloader partition

When the machine is powered on (Or the embedded system is rebooted) the stack pointer will always go to 0x8000000, meaning the bootloader partition is started. In case the new application cannot run and the roll back has failed the bootloader remains active.

3.1 Capabilities

The bootloader software has the following capabilities:

1. Serial communication
   The bootloader can handle a limited set of commands, listed below.
2. LED control
   The bootloader can handle the full set of LED commands. For now, the bootloader will execute its own LED settings. The Linux system will not send any LED commands.
3. Flash read and write operations
   Flash actions to copy partitions

3.2 Commands

The bootloader accepts the following commands from the Linux system:

1. SendPartitionBlock (0x21)
   Receives and saves the received partition block in the update partition
2. UpgradeApplication (0x22)
   Backs up the current application and copies the update partition to the application partition
3. GetPartitionInfo (0x23)
   Returns the status and parameters of the partitions:
   - Application version number
   - Application status
   - RollBack version number
   - RollBack status
   - Update version number
   - Update last block sent
   - Update number of blocks
   - Update status

   Note:
   The version format is uint16_t. The value is calculated with the formula:
   (MAJORVERSION << 12) + (MINORVERSION << 8) + DEBUGVERSION
   MAJORVERSION can be 0 to 15
   MINORVERSION can be 0 to 15
   DEBUGVERSION can be 0 to 255

4. SetPartitionStatus (0x26)
   Sets the partition status to the received status. Normally not used, but will remain as a fallback option.
5. RestoreApplication (0x29)
   Forces roll back of the application stored in the Storage partition.
6. SetLED (0x41)
   Sets the LED's to the received settings. Normally not used, but will remain for potential future use.

Note:
These commands are also available in the application. Normally all commands to copy and upload images are used in the active application. Here they only serve as a fall-back option, in case the application and the roll back cannot start anymore and only the bootloader remains accessible.

3.3 Return messages

The bootloader sends messages during startup and during the upgrade command:

1. GetBootLoaderInfo (0x24)
   Returns the progress of the upgrade in CurrentBlock and NrOfBlocks
2. GetBootLoaderStatus (0x25)
   Returns the bootloader status.
3. GetBackupStatus (0x27)
   Returns the backup progress in percent complete.
4. GetRestoreStatus (0x28)
   Returns the restore (roll back) progress in percent complete.

Note:
These messages are sent without being requested. The data is used in an interrupt capable system (Like the Windows program) to display the progress. The Linux system cannot handle interrupts; therefore, the messages will only be sent to the debug port on the embedded boards and not to the Linux port.

4 Upgrade procedure

The upgrade can only be done after a valid image has been sent to the embedded system. If a valid image is present the upgrade command is sent to the embedded system, which then will performing the upgrade.

4.1 Creating a valid update partition

In order to perform an upgrade, a valid update partition needs to be present in the embedded system’s flash memory. The new update image is an exported bin file of the embedded software with a physical location of 0x8005800.
The name of the image should be:

- xxyyzzzzG.bin for the grinder embedded
- xxyyzzzzC.bin for the coffee machine embedded

where:
xx = Major version (0-15)
yy = Minor version (0-15)
zzzz = Debug version (0-255)

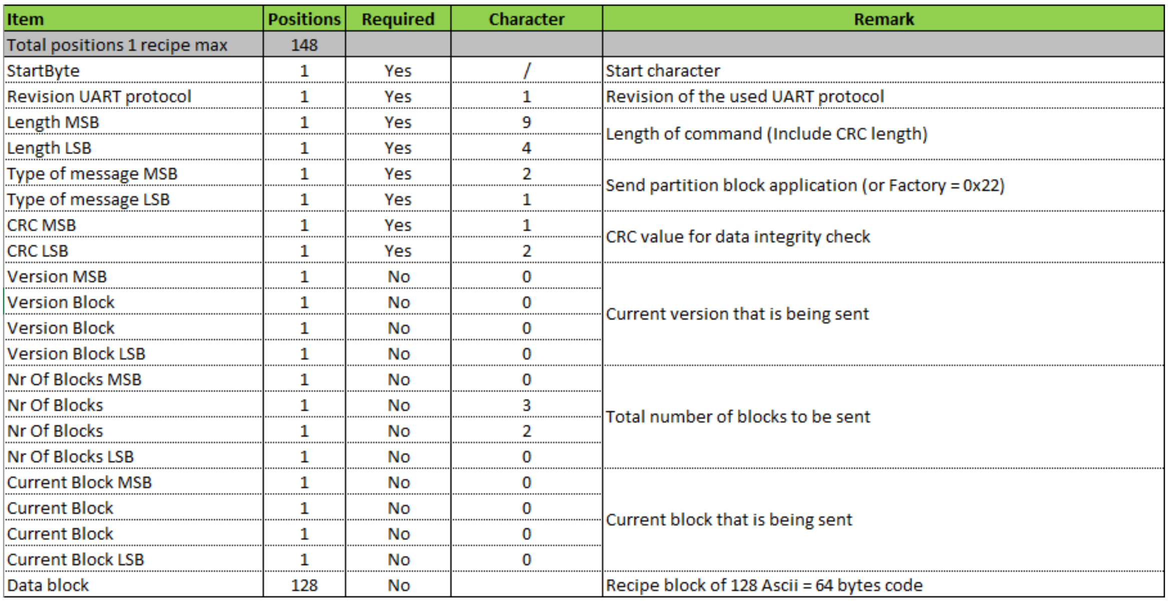
The name of the bin file needs to be consistent with the actual software version. It is used in order to determine whether the new image is a lower, higher or equal version than the already running application.

The Linux or Windows system can determine whether it is ok to downgrade or not. It can also check whether the new image is already running and therefore no update is needed. The Windows application will only warn for equal or lower version update.

The bin file is loaded by the Linux system (or by the Windows application) and sent to the embedded system using the following steps:

1. The bin file is divided into blocks of 64 bytes, (So 128 Hex characters)
   Note:
   If the last block is not 64 bytes, 0's will be added to the end until it is, so the string to send is again 128 hex characters.

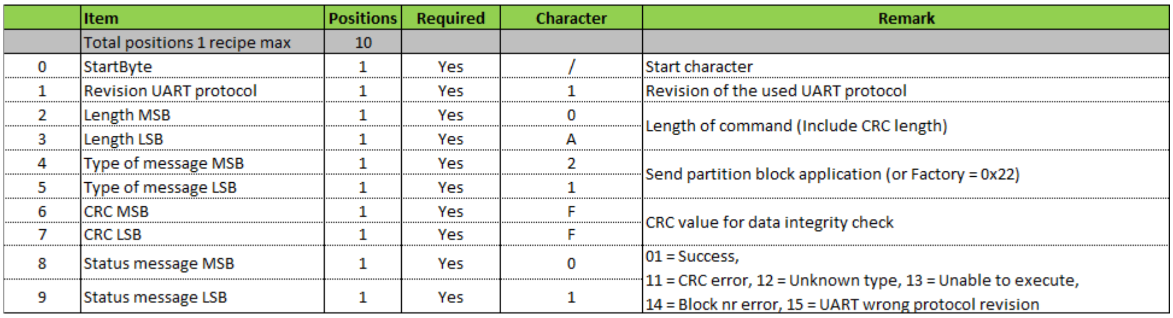
2. Each block is sent in ascending order to the embedded system, using the "SendPartitionBlock" command with below syntax:



Item	Positions	Required	Character	Remark
Total positions 1 recipe max	148			
StartByte	1	Yes	/	Start character
Revision UART protocol	1	Yes	1	Revision of the used UART protocol
Length MSB	1	Yes	9	Length of command (Include CRC length)
Length LSB	1	Yes	4	
Type of message MSB	1	Yes	2	Send partition block application (or Factory = 0x22)
Type of message LSB	1	Yes	1	
CRC MSB	1	Yes	1	CRC value for data integrity check
CRC LSB	1	Yes	2	
Version MSB	1	No	0	Current version that is being sent
Version Block	1	No	0	
Version Block	1	No	0	
Version Block LSB	1	No	0	
Nr Of Blocks MSB	1	No	0	Total number of blocks to be sent
Nr Of Blocks	1	No	3	
Nr Of Blocks	1	No	2	
Nr Of Blocks LSB	1	No	0	
Current Block MSB	1	No	0	Current block that is being sent
Current Block	1	No	0	
Current Block	1	No	0	
Current Block LSB	1	No	0	
Data block	128	No		Recipe block of 128 Ascii = 64 bytes code

   Note:
   When the first block is sent and verified ok by the embedded system, the update partition status is set to UPDATING because it is not VALID anymore until the total new UPDATE is copied.

3. After the block is received, the embedded system programs the data into the corresponding flash memory, verifies the written data and sends the return message to the Linux system according to following syntax:



	Item	Positions	Required	Character	Remark
	Total positions 1 recipe max	10			
0	StartByte	1	Yes	/	Start character
1	Revision UART protocol	1	Yes	1	Revision of the used UART protocol
2	Length MSB	1	Yes	0	Length of command (Include CRC length)
3	Length LSB	1	Yes	A	
4	Type of message MSB	1	Yes	2	Send partition block application (or Factory = 0x22)
5	Type of message LSB	1	Yes	1	
6	CRC MSB	1	Yes	F	CRC value for data integrity check
7	CRC LSB	1	Yes	F	
8	Status message MSB	1	Yes	0	01 = Success, 11 = CRC error, 12 = Unknown type, 13 = Unable to execute, 14 = Block nr error, 15 = UART wrong protocol revision
9	Status message LSB	1	Yes	1	

   The Linux system waits for the return message to be received, checks if it was received OK and sends the next partition block until all partition blocks are sent.

   The embedded system uses a structure named FLA, in order to keep track of the various partition statuses and progress.

Every time a block is received by the embedded system, the UpdateVersion inside the header of the block is compared with FLA.UpDateVersion, in order to make sure the data matches the older data in the partition (Needed when copying the data was interrupted and needs to be restarted):
- If it is the same, it will be processed.
- If it is different, but its number is 0, it will be processed. FLA.UpgradeVersion will be updated
- If none of the above, it will not be processed. The return status will be 0x19 (Wrong version sent)

Note:
If the Linux system sends a wrong block number, the block will not be processed and the return status will be 0x14 (Block nr. Error). If this happens, or if the received data is not verified ok, the Linux system can decide to resend it, or abort the sending.

4. When the last block has been sent, received and verified ok, the embedded system will set FLA.UpdateStatus to VALID.

4.2 Performing the upgrade

The actual upgrade is started by sending the command: /10822CB
When this command is received by the embedded system following will be done:

1. FLA.ApplicationStatus will be changed to UPDATING
   This acts as a flag for the bootloader indicating it to perform the upgrade at the next reboot.
2. Reboot the system

After the reboot, the bootloader is active. It sees that the FLA.ApplicationStatus set to UPDATING and will perform following actions:

1. Check whether FLA.UpdateStatus equals VALID.
   If not, the upgrade will not be performed, the FLA.ApplicationStatus will be reset to VALID and the old application is started.
2. If FLA.UpdateStatus is VALID, the current application partition will be backed up in the Storage partition.
   If this failed, the upgrade will not be performed. The Storage partition status will be set to CORRUPT with version 0. FLA.ApplicationStatus will be reset to VALID and the old application is started.
3. If the backup succeeded, the update image will be copied to the application partition
   If this failed, the application is corrupt and will be rolled back.
4. If the copy of the application succeeded FLA.ApplicationStatus will be set to UPDATED and the new application will be started.
   If the new application crashes, it will reboot the system and the bootloader can see that FLA.ApplicationStatus still is UPDATED. In that case the bootloader will roll back the old application.
5. If all goes well, the application can start and the new software update works. Inside the new application FLA.ApplicationStatus will be set to VALID in that case.

During the copying, the bootloader sends the progress via the debug UART that is used by the Windows program only.

Note:
We need to make sure that UART works, so setting the application status to VALID will be done only after a UART message has been received in the future.

4.3 Failsafe

If the application doesn’t work, an independent watchdog timer will reboot the system after 10 seconds.
In the unlikely case that the application only partially works after the update, there are two possibilities:

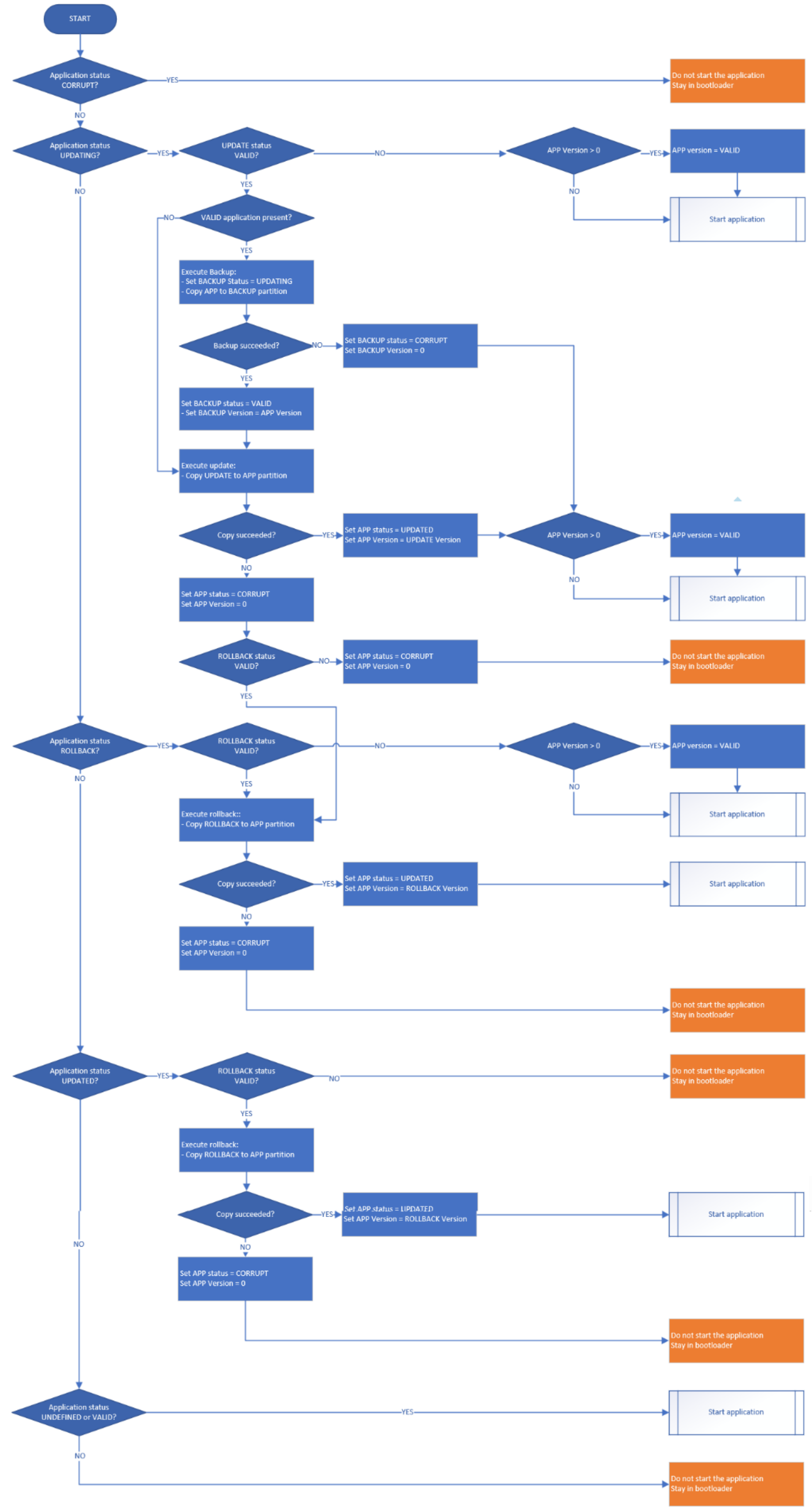
1. UART communication is still possible
   FLA.ApplicationStatus has been set to VALID
2. UART communication is not possible
   FLA.ApplicationStatus is still UPDATING

In the first case, a new image can be sent and the process can be repeated or a rollback can be forced by setting FLA.ApplicationStatus to ROLLBACK

In the second case, the user needs to be asked to reboot the machine by switching it off and on again.
After the reboot, the bootloader is active and the bootloader will retry or roll back the old application

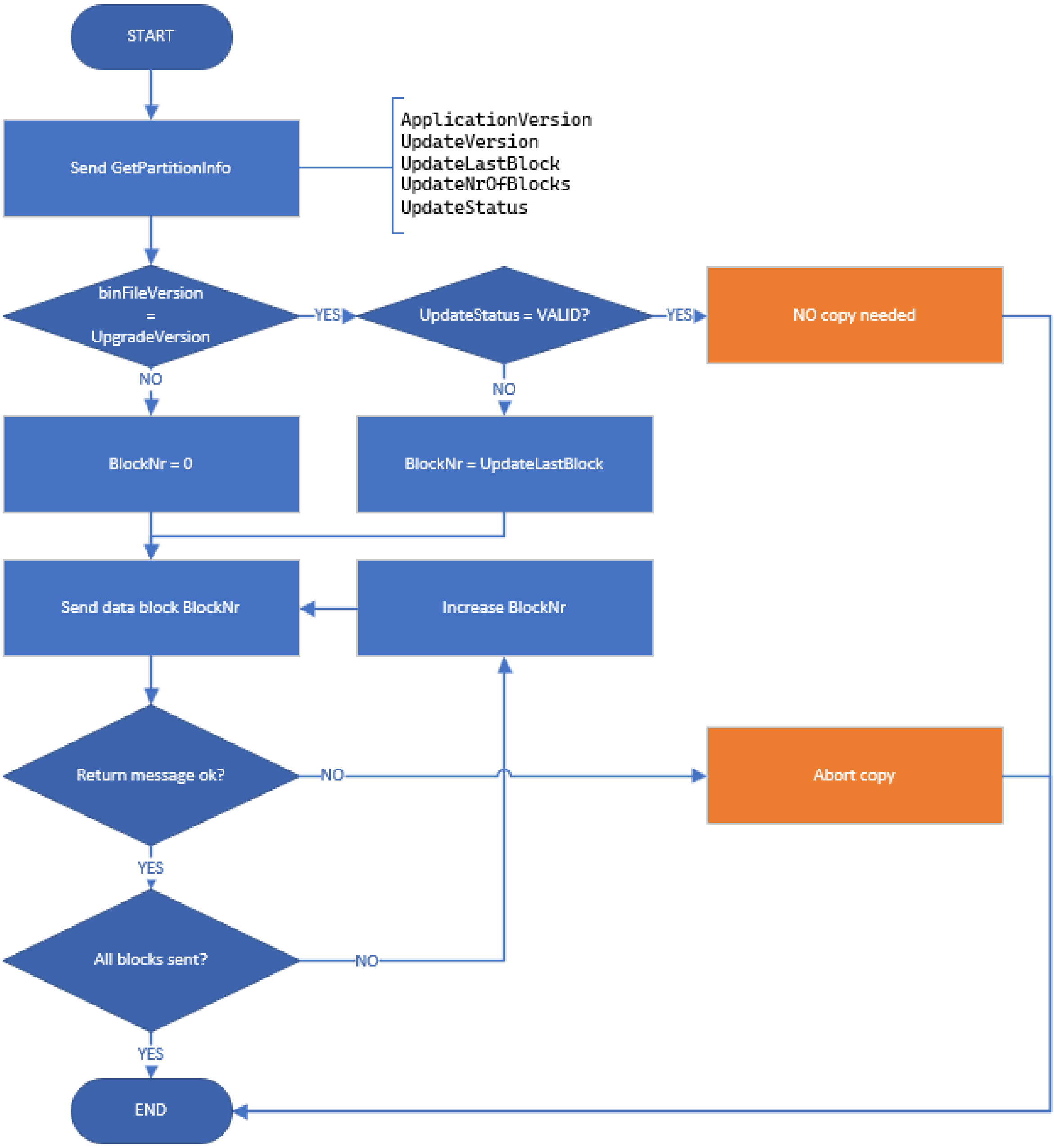
5 Appendix A: Bootloader

Below flowchart represents the flow of the bootloader process:


START
Application status CORRUPT?
YES
Do not start the application Stay in bootloader
NO
Application status UPDATING?
YES
UPDATE status VALID?
NO
APP Version > 0
YES
APP version = VALID
NO
Start application
YES
VALID application present?
NO
YES
Execute Backup: - Set BACKUP Status = UPDATING - Copy APP to BACKUP partition
Backup succeeded?
NO
Set BACKUP status = CORRUPT Set BACKUP Version = 0
YES
Set BACKUP status = VALID - Set BACKUP Version = APP Version
Execute update: - Copy UPDATE to APP partition
Copy succeeded?
YES
Set APP status = UPDATED Set APP Version = UPDATE Version
APP Version > 0
YES
APP version = VALID
NO
Start application
NO
Set APP status = CORRUPT Set APP Version = 0
ROLLBACK status VALID?
NO
Set APP status = CORRUPT Set APP Version = 0
Do not start the application Stay in bootloader
YES
Application status ROLLBACK?
YES
ROLLBACK status VALID?
NO
APP Version > 0
YES
APP version = VALID
NO
Start application
YES
Execute rollback:: - Copy ROLLBACK to APP partition
Copy succeeded?
YES
Set APP status = UPDATED Set APP Version = ROLLBACK Version
Start application
NO
Set APP status = CORRUPT Set APP Version = 0
Do not start the application Stay in bootloader
NO
Application status UPDATED?
YES
ROLLBACK status VALID?
NO
Do not start the application Stay in bootloader
YES
Execute rollback: - Copy ROLLBACK to APP partition
Copy succeeded?
YES
Set APP status = UPDATED Set APP Version = ROLLBACK Version
Start application
NO
Set APP status = CORRUPT Set APP Version = 0
Do not start the application Stay in bootloader
NO
Application status UNDEFINED or VALID?
YES
Start application
NO
Do not start the application Stay in bootloader

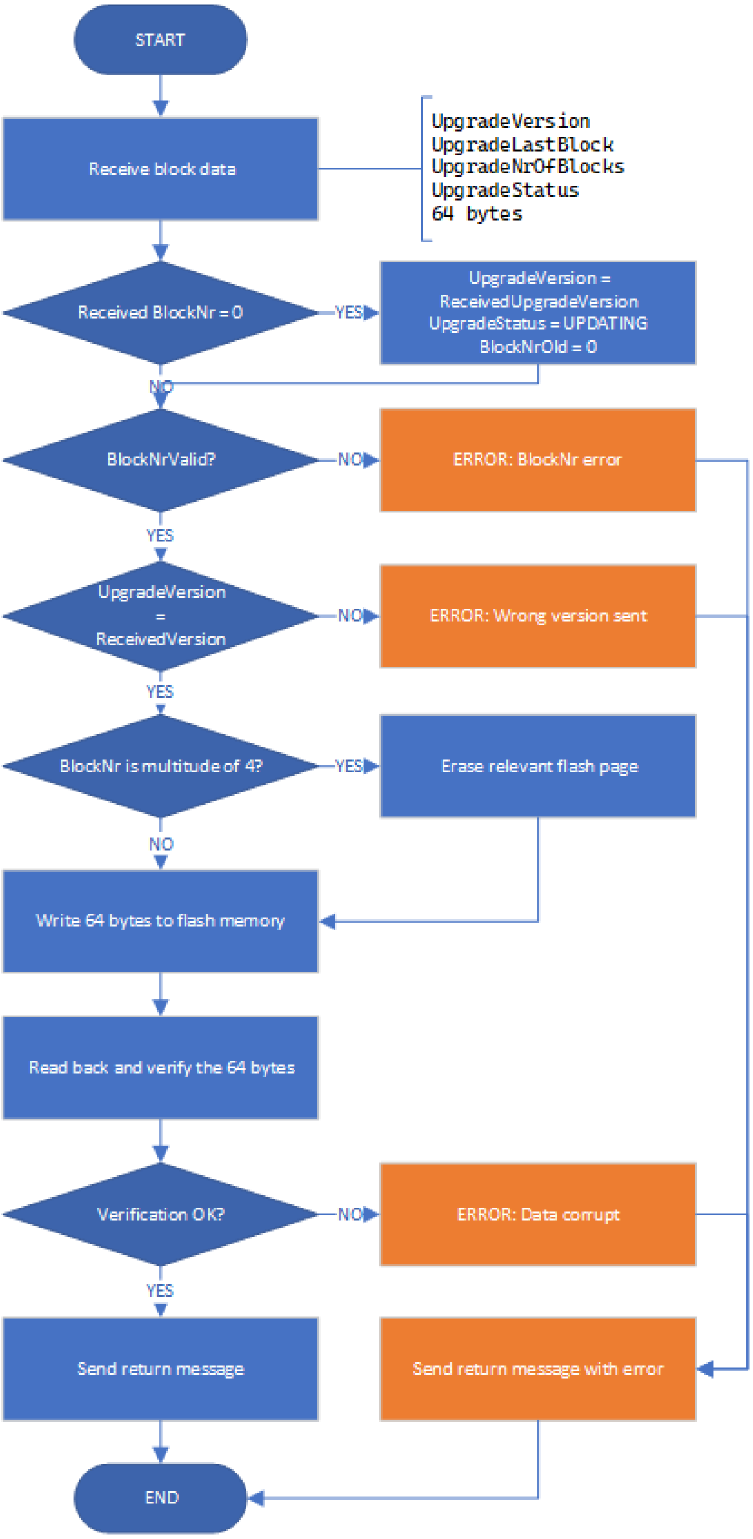
6 Appendix B: Sending from Linux or Windows

Below flowchart represents the flow of sending an image from the Linux or Windows side



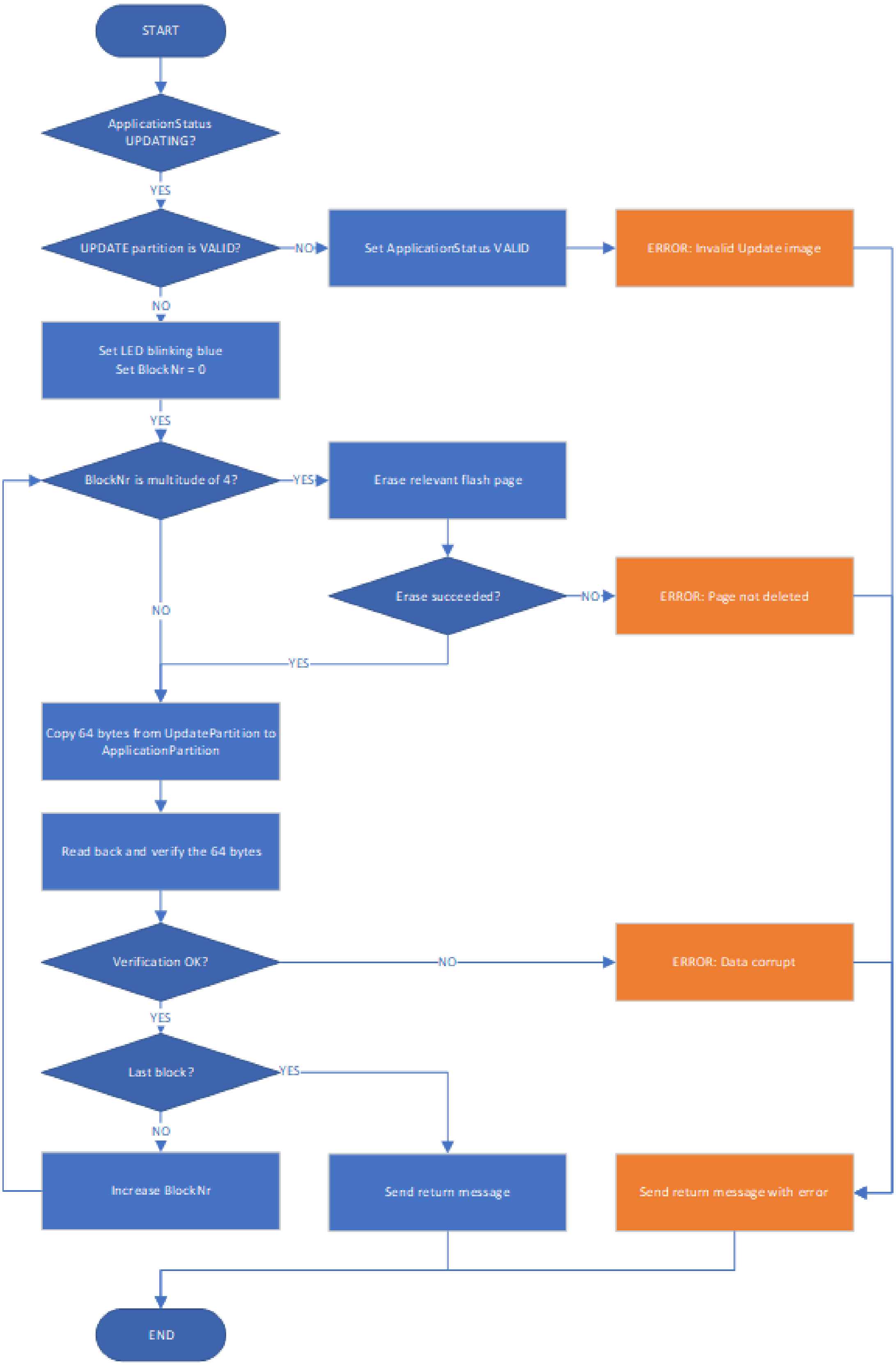
7 Appendix C: Receiving in embedded

Below flowchart represent the flow of receiving an image on the embedded side



8 Appendix D: Upgrading the application

Below flowchart represent the flow of the upgrade of the application image


START
ApplicationStatus UPDATING?
YES
UPDATE partition is VALID?
NO
Set ApplicationStatus VALID
ERROR: Invalid Update image
NO
Set LED blinking blue
Set BlockNr = 0
YES
BlockNr is multitude of 4?
YES
Erase relevant flash page
Erase succeeded?
NO
ERROR: Page not deleted
NO
YES
Copy 64 bytes from UpdatePartition to ApplicationPartition
Read back and verify the 64 bytes
Verification OK?
NO
ERROR: Data corrupt
YES
Last block?
YES
NO
Increase BlockNr
Send return message
Send return message with error
END

9 Appendix D: Windows program

Below screens show the process as done with the Windows program.
The Windows program version needs to be bigger than or equal to 2.0.2.

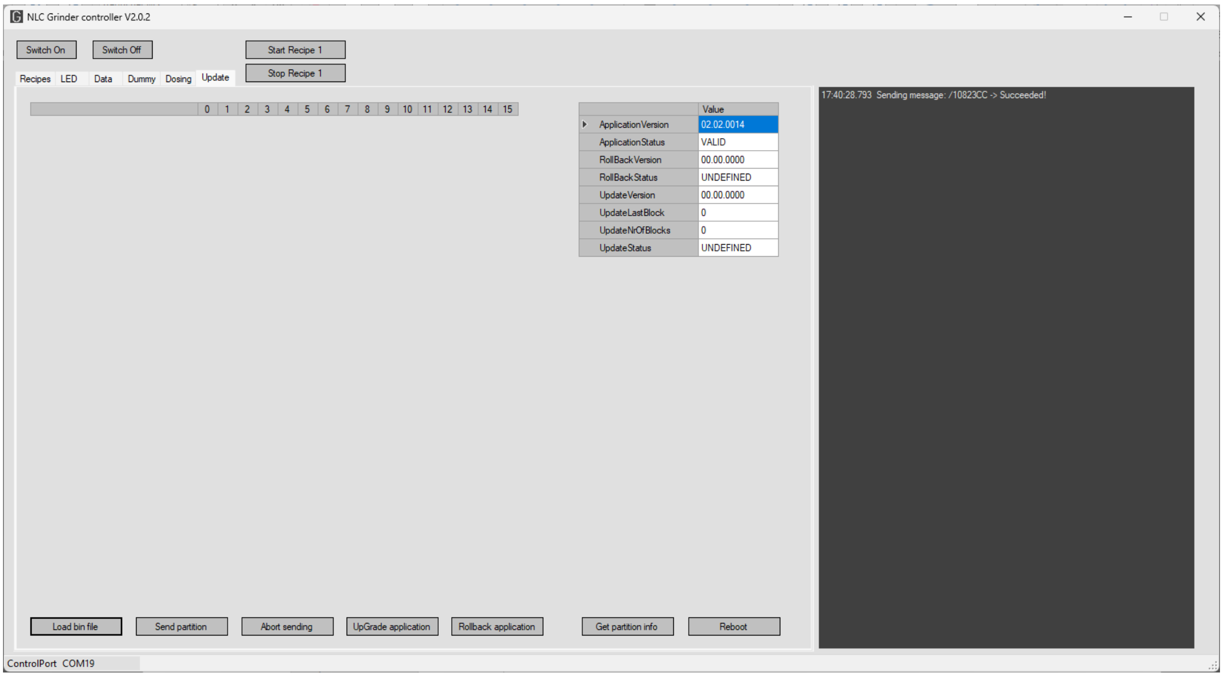
9.1 First setup

In order to convert from the previous versions that didn't have OTA, following steps need to be followed:

1. Erase the whole MCU
   Just to make sure no remaining software is left.
2. Upload the package "80482201 - Grinder V02.02.0014 & Bootloader V02.02.0001.hex"
   This package contains:
   - Bootloader
   - Grinder software V02.02.0014
   - Eeprom memory with application version and status info
3. Install the Windows program V2.0.2
   This program is suitable to use all new functions.

9.2 Main screen

The Windows program now has a tab called "Update". Here, all actions to perform an update can be done.



	Value
ApplicationVersion	02.02.0014
ApplicationStatus	VALID
RollBackVersion	00.00.0000
RollBackStatus	UNDEFINED
UpdateVersion	00.00.0000
UpdateLastBlock	0
UpdateNrOfBlocks	0
UpdateStatus	UNDEFINED

```
17:40:28.793  Sending message: /10823CC -> Succeeded!
```

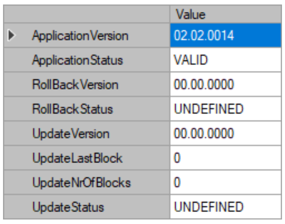
ControlPort COM19

After pressing Get partition info in the tab Update, the partition information is shown.
As can be seen in the partition information, the APP V02.02.0014 is installed and VALID.

9.3 Perform the upgrade

9.3.1 Check the partitions

By pressing Get partition info, the table will be filled.

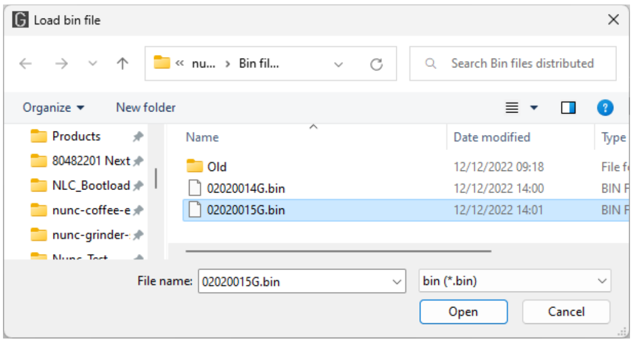


	Value
ApplicationVersion	02.02.0014
ApplicationStatus	VALID
RollBackVersion	00.00.0000
RollBackStatus	UNDEFINED
UpdateVersion	00.00.0000
UpdateLastBlock	0
UpdateNrOfBlocks	0
UpdateStatus	UNDEFINED

We see that the current application is version 02.02.0014 and is VALID.
The update and roll back status are UNDEFINED, meaning they are not available.

9.3.2 Load a new bin file

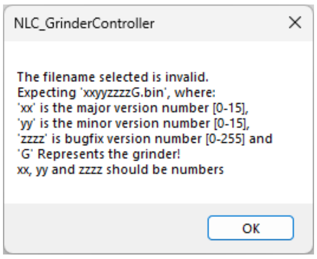
By pressing Load bin file, a dialog box will appear. Here the new bin file can be selected.



Name	Date modified	Type
Old	12/12/2022 09:18	File f
02020014G.bin	12/12/2022 14:00	BIN F
02020015G.bin	12/12/2022 14:01	BIN F

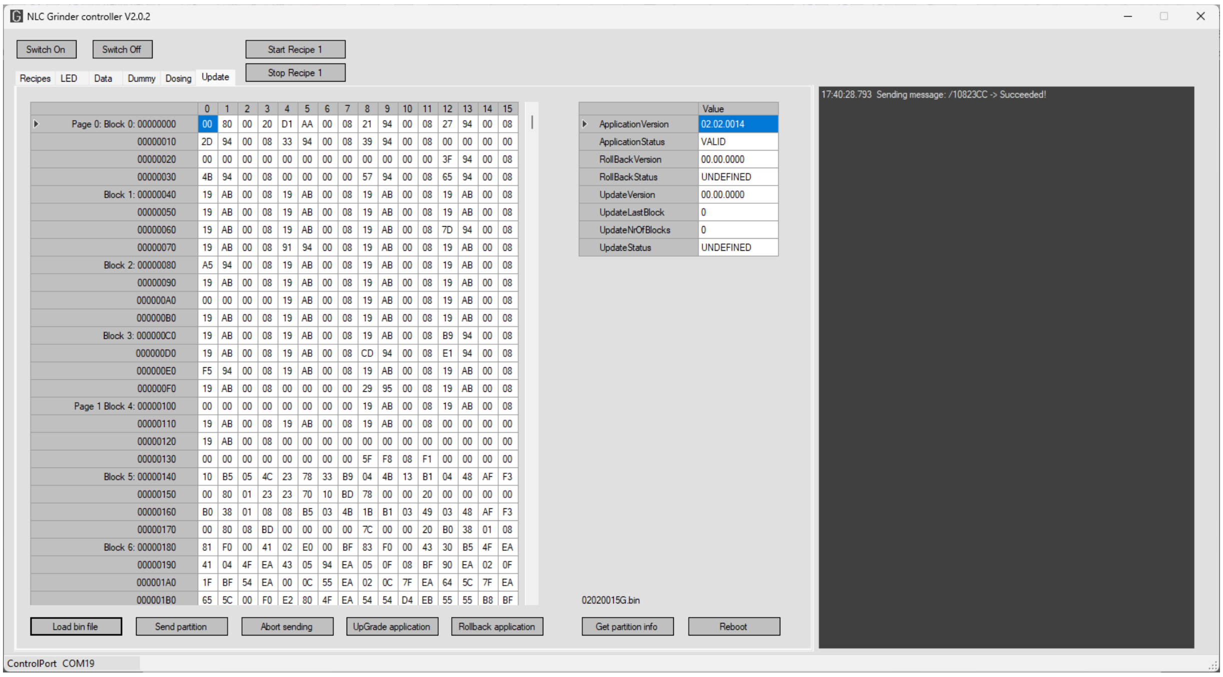
File name: 02020015G.bin
bin (*.bin)

Note:
The BIN file should be of the correct syntax. Otherwise below message will appear and the bin file will not be loaded:



The filename selected is invalid.
Expecting 'xxyyzzzzG.bin', where:
'xx' is the major version number [0-15],
'yy' is the minor version number [0-15],
'zzzz' is bugfix version number [0-255] and
'G' Represents the grinder!
xx, yy and zzzz should be numbers

Upon success, the bin file contents and the version of the loaded file will be shown in the update tab:



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Page 0: Block 0: 00000000	00	80	00	20	D1	AA	00	08	21	94	00	08	27	94	00	08
00000010	2D	94	00	08	33	94	00	08	39	94	00	08	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	3F	94	00	08
00000030	4B	94	00	08	00	00	00	00	57	94	00	08	65	94	00	08
Block 1: 00000040	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
00000050	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
00000060	19	AB	00	08	19	AB	00	08	19	AB	00	08	7D	94	00	08
00000070	19	AB	00	08	91	94	00	08	19	AB	00	08	19	AB	00	08
Block 2: 00000080	A5	94	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
00000090	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
000000A0	00	00	00	00	19	AB	00	08	19	AB	00	08	19	AB	00	08
000000B0	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
Block 3: 000000C0	19	AB	00	08	19	AB	00	08	19	AB	00	08	B9	94	00	08
000000D0	19	AB	00	08	19	AB	00	08	CD	94	00	08	E1	94	00	08
000000E0	F5	94	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
000000F0	19	AB	00	08	00	00	00	00	29	95	00	08	19	AB	00	08
Page 1 Block 4: 00000100	00	00	00	00	00	00	00	00	19	AB	00	08	19	AB	00	08
00000110	19	AB	00	08	19	AB	00	08	19	AB	00	08	00	00	00	00
00000120	19	AB	00	08	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00	5F	F8	08	F1	00	00	00	00
Block 5: 00000140	10	B5	05	4C	23	78	33	B9	04	4B	13	B1	04	48	AF	F3
00000150	00	80	01	23	23	70	10	BD	78	00	00	20	00	00	00	00
00000160	B0	38	01	08	08	B5	03	4B	1B	B1	03	49	03	48	AF	F3
00000170	00	80	08	BD	00	00	00	00	7C	00	00	20	B0	38	01	08
Block 6: 00000180	81	F0	00	41	02	E0	00	BF	83	F0	00	43	30	B5	4F	EA
00000190	41	04	4F	EA	43	05	94	EA	05	0F	08	BF	90	EA	02	0F
000001A0	1F	BF	54	EA	00	0C	55	EA	02	0C	7F	EA	64	5C	7F	EA
000001B0	65	5C	00	F0	E2	80	4F	EA	54	54	D4	EB	55	55	B8	BF

	Value
ApplicationVersion	02.02.0014
ApplicationStatus	VALID
RollBackVersion	00.00.0000
RollBackStatus	UNDEFINED
UpdateVersion	00.00.0000
UpdateLastBlock	0
UpdateNrOfBlocks	0
UpdateStatus	UNDEFINED

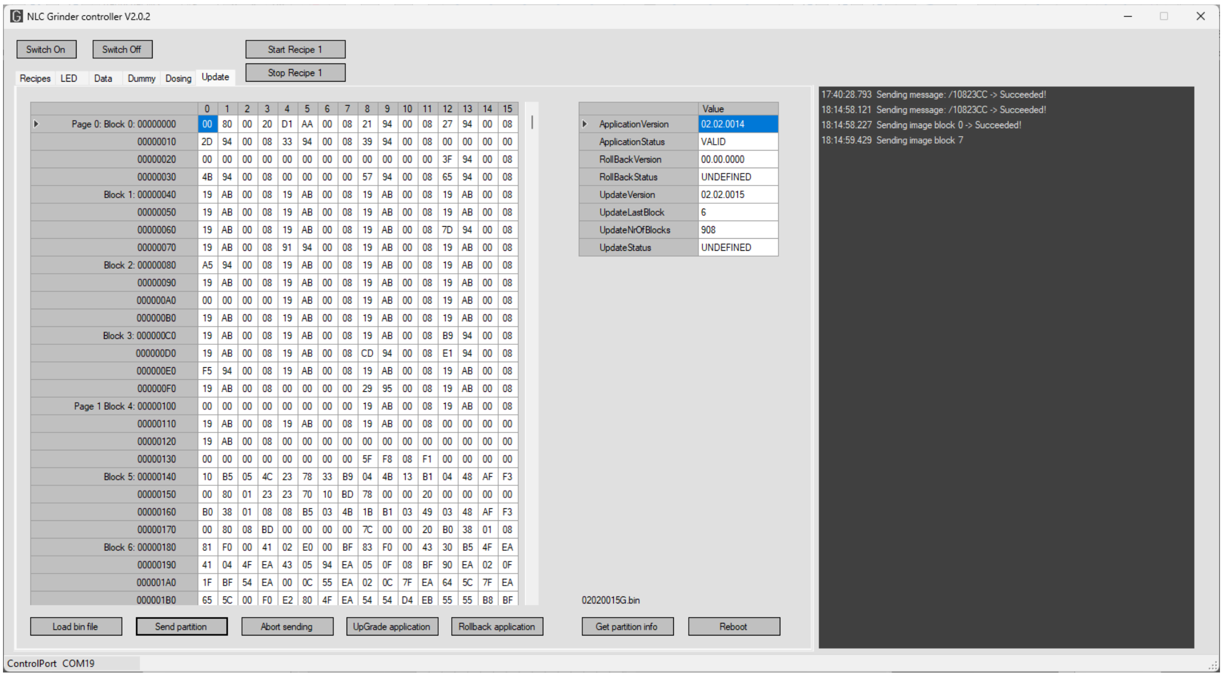
02020015G.bin

```
17:40:28.793  Sending message: /10823CC -> Succeeded!
```

ControlPort COM19

9.3.3 Sending the update

By pressing Send partition, the image will be sent to and stored in the embedded system.
The UpdateStatus will be set to UPDATING by the embedded system internally after the first block has been copied and verified successfully (This will not be shown in the Windows program). The progress can be seen in the log window:



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Page 0: Block 0: 00000000	00	80	00	20	D1	AA	00	08	21	94	00	08	27	94	00	08
00000010	2D	94	00	08	33	94	00	08	39	94	00	08	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	3F	94	00	08
00000030	4B	94	00	08	00	00	00	00	57	94	00	08	65	94	00	08
Block 1: 00000040	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
00000050	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
00000060	19	AB	00	08	19	AB	00	08	19	AB	00	08	7D	94	00	08
00000070	19	AB	00	08	91	94	00	08	19	AB	00	08	19	AB	00	08
Block 2: 00000080	A5	94	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
00000090	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
000000A0	00	00	00	00	19	AB	00	08	19	AB	00	08	19	AB	00	08
000000B0	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
Block 3: 000000C0	19	AB	00	08	19	AB	00	08	19	AB	00	08	B9	94	00	08
000000D0	19	AB	00	08	19	AB	00	08	CD	94	00	08	E1	94	00	08
000000E0	F5	94	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
000000F0	19	AB	00	08	00	00	00	00	29	95	00	08	19	AB	00	08
Page 1 Block 4: 00000100	00	00	00	00	00	00	00	00	19	AB	00	08	19	AB	00	08
00000110	19	AB	00	08	19	AB	00	08	19	AB	00	08	00	00	00	00
00000120	19	AB	00	08	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00	5F	F8	08	F1	00	00	00	00
Block 5: 00000140	10	B5	05	4C	23	78	33	B9	04	4B	13	B1	04	48	AF	F3
00000150	00	80	01	23	23	70	10	BD	78	00	00	20	00	00	00	00
00000160	B0	38	01	08	08	B5	03	4B	1B	B1	03	49	03	48	AF	F3
00000170	00	80	08	BD	00	00	00	00	7C	00	00	20	B0	38	01	08
Block 6: 00000180	81	F0	00	41	02	E0	00	BF	83	F0	00	43	30	B5	4F	EA
00000190	41	04	4F	EA	43	05	94	EA	05	0F	08	BF	90	EA	02	0F
000001A0	1F	BF	54	EA	00	0C	55	EA	02	0C	7F	EA	64	5C	7F	EA
000001B0	65	5C	00	F0	E2	80	4F	EA	54	54	D4	EB	55	55	B8	BF

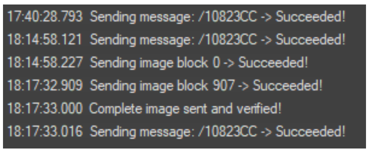
	Value
ApplicationVersion	02.02.0014
ApplicationStatus	VALID
RollBackVersion	00.00.0000
RollBackStatus	UNDEFINED
UpdateVersion	02.02.0015
UpdateLastBlock	6
UpdateNrOfBlocks	908
UpdateStatus	UNDEFINED

02020015G.bin

```
17:40:28.793  Sending message: /10823CC -> Succeeded!
18:14:58.121  Sending message: /10823CC -> Succeeded!
18:14:58.227  Sending image block 0 -> Succeeded!
18:14:59.429  Sending image block 7
```

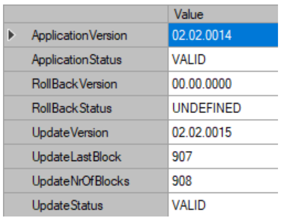
ControlPort COM19

The copy process takes around 2.5 minutes. It should be done in the application when it is in STANDBY, or in the bootloader at any time. At the end of the copy process, the UpdateStatus will be set to VALID and the Update version to the bin file version if verification is ok.



```
17:40:28.793  Sending message: /10823CC -> Succeeded!
18:14:58.121  Sending message: /10823CC -> Succeeded!
18:14:58.227  Sending image block 0 -> Succeeded!
18:17:32.909  Sending image block 907 -> Succeeded!
18:17:33.000  Complete image sent and verified!
18:17:33.016  Sending message: /10823CC -> Succeeded!
```

The table with partition info is updated:



	Value
ApplicationVersion	02.02.0014
ApplicationStatus	VALID
RollBackVersion	00.00.0000
RollBackStatus	UNDEFINED
UpdateVersion	02.02.0015
UpdateLastBlock	907
UpdateNrOfBlocks	908
UpdateStatus	VALID

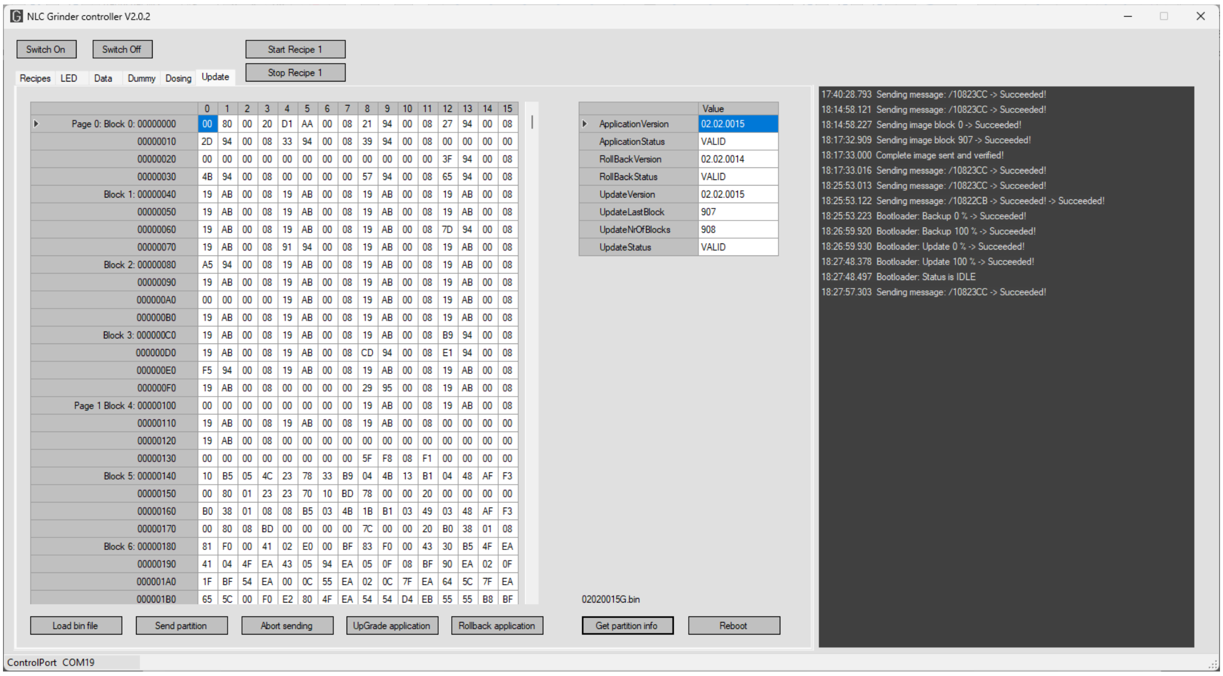
We see that the current application is still version 02.02.0014 and still is VALID.
The update version now is 02.02.0015 and VALID, so this means the new update image partition is valid and holds the new application and the upgrade can be started. Note that the RollBack status still is UNDEFINED.

9.3.4 Starting the upgrade

By pressing UpGrade application, the actual upgrade will start.
In the background, the embedded system will set the ApplicationStatus to UPDATING and reboots itself.
After rebooting, the bootloader takes over and checks and starts the upgrade which consists of the following steps:

1. The current application is copied to the RollBack partition.
   If this fails, the upgrade stops and the old application will be started again.
2. If the copy to RollBack succeeded, the Update will be copied to the Application partition.
   If this fails, the old application is rolled back and will be started again.
3. If the copy of the update succeeded, the new application will be started.

The progress can be seen in the log window:



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Page 0: Block 0: 00000000	00	80	00	20	D1	AA	00	08	21	94	00	08	27	94	00	08
00000010	2D	94	00	08	33	94	00	08	39	94	00	08	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	3F	94	00	08
00000030	4B	94	00	08	00	00	00	00	57	94	00	08	65	94	00	08
Block 1: 00000040	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
00000050	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
00000060	19	AB	00	08	19	AB	00	08	19	AB	00	08	7D	94	00	08
00000070	19	AB	00	08	91	94	00	08	19	AB	00	08	19	AB	00	08
Block 2: 00000080	A5	94	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
00000090	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
000000A0	00	00	00	00	19	AB	00	08	19	AB	00	08	19	AB	00	08
000000B0	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
Block 3: 000000C0	19	AB	00	08	19	AB	00	08	19	AB	00	08	B9	94	00	08
000000D0	19	AB	00	08	19	AB	00	08	CD	94	00	08	E1	94	00	08
000000E0	F5	94	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
000000F0	19	AB	00	08	00	00	00	00	29	95	00	08	19	AB	00	08
Page 1 Block 4: 00000100	00	00	00	00	00	00	00	00	19	AB	00	08	19	AB	00	08
00000110	19	AB	00	08	19	AB	00	08	19	AB	00	08	00	00	00	00
00000120	19	AB	00	08	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00	5F	F8	08	F1	00	00	00	00
Block 5: 00000140	10	B5	05	4C	23	78	33	B9	04	4B	13	B1	04	48	AF	F3
00000150	00	80	01	23	23	70	10	BD	78	00	00	20	00	00	00	00
00000160	B0	38	01	08	08	B5	03	4B	1B	B1	03	49	03	48	AF	F3
00000170	00	80	08	BD	00	00	00	00	7C	00	00	20	B0	38	01	08
Block 6: 00000180	81	F0	00	41	02	E0	00	BF	83	F0	00	43	30	B5	4F	EA
00000190	41	04	4F	EA	43	05	94	EA	05	0F	08	BF	90	EA	02	0F
000001A0	1F	BF	54	EA	00	0C	55	EA	02	0C	7F	EA	64	5C	7F	EA
000001B0	65	5C	00	F0	E2	80	4F	EA	54	54	D4	EB	55	55	B8	BF

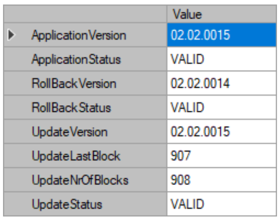
	Value
ApplicationVersion	02.02.0015
ApplicationStatus	VALID
RollBackVersion	02.02.0014
RollBackStatus	VALID
UpdateVersion	02.02.0015
UpdateLastBlock	907
UpdateNrOfBlocks	908
UpdateStatus	VALID

02020015G.bin

```
17:40:28.793  Sending message: /10823CC -> Succeeded!
18:14:58.121  Sending message: /10823CC -> Succeeded!
18:14:58.227  Sending image block 0 -> Succeeded!
18:17:32.909  Sending image block 907 -> Succeeded!
18:17:33.000  Complete image sent and verified!
18:17:33.016  Sending message: /10823CC -> Succeeded!
18:25:53.013  Sending message: /10823CC -> Succeeded!
18:25:53.122  Sending message: /10822CB -> Succeeded! -> Succeeded!
18:25:53.223  Bootloader: Backup 0 % -> Succeeded!
18:26:59.920  Bootloader: Backup 100 % -> Succeeded!
18:26:59.930  Bootloader: Update 0 % -> Succeeded!
18:27:48.378  Bootloader: Update 100 % -> Succeeded!
18:27:48.497  Bootloader: Status is IDLE
18:27:57.303  Sending message: /10823CC -> Succeeded!
```

The backup process and the update process both take a little over 1 minute.

By pressing Get partition info we get the latest status.



	Value
ApplicationVersion	02.02.0015
ApplicationStatus	VALID
RollBackVersion	02.02.0014
RollBackStatus	VALID
UpdateVersion	02.02.0015
UpdateLastBlock	907
UpdateNrOfBlocks	908
UpdateStatus	VALID

We see that the current application is now version 02.02.0015 and is VALID, meaning the update succeed and the application is running. The RollBack version is now 02.02.0014 and VALID. The update version is 02.02.0015 and VALID.

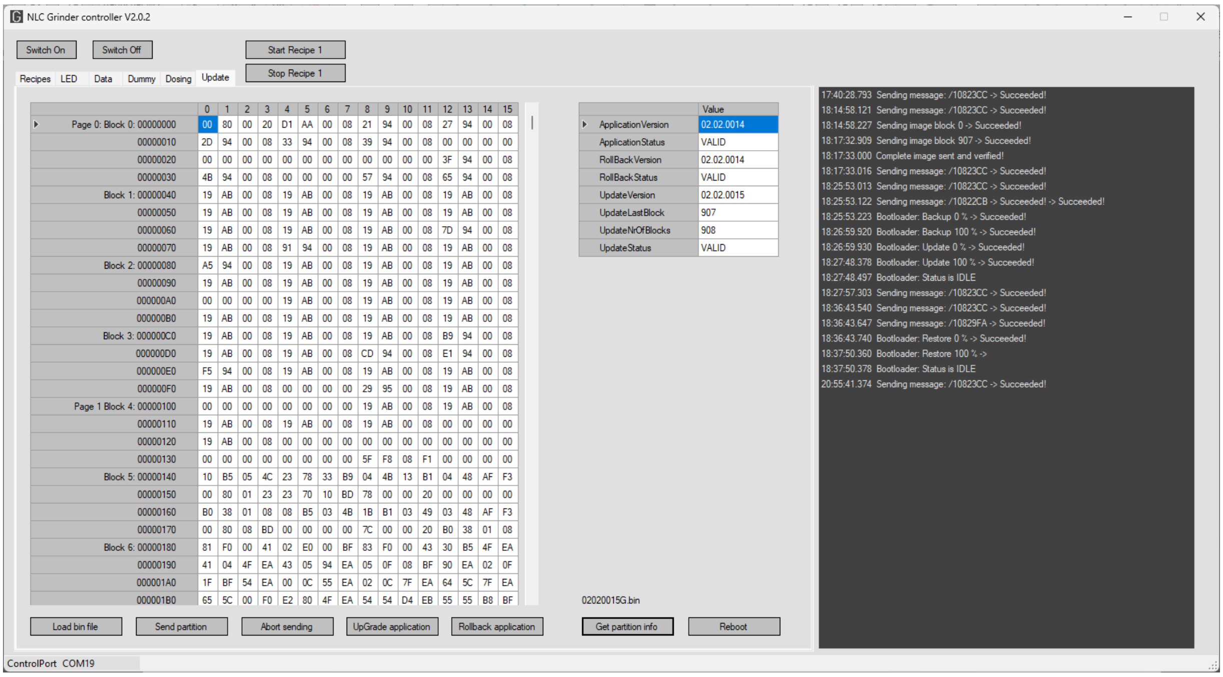
9.3.5 Starting the Roll back

If the application is technically working and VALID, but not functioning as expected, it can be rolled back by pressing Rollback application.

The embedded system will follow the following steps:

1. Check if a valid Roll Back partition is available.
   If not, the RollBack will be canceled and the old application will be started again.
2. If the RollBack partition is VALID, the Rollback partition will be copied over the application.
   If this fails, the application is CORRUPT and the bootloader will remain active.
   In this case, the CM4 can retry the RollBack, or send a new UPDATE.
3. If this succeeds, the Application is rolled back and started.

The progress can be seen in the log window:



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Page 0: Block 0: 00000000	00	80	00	20	D1	AA	00	08	21	94	00	08	27	94	00	08
00000010	2D	94	00	08	33	94	00	08	39	94	00	08	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	3F	94	00	08
00000030	4B	94	00	08	00	00	00	00	57	94	00	08	65	94	00	08
Block 1: 00000040	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
00000050	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
00000060	19	AB	00	08	19	AB	00	08	19	AB	00	08	7D	94	00	08
00000070	19	AB	00	08	91	94	00	08	19	AB	00	08	19	AB	00	08
Block 2: 00000080	A5	94	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
00000090	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
000000A0	00	00	00	00	19	AB	00	08	19	AB	00	08	19	AB	00	08
000000B0	19	AB	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
Block 3: 000000C0	19	AB	00	08	19	AB	00	08	19	AB	00	08	B9	94	00	08
000000D0	19	AB	00	08	19	AB	00	08	CD	94	00	08	E1	94	00	08
000000E0	F5	94	00	08	19	AB	00	08	19	AB	00	08	19	AB	00	08
000000F0	19	AB	00	08	00	00	00	00	29	95	00	08	19	AB	00	08
Page 1 Block 4: 00000100	00	00	00	00	00	00	00	00	19	AB	00	08	19	AB	00	08
00000110	19	AB	00	08	19	AB	00	08	19	AB	00	08	00	00	00	00
00000120	19	AB	00	08	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00	5F	F8	08	F1	00	00	00	00
Block 5: 00000140	10	B5	05	4C	23	78	33	B9	04	4B	13	B1	04	48	AF	F3
00000150	00	80	01	23	23	70	10	BD	78	00	00	20	00	00	00	00
00000160	B0	38	01	08	08	B5	03	4B	1B	B1	03	49	03	48	AF	F3
00000170	00	80	08	BD	00	00	00	00	7C	00	00	20	B0	38	01	08
Block 6: 00000180	81	F0	00	41	02	E0	00	BF	83	F0	00	43	30	B5	4F	EA
00000190	41	04	4F	EA	43	05	94	EA	05	0F	08	BF	90	EA	02	0F
000001A0	1F	BF	54	EA	00	0C	55	EA	02	0C	7F	EA	64	5C	7F	EA
000001B0	65	5C	00	F0	E2	80	4F	EA	54	54	D4	EB	55	55	B8	BF

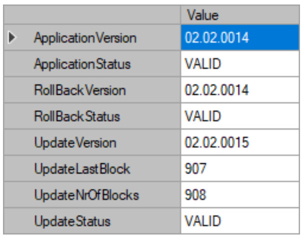
	Value
ApplicationVersion	02.02.0014
ApplicationStatus	VALID
RollBackVersion	02.02.0014
RollBackStatus	VALID
UpdateVersion	02.02.0015
UpdateLastBlock	907
UpdateNrOfBlocks	908
UpdateStatus	VALID

02020015G.bin

```
17:40:28.793  Sending message: /10823CC -> Succeeded!
18:14:58.121  Sending message: /10823CC -> Succeeded!
18:14:58.227  Sending image block 0 -> Succeeded!
18:17:32.909  Sending image block 907 -> Succeeded!
18:17:33.000  Complete image sent and verified!
18:17:33.016  Sending message: /10823CC -> Succeeded!
18:25:53.013  Sending message: /10823CC -> Succeeded!
18:25:53.122  Sending message: /10822CB -> Succeeded! -> Succeeded!
18:25:53.223  Bootloader: Backup 0 % -> Succeeded!
18:26:59.920  Bootloader: Backup 100 % -> Succeeded!
18:26:59.930  Bootloader: Update 0 % -> Succeeded!
18:27:48.378  Bootloader: Update 100 % -> Succeeded!
18:27:48.497  Bootloader: Status is IDLE
18:27:57.303  Sending message: /10823CC -> Succeeded!
18:36:43.540  Sending message: /10823CC -> Succeeded!
18:36:43.647  Sending message: /10829FA -> Succeeded!
18:36:43.740  Bootloader: Restore 0 % -> Succeeded!
18:37:50.360  Bootloader: Restore 100 % ->
18:37:50.378  Bootloader: Status is IDLE
20:55:41.374  Sending message: /10823CC -> Succeeded!
```

The Roll Back process takes a little over 1 minute.

By pressing Get partition info we get the latest status.



	Value
ApplicationVersion	02.02.0014
ApplicationStatus	VALID
RollBackVersion	02.02.0014
RollBackStatus	VALID
UpdateVersion	02.02.0015
UpdateLastBlock	907
UpdateNrOfBlocks	908
UpdateStatus	VALID