

# Using flash devices as development platforms for OTP devices

nAN24-14

## Application Note

## 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.

## Life support applications

These 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.

## Contact details

For your nearest dealer, please see <http://www.nordicsemi.no>

Receive available updates automatically by subscribing to eNews from our homepage or check our website regularly for any available updates.

### Main office:

Otto Nielsens veg 12  
7004 Trondheim  
Phone: +47 72 89 89 00  
Fax: +47 72 89 89 89  
[www.nordicsemi.no](http://www.nordicsemi.no)



## Revision History

Date	Version	Description
October 2009	1.0	Application note

---

## Contents

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
<b>2</b>	<b>General memory differences.....</b>	<b>5</b>
2.1	nRF24LE1 and nRF24LE1 OTP memory differences .....	5
2.2	nRF24LU1+ and nRF24LU1+ OTP memory differences .....	7
<b>3</b>	<b>Using nRFgo SDK .....</b>	<b>9</b>
3.1	Checking your code for incompatibility.....	9
<b>4</b>	<b>Summary .....</b>	<b>10</b>

---

## 1 Introduction

When developing firmware for nrF24LE1 OTP or nRF24LU1+ OTP the flash based variants are suitable development platforms. The flash variants are almost identical to the OTP variants, the only difference being the memories used for program and non-volatile memory. If you keep these memory differences in mind when developing, you get the convenience of a flash memory during the development phase.

In this document we will explain the differences in detail, highlight some subtle implications, and give some advices on how to avoid and detect bugs caused by the differences.

---

## 2 General memory differences

One major – and rather obvious – difference from the flash devices is that the OTP memory cannot be erased. The firmware developed on a flash based circuit must therefore never use this functionality. Multiple writes are possible as long as you change a '1' to a '0', but you can never change a '0' back to a '1'.

Another difference is in the MCU programming functions. For OTP the Cclk must be set to 16 MHz. The write time is significantly longer; 12800 clock cycles (0.8 ms) per byte in OTP against 740 clock cycles per byte for flash. The write process is self-timed and the MCU will stop until it is finished.

As a consequence, software timing must be independent of the OTP timing. The implications of the timing difference must be taken into account if the memory is used as data storage. The programming time for OTP is comparable to protocol timing, and can therefore cause different behavior from that observed on a flash device.

### 2.1 nRF24LE1 and nRF24LE1 OTP memory differences

In the nRF24LE1 OTP the Non-Volatile Extended endurance memory in the nRF24LE1 flash version is not available. To ensure flash and OTP compatibility the firmware cannot use this memory. As an alternative,

the application can use the normal Non-Volatile data memory for parameter storage, which has the same memory mapping for both the OTP and flash variant. The following figure illustrates this difference.

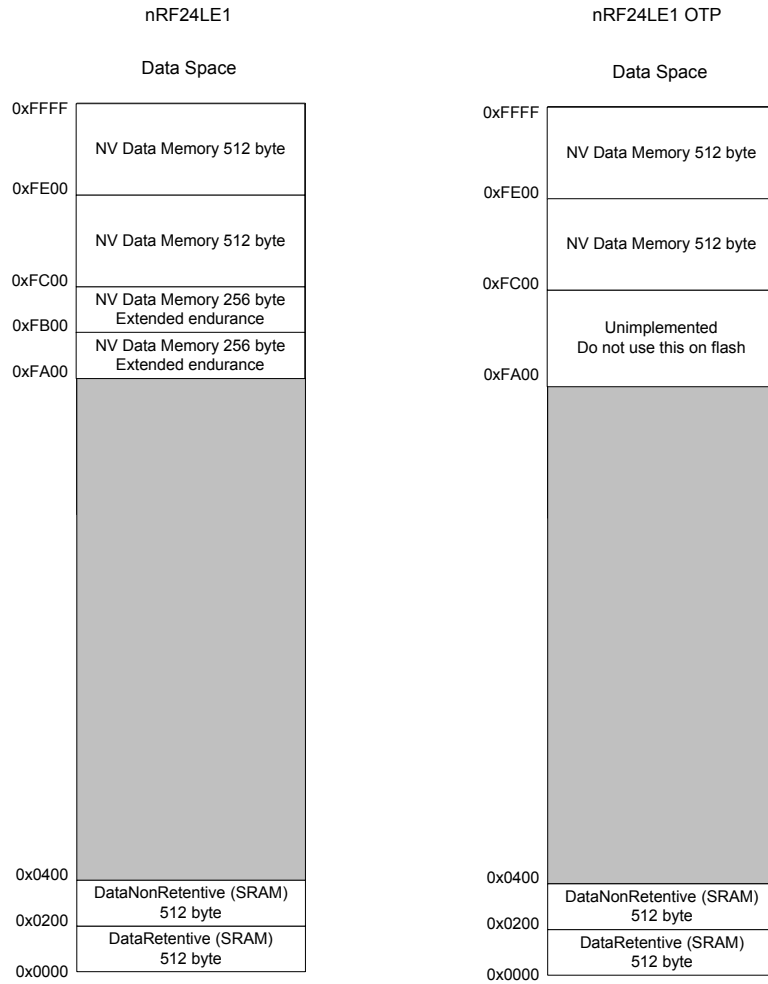


Figure 1. Non-volatile data memory, nRF24 LE1 and nRF24 LE1 OTP

**Note:** The non-volatile data memory can never be erased due to the general OTP limitations.

## 2.2 nRF24LU1+ and nRF24LU1+ OTP memory differences

The flash nRF24LU1+ has 32 kB code space while the nRF24LU1+ OTP has only 17 kB of code space. [Figure 2. on page 7](#) illustrates the differences. To ensure compatibility the firmware must not use the unimplemented memory.

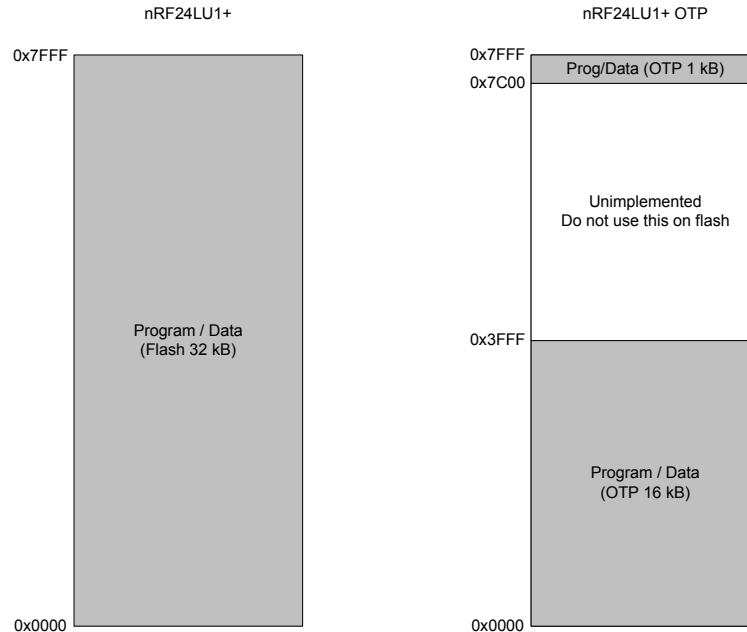


Figure 2. Differences in memory between nRF24LU1+ and nRF24LU1+ OTP

Keil  $\mu$ Vision lets you set up the code memory manually. [Figure 3. on page 8](#) shows how to set up the code memory to only use what is implemented on nRF24LU1+ OTP.

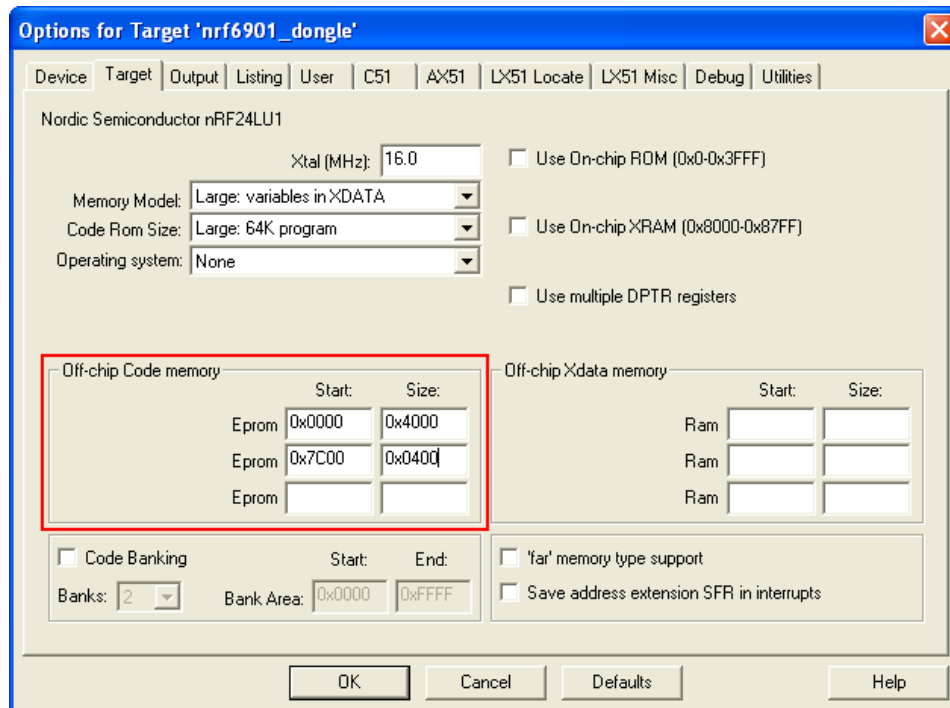


Figure 3. Graphic user interface for setting up code memory

In the “Off-chip memory” we have defined the first block to start at address 0 and have a size of 16 kB (0x4000). The next block starts at address 0x7C00 and its size is 1 kB (0x0400).

---

## 3 Using nRFgo SDK

All libraries for nRF24LE1 / nRF24LU1+ in nRFgo SDK can be used on nRF24LE1 OTP / nRF24LU1+ OTP except for those using the **hal\_flash** hardware abstraction layer, or libraries using the extended endurance NV data memory in nRF24LE1, such as **lib\_eeeprom255**.

To be precise, in **hal\_flash** it is the function **hal\_flash\_page\_erase()** that will not work on the OTP device.

### 3.1 Checking your code for incompatibility

#### 3.1.1 Page erase

Detection of “page erase” can be done at multiple levels:

- Function: `hal_flash_page_erase()`
- C Code: `WEN = 1; FCR = pn;`
- Assembly: `SETB WEN(F8.5) MOV FCR(FA), pn`
- Compiled binary: code sequence `D2FD8FFA`

Searching in the binary code can give a clue, but it depends on the compiler if the code is compiled exactly like this. Also the sequence could be the contents of a constant array.

#### 3.1.2 Writing to unimplemented memory

When reading from a memory address that is not implemented in OTP the obtained value should be `0xFF`. This can be used for a test that ensures that the firmware does not write to unavailable memory.

The firmware under test should run for a while and use the specified functionality. We then read the illegal address range to see if any change has occurred.

To do this we create a HEX file that, in addition to the program itself, defines the unavailable memory to contain `0xFF`. The flash main block is then verified against the HEX file. If the firmware has written anything to the unavailable memory the verification will fail.

---

## 4 Summary

To ensure code compatibility when using the nRF24LE1 or nRF24LU1+ when developing firmware for an OTP device, you must take into account the following differences:

- The available memory is different: 17 kB vs 32 kB on nRF24LU1+
- NV Data Memory extended endurance is not available in nRF24LE1 OTP
- OTP memory cannot be erased
- Write time for OTP is significantly longer

Setting up the valid memory in Keil  $\mu$ Vision and avoid using `hal_flash` are simple ways to avoid incompatible firmware.

You can also check that your firmware does not exceed the available memory space, is free of page erase instructions, and does not write to unimplemented memory.