nRF52832 Production Programming

nWP-027

White Paper

v1.1

NORDIC® SEMICONDUCTOR

Contents

Revision history

Date	Version	Description
December 2019	1.1	Fixed broken links
October 2016	1.0	First release

1 nRF52832 Production Programming

This document provides information on downloading software to nRF52 Series devices and is intended for developers of flash download tools. It is meant to serve as a starting point for nRF52 device support in production tools and is intended to accelerate the engineering process of supporting nRF52 devices. This document describes a robust way to program devices, and in many cases steps can be skipped if assumptions can be made, for example if the chip has never been programmed before and its flash is completely erased or if the chip is unprotected.

Important: This paper focuses on nRF52 specific details. It does not explain general concepts, such as Arm®, CoreSight™, or SWD. For information on these, see ARM Infocenter.

Before continuing to read this document, we recommend that you get to know solutions by our partners Elnec, Hi-Lo Systems, and SEGGER Production Programmers, who fully support programming nRF52 series devices in production.

Elnec

- Production Programmers: Gang programmers and in-system programmers (ISP)

Hi-Lo Systems

- Production Programmers: Gang programmers
- Automated Programming Systems
- Programming Service
- Nordic Developer Zone blog post

SEGGER

- SEGGER Production Programmers: ISP programmers that program one device at a time. Ease the implementation into a production site by allowing the flash programming to be triggered manually or remotely.

We also recommend that you read the following documentation:

- nRF52832 Product Specification
  - Key sections: CPU, Memory, NVMC — Non-volatile memory controller, BPROT — Block protection, UICR — User information configuration registers, Debug interface (DIF) mode
  - Useful sections: POWER — Power supply, CLOCK — Clock control
- nRF52 Development Kit Hardware Files: Provides guidance for pin connections among other things.
- Test cases: Helps verify that your programming algorithms cover important edge cases (not complete test coverage by any means).

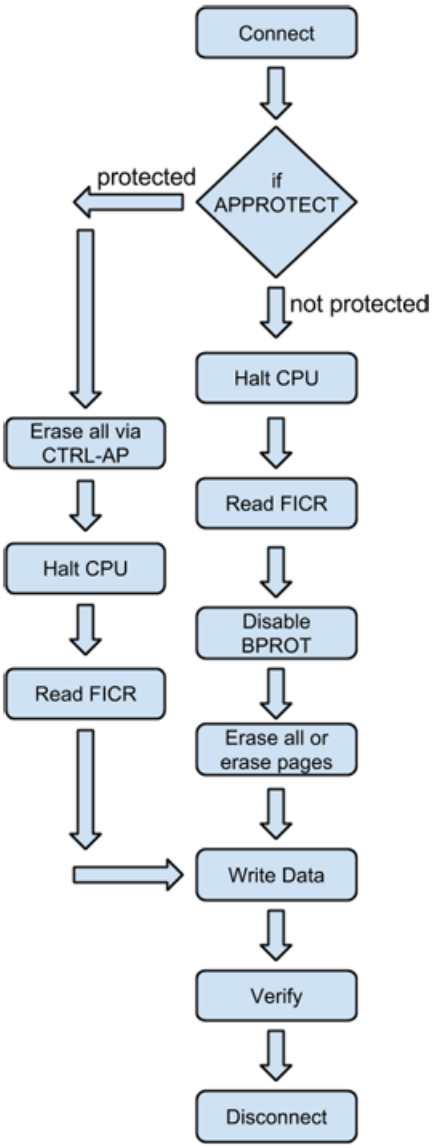

Figure 1: Programming process high-level flow

The following steps cover most corner cases. Things that this document takes into consideration when coming up with this programming flow are:

- Access port protection was enabled by a previously programmed application on the device that needs to be overwritten for any reason.
- Block protection was enabled by a previously programmed application on the device that may need to be overwritten or a different region of flash may need to be programmed while keeping the existing program unchanged.
- A watchdog timer has been enabled by a previously programmed application on the device and may reset the device. However, this document does not cover all edge cases.

2 Programming

When programming an nRF52 Series 32-bit SoC, the software is written to specific memory regions in flash.

The following sections in the nRF52832 Product Specification contain relevant information for programming SoCs:

- nRF52 series devices use flash-based non-volatile memory (NVM) in the code flash and UICR memory regions. For details, see section Memory in the nRF52832 Product Specification.
- See section Memory in the nRF52832 Product Specification for information about flash page size and number.

  Important: The Non-volatile memory controller (NVMC) is only able to write bits in the NVM that are erased (set to '1'). The NVMC is used for writing and erasing all flash memory. Before a write can be performed, the NVM must be enabled for writing in CONFIG.WEN. Similarly, before an erase can be performed, the NVM must be configured for erasing in CONFIG.EEN. The CPU is halted when the NVMC is performing a write/erase operation. Check the Ready flag to make sure the NVMC is not busy with an ongoing write or erase operation before performing an operation with the NVMC.
- The Debug and trace module provides access to the on-chip debug functionality. This is a standard two-pin serial wire debug (SWD) interface as defined by Arm For details, see Debug and trace section in the nRF52832 Product Specification.

  Note: The SWDIO line has an internal pull-up resistor and the SWDCLK line has an internal pull-down resistor.
- See section NVMC — Non-volatile memory controller in the nRF52832 Product Specification for detailed specifications regarding timing for write/erase operations. For example, it takes the NVMC 67.5 (min) to 338 microseconds (max) to write one word in flash, and it takes between 6.72 (min) and 295.3 (max) milliseconds to erase all flash. Based on this information, the theoretical fastest run-time of flash programming algorithms can be calculated. The theoretical best time to write the entire flash of an nRF52832 device is approximately 8.85 seconds: (512 * 1024) / 4 * (67.5 * 10^-6).

  Note: This is theoretical because it does not take real-world overhead into consideration, such as external tester interface, algorithm executed in RAM, SWDP speed, etc.

3 Connecting

Use the standard SWD Arm CoreSight Debug Access Port (DAP) protocol to enter Debug interface (DIF) mode. A standard DAP as provided by ARM is used and the Debug Port (DP) is in an always-on domain to secure that the CxxxPWRUPREQ can be issued even if the device is in System OFF mode.

For more information about DIF mode, see section Debug interface (DIF) mode in the nRF52832 Product Specification.

Before the external debugger can access the CPU, it must first request and make sure that the appropriate power domains are powered up. This is handled using the built-in CxxxPWRUPREQ and CxxxPWRUPACK feature found in the Arm CoreSight DAP. As long as the debugger is requesting the debug domain or the complete system to be powered up, the device will be in debug interface mode.

4 Checking if access port protection is enabled (APPROTECT)

The CTRL-AP - Control Access Port is a custom access port that enables control of the device even if the other access ports in the DAP are being disabled by access port protection.

If access port protection has been enabled in the APPROTECT register (0x10001208) of the UICR, the debugger's read/write access to all CPU registers and memory mapped addresses is blocked.

For more information about CTRL-AP, see section CTRL-AP - Control Access Port in the nRF52832 Product Specification.

Using the standard SWD Arm CoreSight DAP protocol:

1. Select/connect to the control access port.

   This access port is at index 0x01.
2. Read the APPROTECTSTATUS register (0x00C) of the CTRL-AP.

   If the least significant bit of this register is '0', access port protection is enabled. Go to Erasing all through CTRL-AP on page 8.

   If the least significant bit of this register is '1', access port protection is not enabled. Go to Halting CPU on page 8.

4.1 Access port protection is enabled (protected)

If access port protection is enabled on the device, access port 0 is unavailable.

The only way to 'reopen/unlock' the device is to issue an ERASEALL command through the CTRL-AP access port, and then issue a reset through the CTRL-AP. This will erase the entire code flash and UICR area of the device, in addition to the entire RAM. This method of erasing is slower than performing an erase all with the NVMC since it also must erase all RAM, but if access port protection is enabled, it is the only way to unlock the device.

4.1.1 Erasing all through CTRL-AP

Use the standard SWD Arm CoreSight DAP protocol to erase all while the CTRL-AP is still selected by the DP.

1. Write the value 0x00000001 to the ERASEALL register (0x004) of the CTRL-AP.

   This will start the ERASEALL operation which erases all flash and RAM on the device.
2. Read the ERASEALLSTATUS register (0x008) of the CTRL-AP until the value read is 0x00 or 15 seconds from ERASEALL write has expired.
3. Write the value 0x1 to RESET register (0x000) of the CTRL-AP to issue a "soft reset" to the device and complete the erase and unlocking of the chip.
4. Write the value 0x0 to RESET register (0x000).
5. Write the value 0x0 to the ERASEALL register (0x004) of the CTRL-AP.

   This is necessary after the erase sequence is completed.

4.1.2 Halting CPU

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

4.1.3 Reading FICR

Factory information configuration registers (FICR) are pre-programmed in the factory and cannot be erased by the user. These registers contain chip-specific information and configuration.

Using the standard SWD Arm CoreSight DAP protocol:

1. Read the CODEPAGESIZE register (0x10000010) of the FICR.

   The value of this register will contain the code memory page size in hexadecimal format, so 0x00001000 stored in this register corresponds to a page size of 4096 bytes.
2. Read the CODESIZE register (0x10000014) of the FICR.

   The value of this register will contain the number of pages in code memory in hexadecimal format, so 0x0000080 stored in this register corresponds to 128 total pages in flash memory.

Note: Total flash memory (in bytes) = CODEPAGESIZE * CODESIZE. This information will be used later to determine the valid range of addresses to program.

4.2 Access port protection is not enabled (not protected)

This means that the UICR has not been previously configured to enable access port protection.

In some cases, you may assume that the entire flash has already been erased. If the flash is already erased and the device has never been programmed before, skip to Writing data.

4.2.1 Reading FICR

Factory information configuration registers (FICR) are pre-programmed in the factory and cannot be erased by the user. These registers contain chip-specific information and configuration.

Using the standard SWD Arm CoreSight DAP protocol:

1. Read the CODEPAGESIZE register (0x10000010) of the FICR.

   The value of this register will contain the code memory page size in hexadecimal format, so 0x00001000 stored in this register corresponds to a page size of 4096 bytes.
2. Read the CODESIZE register (0x10000014) of the FICR.

   The value of this register will contain the number of pages in code memory in hexadecimal format, so 0x0000080 stored in this register corresponds to 128 total pages in flash memory.

Note: Total flash memory (in bytes) = CODEPAGESIZE * CODESIZE. This information will be used later to determine the valid range of addresses to program.

4.2.2 Halting CPU

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

Important: An application running on the device that was previously programmed may use the watch dog timer (WDT). If this is the case, the WDT will be paused when the CPU is halted by default.

4.2.3 Disabling block protection (BPROT)

The mechanism used to protect NVM from erroneous application code erasing/writing to protected pages in code flash will cause the CPU to hard fault if an erase or write to a protected page is attempted. This can be turned off when in debug mode by configuring the DISABLEINDEBUG register.

For more information on BPROT, see section BPROT — Block protection in the nRF52832 Product Specification.

Using the standard SWD Arm CoreSight DAP protocol, write the value 0x00000001 to the DISABLEINDEBUG register (0x40000608) of the BPROT. This will disable the block protection mechanism in debug mode.

5 Erasing all or erasing pages (optional)

An erase all operation takes the same amount of time as erasing three pages one by one. As there are 128 total pages of flash, erasing all will be more efficient than erasing page by page. In the case that a region of the chip has been preprogrammed, you can erase the flash you intend to program page by page and then write those addresses with data leaving pre-programmed flash untouched. If the entire flash of the device is erased (0xFFFFFFFF), skip this step.

5.1 Erase-all

Use the standard SWD Arm CoreSight DAP protocol to erase all.

1. Write the value 0x00000002 to the CONFIG register (0x4001E504) of the NVMC.
   This will configure the NVM for erasing.
2. Read the READY register (0x4001E400) of the NVMC until the value is 0x00000001.
   When this value is read, the NVMC is ready and not currently performing any operations.
3. Write the value 0x00000001 to the ERASEALL register (0x4001E50C) of the NVMC.
   This will erase all NVM including UICR registers.
4. Read the READY register (0x4001E400) of the NVMC until the value is 0x00000001 before continuing to ensure the erase all operation has completed.
5. Write the value 0x00000000 to the CONFIG register (0x4001E504) of the NVMC.
   This will configure the NVM back to read-only.

5.2 Erasing page by page

Use the standard SWD Arm CoreSight DAP protocol to erase page by page.

1. Write the value 0x00000002 to the CONFIG register (0x4001E504) of the NVMC.
   This will configure the NVM for erasing.
2. Read the READY register (0x4001E400) of the NVMC until the value is 0x00000001.
   When this value is read, the NVMC is ready and not currently performing any operations.
3. Write the value of the address of the first word in desired page to be erased to the ERASEPAGE register (0x4001E508). This will start the erase of a page in code flash. Use the data read from the FICR for information about the total code size of the device you are programming, number of pages and page size. Attempts to erase pages that are outside the code flash area may result in undesirable behavior. If the page to be erased is the UICR page, write 0x00000001 to ERASEUICR register (0x4001E514) instead of erasing as you would for a normal page in flash.
4. Read the READY register (0x4001E400) of the NVMC until the value is 0x00000001 before continuing to ensure the erase page operation has completed.
5. Continue erasing page by page until done.
   If you are erasing more than three pages, an erase-all will be much faster.
6. Write the value 0x00000000 to the CONFIG register (0x4001E504) of the NVMC.
   This will configure the NVM back to read-only.

6 Writing data

When writing is enabled, the NVM is written by writing a word to a word-aligned address in the code or UICR. Only word-aligned writes are allowed. Byte or half-word-aligned writes will result in a hard fault.

To write the data into flash using the standard SWD Arm CoreSight DAP protocol:

1. Write the value 0x00000001 to the CONFIG register (0x4001E504) of the NVMC.
   This will configure the NVM for writing.
2. Read the READY register (0x4001E400) of the NVMC until the value is 0x00000001.
   When this value is read, the NVMC is ready and not currently performing any operations.
3. Write the data to the desired, word-aligned address.
4. Read the READY register (0x4001E400) of the NVMC until the value is 0x00000001 before continuing to ensure the write operation has completed.
5. Continue writing and then reading the READY register (0x4001E400) as necessary.
6. Write the value 0x00000000 to the CONFIG register (0x4001E504) of the NVMC.

   This will configure the NVM back to read only.

   The ranges of writeable addresses are:

   - UICR addresses (located in addresses 0x10001000 through 0x10002000)

     Important: These addresses must be writeable by the production programming tools. Users expect these to be written when the hex file is programmed by the programmer. It is bad practice for application to write these values at run time.

   - All program flash (located in addresses 0x00000000 through ((CODESIZE * CODEPAGESIZE) – 0x00000004)

     Different methods can be used to write flash. For the nRF52, a good flash algorithm should take around 10 seconds to write the entire flash. If you are not able to reach this time, contact Nordic Semiconductor for assistance.

7 Verifying (optional)

To verify the contents of flash after programming, use the standard SWD Arm CoreSight DAP protocol and read every address written and compare with the expected value.

Important: It is possible that the hex file being programmed will enable access port (read-back) protection that will make it impossible to verify the contents of flash. This protection will only take effect after a reset is applied. Make sure not to reset between Writing data and Verifying.

8 Disconnecting

Use the standard SWD ArmCoreSight DAP protocol to exit DIF mode. This is handled 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 will exit the debug interface mode.

We recommend a hard reset of the device after programming by the use of a power cycle to the device.

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 Product Specification 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

© 2019 Nordic Semiconductor ASA. All rights are reserved. Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

COMPANY WITH
QUALITY SYSTEM
CERTIFIED BY DNV GL
= ISO 9001 =