nRF54L Series Production Programming

Application Note

Contents

Revision history . . . iii

1 Introduction . . . 4

2 Programming flow . . . 5

3 Selecting the access port . . . 6

4 Connecting . . . 7

5 Disabling APPROTECT . . . 8

6 Halting the CPU . . . 10

7 Programming . . . 11

7.1 Writing data - SECUREAPPROTECT disabled . . . 11
7.2 Verifying non-volatile memory . . . 11
7.3 Provisioning KMU data . . . 11
7.4 Enabling device protection . . . 12

8 Disconnecting . . . 14

9 Troubleshooting . . . 15

9.1 Checking protection status . . . 15

Glossary . . . 16

Recommended reading . . . 18

Legal notices . . . 19

Revision history

Date	Description
November 2024	First release

1 Introduction

This document provides information on writing software to the nRF54L Series devices and is intended for developers of flash programming tools.

It serves as a starting point for the nRF54L Series device support in production tools and accelerates the engineering process of supporting the nRF54L Series devices. This document describes a robust method for programming devices. You might not need to follow every step in some cases, for example, if the device has never been programmed before and its flash is completely erased, or if the device is unprotected.

2 Programming flow

The diagram shows the flow of production programming under normal circumstances.

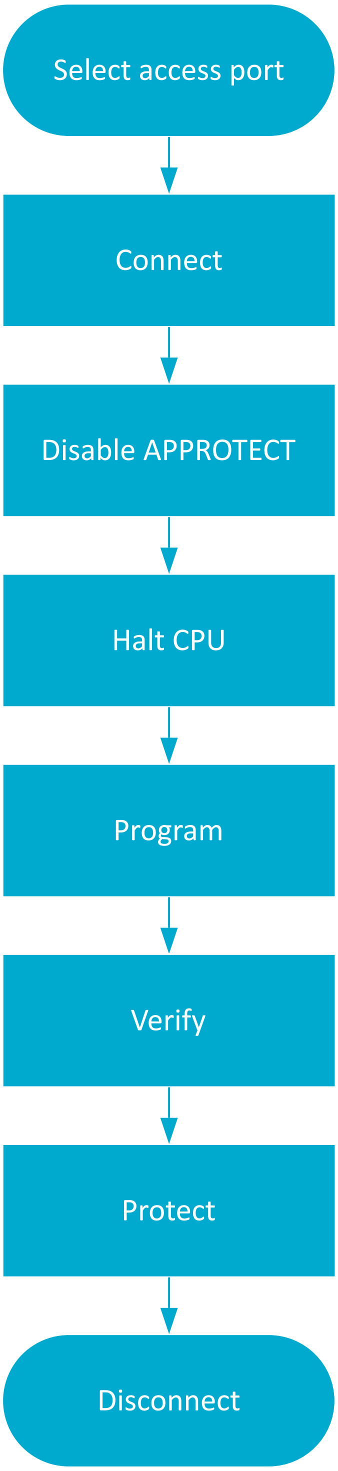

Figure 1: Normal programming flow

3 Selecting the access port

The nRF54L Series devices have one Arm® Cortex-M33 processor and one RISC-V coprocessor.

The non-volatile memory (RRAM) is shared between the two CPUs and is programmed using the access port of the Arm Cortex-M33.

When communicating with the device using the *Serial Wire Debug Port (SW-DP)*, the Cortex-M33 core at AHB-AP=0 is selected by default. The control access port uses AHB-AP=2. The coprocessor core (AUX access port) at AHB-AP=1 is only available for debugging the coprocessor code and cannot be used for device programming. To select the target processor, set AHB-AP to the corresponding value.

4 Connecting

Use the standard *Serial Wire Debug (SWD)* Arm CoreSight™ *Debug Access Port (DAP)* protocol to enter debug interface mode.

Before the external debugger can access the CPU, it must first request the device to power up and make sure that the appropriate power domains are powered up. This is managed using the built-in CxxxPWRUPREQ and CxxxPWRUPACK features found in the DAP. If the debugger requests the debug domain or the complete system to power up, the device stays in debug interface mode.

5 Disabling APPROTECT

If the device has access port protection enabled and is not erase protected, you can disable access port protection by using a *Control Access Port (CTRL-AP)* erase all operation.

To disable access port protection, complete the following steps:

1. Write `1` (Erase) to the CTRLAP.ERASEALL register (0x004).
2. Wait for the value of the CTRLAP.ERASEALLSTATUS register (0x008) to change from `2` (Busy).
3. Read the value of the CTRLAP.ERASEALLSTATUS register (0x008).
4. If the value of the CTRLAP.ERASEALLSTATUS register (0x008) is `1` (ReadyToReset), complete the following steps to do a hard reset:

   a) Write `2` (HardReset) to the CTRLAP.RESET register (0x000).

   b) Write `0` (NoReset) to the CTRLAP.RESET register (0x000).

   This completes the operation to disable the access port protection.

   If the value of the CTRLAP.ERASEALLSTATUS register (0x008) is a non-zero value other than `1`, the operation has failed and a pin reset is needed. To perform a pin reset, write `4` (PinReset) to the CTRLAP.RESET register (0x000), followed by writing `0` (NoReset) to the same register. After the pin reset, return to Step 1 to repeat the procedure.

For more information on erase protection, see Checking protection status on page 15.

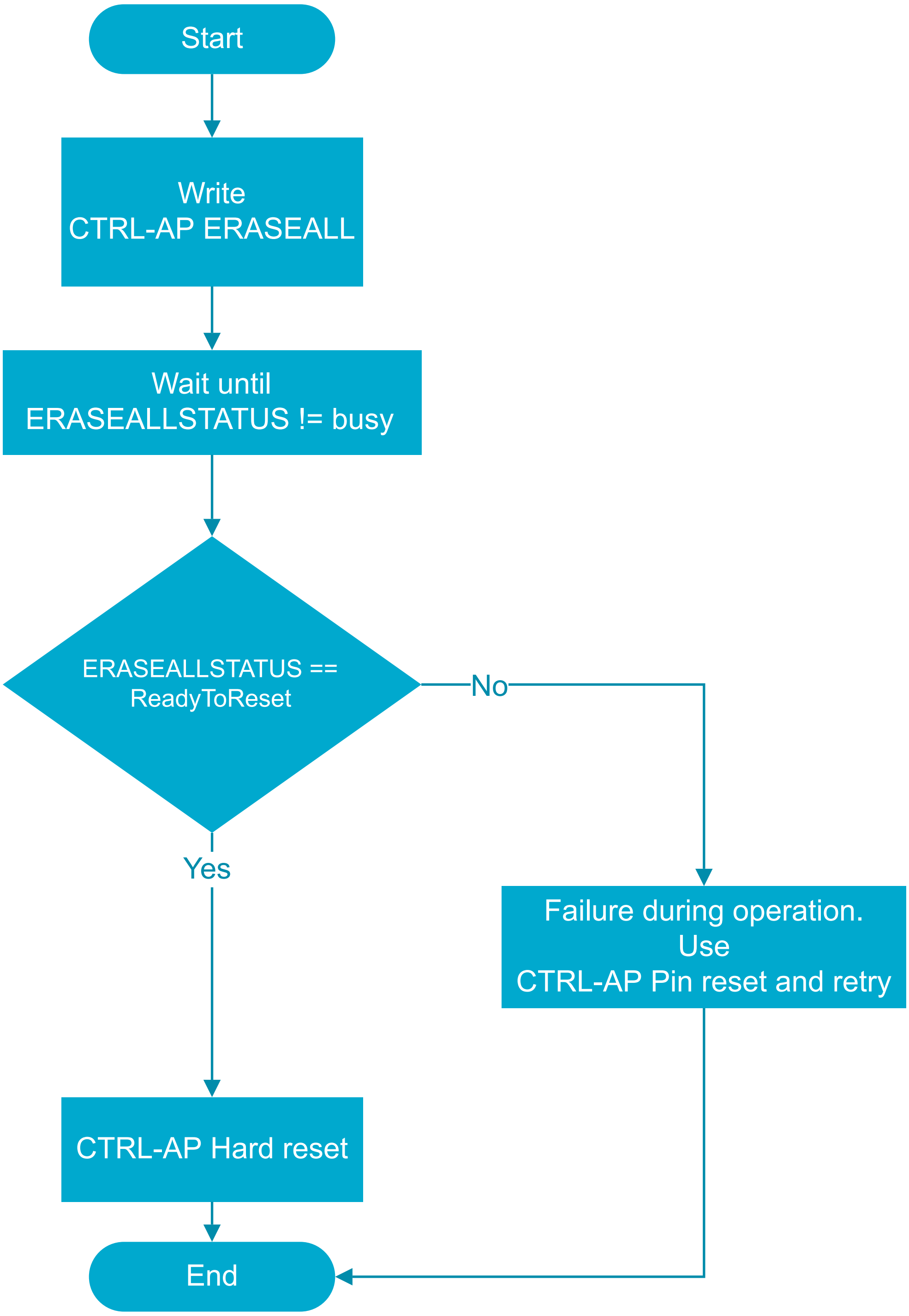

Figure 2: Erase all states

6 Halting the CPU

Use the standard *SWD* Arm CoreSight *DAP* protocol to issue a halt command to the chip.

An application running on the device that was previously programmed can use the *Watchdog timer (WDT)*. The default configuration of the WDT will pause it, if the CPU is halted.

7 Programming

Programming the device's non-volatile memory (RRAM) and *User Information Configuration Registers (UICR)* is managed through the *Resistive Random-Access Memory Controller (RRAMC)*. The key management unit (KMU) provisions assets into the SICR by copying data and permission policies from RAM to SICR.

When RRAMC write is enabled, you can write to the non-volatile memory or UICR. Only word-aligned writes are allowed when programming the UICR.

7.1 Writing data - SECUREAPPROTECT disabled

Use the standard *SWD* Arm CoreSight *DAP* protocol to write data into the non-volatile memory (RRAM).

1. Write `0x2001` to set the CONFIG register (0x5004B500) of the *RRAMC* to `WEN.Enabled` and `WRITEBUFSIZE = 32`.
   This enables buffered write to the non-volatile memory.
2. Read the READY register (0x5004B400) of the RRAMC until the value is `0x1`.

   When this value is read, the RRAMC is ready and not performing any operations.
3. Write the data in 512-byte chunks to the desired, word-aligned address.

   If the data is not 512 bytes aligned, use the COMMITWRITEBUF task to commit the data from the internal write-buffer to the non-volatile memory. Write `0x1` to the task register (0x5004B008) of the RRAMC to trigger this task.
4. When all data is written, read the READY register (0x5004B400) of the RRAMC again until the value is `0x1` to ensure the write operation has completed.
5. Write `0x0` to the CONFIG register (0x5004B500) of the RRAMC.
   This configures the non-volatile memory as read-only.

The ranges of writeable addresses are the following:

- *UICR* addresses, located in addresses 0x00FFD000 through 0x00FFDFFC
- All program memory, located in addresses 0x00000000 through ((FICR.INFO.RRAM * 1024) – 4)

7.2 Verifying non-volatile memory

To verify the contents of the non-volatile memory (RRAM) after programming, use the standard *SWD* Arm CoreSight *DAP* protocol to read every written address and compare with the expected values.

You can optimize the memory verification process by running a digest calculation of the entire memory on the device and then compare this digest to ensure it has been written correctly.

7.3 Provisioning KMU data

The KMU manages provisioning data and permission policy by copying from RAM to SICR.

1. Write the KMU `SRC data` struct as an array to RAM.
2. Write the RAM address of the KMU `SRC data` to the KMU SRC register (0x50045504).
3. Write the target key slot ID to the KMU KEYSLOT register (0x50045500).
4. Set the CONFIG register of the *RRAMC* to `WEN.Enabled` by writing `0x1` to address 0x5004B500.

5. Read the READY register (0x5004B400) of the RRAMC until the value is `0x1`.
6. Write `0x1` to address 0x50045000 to trigger the KMU provision task.
7. Read the PROVISIONED event (0x50045100) until the value is `0x1`.
8. Clear the event by writing `0x0` to the event register (0x50045100).
9. Repeat steps 1 to 8 for the required number of key slots to be provisioned.
10. Write `0x0` to the CONFIG register (0x5004B500) of the RRAMC.

The KMU data and permission policies are write-only and can therefore be verified only through a cryptographic operation during production testing.

The KMU data and permission policies can be erased only by using the CTRL-AP erase all operation.

7.4 Enabling device protection

There are several ways to protect the nRF54L Series devices. *Access Port Protection (APPROTECT)* secures the access port, *Erase Protection (ERASEPROTECT)* stops the device from being erased, and *Secure Access Port Protection (SECUREAPPROTECT)* stops unauthorized access to secure resources.

> **Note:** If the device has activated both APPROTECT and ERASEPROTECT, it cannot be recovered without a proper software solution. See Checking protection status on page 15 for more information.

APPROTECT

APPROTECT locks the *Tamper Controller (TAMPC)* PROTECT.DOMAIN signal protectors, which blocks debugger's read and write access to all CPU registers and memory mapped addresses.

To check if APPROTECT is enabled, read the UICR.APPROTECT[0].PROTECT0 (0x00FFD000) and UICR.APPROTECT[0].PROTECT1 (0x00FFD01C) registers. If these registers have any value other than `0xFFFFFFFF`, the device is protected. If the registers show the device as unprotected, write `0x50FA50FA` to them.

The protection activates after a reset. Ensure all protections are configured before performing a reset if enabling ERASEPROTECT or SECUREAPPROTECT.

ERASEPROTECT

ERASEPROTECT blocks RRAMC ERASEALL and CTRL-AP.ERASEALL functionality.

To check if ERASEPROTECT is enabled, read the UICR.ERASEPROTECT[0].PROTECT0 (0x00FFD060) and UICR.ERASEPROTECT[0].PROTECT1 (0x00FFD07C) registers. If these registers have any value other than `0xFFFFFFFF`, the device is protected. If the registers show the device as unprotected, write `0x50FA50FA` to them.

The protection activates after a reset. Ensure all protections are configured before performing a reset if enabling APPROTECT or SECUREAPPROTECT.

SECUREAPPROTECT

SECUREAPPROTECT blocks debugger's read and write access to all secure CPU registers and secures memory mapped addresses.

To check if SECUREAPPROTECT is enabled, read the UICR.SECUREAPPROTECT[0].PROTECT0 (0x00FFD020) and UICR.SECUREAPPROTECT[0].PROTECT1 (0x00FFD03C) registers. If these registers have any value other than `0xFFFFFFFF`, the device is protected. If the registers show the device as unprotected, write `0x50FA50FA` to it.

The protection activates after a reset. Ensure all protections are configured before performing a reset if enabling APPROTECT or ERASEPROTECT.

Note: A cold boot through a pin reset or power cycling the device activates the TAMPC signal protectors even if the UICR APPROTECT registers are set to unprotected. In this case, protection can be disabled only through an ERASEALL operation. For more information, see Disabling APPROTECT on page 8.

8 Disconnecting

Use the standard *SWD* Arm CoreSight *DAP* protocol to exit the debug interface mode.

This is managed using the built-in CxxxPWRUPREQ and CxxxPWRUPACK features found in the Arm CoreSight DAP. When the debugger stops requesting the debug domain or the complete system to be powered up, the device exits the debug interface mode.

We recommend a reset of the device that results in cold boot after programming by doing a power cycle.

9 Troubleshooting

The nRF54L Series devices can be reprogrammed by overwriting the memory contents. Reprogramming depends also on the current nRF54L Series security settings.

9.1 Checking protection status

In the nRF54L Series devices, access port protection is enabled by default, but the devices can also have secure access port protection and erase protection.

For more information on access port protection, see Enabling device protection on page 12.

To check if your device is protected, read the following registers in the AHB-AP:

- AHP-AP Control/Status Word (CSW) register – use this register to read the *APPROTECT* status. This register is defined in the Arm® CoreSight SoC-400 Technical Reference Manual. Use the following fields to check the access port protection status:
  - DgbStatus field (bit 6 in AHB-AP.CSW) – indicates if AHB transfers are permitted. If the value of AHB-AP.CSW->DbgStatus is `1`, then AHB transfers are allowed, and the device does not have APPROTECT.
  - SPIStatus field (bit 23 in AHB-AP.CSW) – indicates if secure protection is enabled. If the value of AHB-AP.CSW->SPIStatus is `1`, *SECUREAPPROTECT* is not set and both secure and non-secure transfers are allowed.
- CTRL-AP.ERASEPROTECT.STATUS – If access port protection is not enabled, use this register to check if *ERASEPROTECT* is set. If both APPROTECT and ERASEPROTECT are set, the device cannot be unlocked unless programmed software changes the settings.

Glossary

Access Port Protection (APPROTECT)

A register used to prevent read and write access to all CPU registers and memory-mapped addresses.

Control Access Port (CTRL-AP)

A custom access port that enables control of the device even if other access ports in the debug access port are disabled by the access port protection.

Debug Access Port (DAP)

Provides multiple master driving ports, all accessible and controlled through a single external interface port to provide system-wide debug.

Erase Protection (ERASEPROTECT)

A register used to block NVMC ERASEALL, RRAMC ERASEALL, and CTRL-AP.ERASEALL functionality.

Factory Information Configuration Registers (FICR)

Pre-programmed registers that contain chip-specific information and configuration. FICRs cannot be erased by users.

Non-volatile Memory Controller (NVMC)

A controller used for writing and erasing the internal flash memory and the *UICR*.

Resistive Random-Access Memory Controller (RRAMC)

A controller used for writing and erasing the non-volatile memory and the UICR.

Secure Access Port Protection (SECUREAPPROTECT)

A register used to prevent read and write access to all secure CPU registers and secure memory-mapped addresses.

Serial Wire Debug (SWD)

A standard two-wire interface for programming and debugging Arm CPUs.

Serial Wire Debug Port (SW-DP)

An interface that provides a low pin count bi-directional connection to the DAP with a reference clock signal for synchronous operation.

System on Chip (SoC)

A microchip that integrates all the necessary electronic circuits and components of a computer or other electronic systems on a single integrated circuit.

Tamper Controller (TAMPC)

A controller that manages inputs from internal and external physical attack detectors and controls the device's response.

User Information Configuration Registers (UICR)

Non-volatile memory registers used to configure user-specific settings.

Watchdog timer (WDT)

A timer that causes a system reset if it is not poked periodically.

Recommended reading

In addition to the information in this document, you might need to consult other documents.

Nordic documentation

- nRF54L15 | nRF54L10 | nRF54L05 Preliminary Datasheet

Legal notices

By using this documentation you agree to our terms and conditions of use. Nordic Semiconductor may change these terms and conditions at any time without notice.

Liability disclaimer

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function, or design. Nordic Semiconductor ASA does not assume any liability arising out of the application or use of any product or circuits described herein.

Nordic Semiconductor ASA does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. If there are any discrepancies, ambiguities or conflicts in Nordic Semiconductor's documentation, the Datasheet prevails.

Nordic Semiconductor ASA reserves the right to make corrections, enhancements, and other changes to this document without notice.

Life support applications

Nordic Semiconductor products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury.

Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

RoHS and REACH statement

Complete hazardous substance reports, material composition reports and latest version of Nordic's REACH statement can be found on our website www.nordicsemi.com.

Trademarks

All trademarks, service marks, trade names, product names, and logos appearing in this documentation are the property of their respective owners.

Copyright notice

QUALITY SYSTEM CERTIFICATION
DNV
ISO 9001