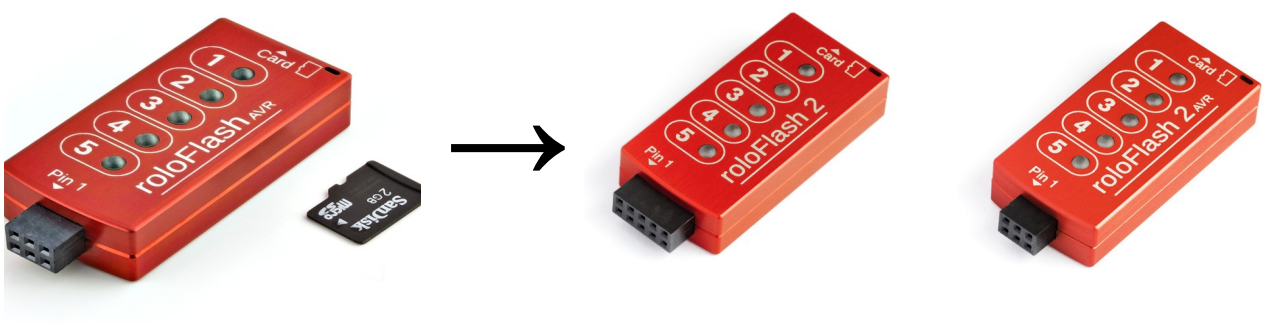
**halec**
Herrnröther Str. 54
63303 Dreieich
Germany

www.halec.de

# Migration Guide roloBasic Scripts for roloFlash AVR and roloFlash 2 Family




Guide to migrating roloBasic scripts from roloFlash AVR to the roloFlash 2 family (roloFlash 2 and roloFlash 2 AVR)

Document version 1.7.1 as of 2025-04-27
(Software version: 07.AA)

# Table of Contents

# I roloFlash: API Versions

The following roloFlash API versions are currently in circulation:

- For roloFlash AVR:
  - (only one version in circulation)

- For the roloFlash 2 family:
  - 02
  - 04
  - 05
  - 06
  - 07

The API version is encoded in the complete software version string for the roloFlash 2 family as major number.

**Example:** "07.AA": API version 7 (major version = "07", minor version = "AA")

This guide illustrates how to adapt your roloBasic scripts from roloFlash AVR to the roloFlash 2 family in API version 7.

**Note:**

The folder "scripts" on the microSD card that comes with roloFlash contains numerous example scripts for various typical applications of roloFlash and targeting various microcontrollers. These scripts might be a better starting point for converting your scripts than to follow this migration guide.

**Note:**

If you need help converting your roloBasic scripts

- from roloFlash AVR to the roloFlash 2 family or
- from one roloFlash 2 API version to another,

contact us via e-mail to <rh@halec.de>.

# II Conceptional Difference

| **roloFlash AVR** | **roloFlash 2 family** |
|---|---|
| A roloBasic script **can** have a magic cookie in its first line. This must be coded as a comment:<br><br>`!roloFlash AVR` | A roloBasic script **can** have a magic cookie in its first line. It **must** begin with a „#“ and read, for instance:<br><br>`#roloFlash 2, v07.AB`<br><br>• Please write „#roloFlash 2“ for roloFlash 2 AVR, not „#roloFlash 2 AVR“.<br>• "V07" is the API version, which is identical to the major version of the roloFlash software.<br>• „AB“ is the minor version of the roloFlash software and only serves documentation purposes. Instead of it, you can also a „*“ as Wildcard:<br><br>`#roloFlash 2, v07.*`<br><br>Or you can use a “+“ to indicate that the script can run on this firmware or higher. **This is the recommended default:**<br><br>**`#roloFlash 2, v07+`** |
| Supports only one target family, the controllers of which are always connected via ISP.<br><br>Impact on the API:<br>- Immediate communication with the target is possible (e. g. `getSignature`) | Supports multiple target families, which are connected via different busses. Some busses allow for connection of multiple targets (JTAG for STM32).<br><br>Impact on the API:<br>- First a bus must be opened (e. g. `busHandle = bus_open(ISP, 100000)`), which returns a bus handle.<br><br>- Then the target has to be opened, specifying the aforementioned bus handle, e. g. `targetHandle = target_open(busHandle, 0)`<br><br>- Only now it is possible to communicate with the target using the target handle, e. g. `target_getDeviceId(targetHandle)` |

| | |
|---|---|
| `getSignature` **implicitly** determines additional parameters like flash size and flash page size. | `target_getDeviceId` has no side effects.<br><br>The required parameters can be read from an internal database and, if you want to flash the target, have to be set **explicitly**:<br><br>`dbHandle = db_getHandle(<targetName>)`<br>`Example:`<br>`  dbHandle = db_getHandle(Atmega128)`<br><br>`flashSize = db_get(dbHandle, DB_FLASHSIZE)`<br><br>`pageSize = db_get(dbHandle, DB_FLASHPAGESIZE)`<br><br>`target_setMemoryMap targetHandle, FLASH, MEM_SIZE, flashSize`<br><br>`target_setMemoryMap targetHandle, FLASH, MEM_PAGESIZE, pageSize` |

# III Individual Functions

| roloFlash AVR | roloFlash 2 Family |
|---|---|
| targetPresent | target_getPresent(<targetHandle>) |
| programTarget | target_setMode targetHandle, PROGRAMMODE |
| runTarget | target_setMode targetHandle, RUNMODE |
| restartTarget | target_restart <targetHandle> |
| SetProgrammingSpeed <speed> | bus_open(ISP, <index>, <speed>)<br>or<br>bus_setSpeed <busHandle>, <speed> |
| getTargetVoltage | getTargetBoardVoltage |
| readBits(<index>) | target_readBits(<targetHandle>, <index>) |
| writeBits index, values | target_writeBits <targetHandle>, <index>, <values> |
| getSignature | target_getDeviceID(<targetHandle>) |
| getFlashLayout | target_getMemoryMap(<targetHandle>, FLASH, MEM_SIZE)<br>and<br>target_getMemoryMap(<targetHandle>, FLASH, MEM_PAGESIZE) |
| setFlashLayout(<size>, <pagesize>) | target_setMemoryMap <targetHandle>, FLASH, MEM_SIZE, <size>)<br>and<br>target_setMemoryMap <targetHandle>, FLASH, MEM_PAGESIZE, <pagesize>) |
| getEepromLayout | target_getMemoryMap(<targetHandle>, EEPROM, MEM_SIZE)<br>and<br>target_getMemoryMap(<targetHandle>, EEPROM, MEM_PAGESIZE) |
| setEepromLayout(<size>, <pagesize>) | target_setMemoryMap <targetHandle>, EEPROM, MEM_SIZE, <size>)<br>and<br>target_setMemoryMap <targetHandle>, |

| | |
|---|---|
| | EEPROM, MEM_PAGESIZE, <pagesize>) |
| setExtendedAddress Mode <value> | target_setExtendedAddressMode <targetHandle>, <value> |
| clearMemoryLayout | target_clearMemoryLayout <targetHandle> |
| eraseFlash | target_erase <targetHandle> |
| writeFileToFlash 0, <filename> | target_writeFromFile <targetHandle>, 0, <filename>, HEX, FLASH, WRITEONLY |
| writeVerifyFileTo-Flash 0, <filena-me> | target_writeFromFile <targetHandle>, 0, <filename>, HEX, FLASH, WRITEVERIFY |
| verifyFileToFlash 0, <filename> | target_writeFromFile <targetHandle>, 0, <filename>, HEX, FLASH, VERIFYONLY |
| writeFileToEeprom 0, <filename> | target_writeFromFile <targetHandle>, 0, <filename>, HEX, EEPROM, WRITEONLY |
| writeVerifyFileTo-Eeprom 0, <filena-me> | target_writeFromFile <targetHandle>, 0, <filename>, HEX, EEPROM, WRITEVERIFY |
| verifyFileToEeprom 0, <filename> | target_writeFromFile <targetHandle>, 0, <filename>, HEX, EEPROM, VERIFYONLY |
| All filesystem functions, e. g.:<br><br>fsOpen<br>... | <br><br>fs_open<br>... |
| All LED functions, e. g.:<br><br>ledOn<br>... | <br><br>led_on<br>... |

# IV Files Used

As of major version 05 (inclusively), the files used have been renamed:

| **roloFlash 2 family with major version of at most 04**<br>**or**<br>**roloFlash AVR** | **roloFlash 2 family with major version of at least 05**<br>**Here for V07:** |
|---|---|
| `run.bas` | `run_V07.bas` |
| `RUN.BIN` | `RUN_V07.BIN` |
| `rbc.exe` | `rbc_V07.exe` |
| `compile.bat` | `compile_V07.bat` |

As a result of these changes, the following possibilities arise:

- If you use both roloFlash AVR and roloFlash 2 (or roloFlash 2 AVR), you can prepare microSD cards so that both versions are present. Such a card can be used in roloFlash AVR as well as in roloFlash 2 or roloFlash 2 AVR, it will contain the files RUN.BIN for roloFlash and RUN_V07.BIN for roloFlash 2 or roloFlash 2 AVR with major version 07.

- With future versions (e. g. major version 07) for roloFlash 2, you can copy the scripts for multiple major versions (e. g. RUN_V07.BIN and RUN_V07.BIN) to the microSD card. With such a card, you can use the script with different roloFlash 2 having different firmware versions.

- Note: Changes of only the minor version (e. g. from V07.AA to V07.AB) do not affect the compatibility of roloBasic scripts.

- Note: You can downgrade roloFlash 2 to an older firmware version at any time. Therefore, you can test a new version and afterwards decide if you keep the new version or change to a different version.