

---

## The BlueNRG-1 and BlueNRG-2 UART bootloader protocol

### Introduction

The BlueNRG-1 and BlueNRG-2 are very low power Bluetooth Low Energy (BLE) single-mode systems-on-chip, compliant with Bluetooth® specifications. The architecture core is a Cortex-M0 32-bit.

The BlueNRG-2N is an ultra low power (ULP) network coprocessor solution for Bluetooth® Low Energy applications.

This application note contains the specifications of the BlueNRG-1 and BlueNRG-2/BlueNRG-2N UART bootloader.

*Note: The contents of this document are valid for the BlueNRG-1 and the BlueNRG-2, BlueNRG-2N devices. Therefore, any reference to the BlueNRG-2 device is also valid for the BlueNRG-2N device. Specific differences have been highlighted whenever needed.*

## 1 UART bootloader configuration

---

To communicate with the BlueNRG-1 and BlueNRG-2 bootloader the host UART has to be configured as follows:

- UART data 8 bit
- NO parity
- Stop bit 1
- NO flow control
- Baud rate range [500 – 460800]

The bootloader is configured to use the UART pin:

- UART RX = DIO11
- UART TX = DIO8

## 2 UART bootloader activation

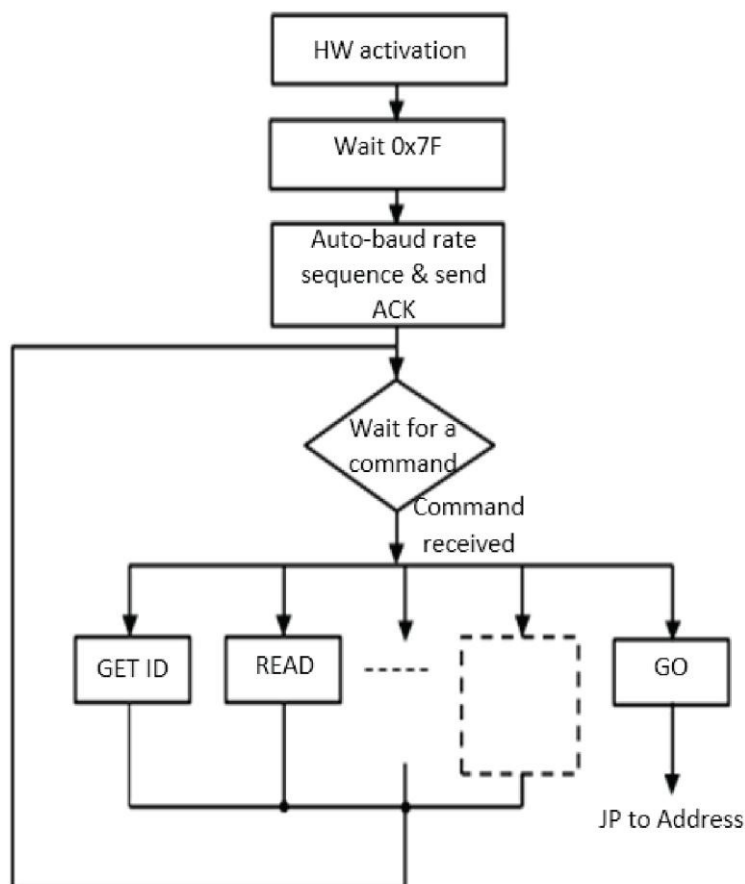
The BlueNRG-1 and BlueNRG-2 bootloader is activated by hardware forcing high DIO7 at device reset. Once the bootloader is activated, the code starts a procedure to auto-detect the host UART baud rate and begins to scan the USART RX line pin, waiting for the 0x7F data frame from the host: one start bit, 0x7F data bits, no parity bit and one stop bit.

The duration of this data frame is measured using the SysTick timer. The timer count value is then used to calculate the corresponding baud rate factor with respect to the current system clock.

Once the baud rate is calculated, the firmware initializes the BlueNRG-1 or BlueNRG-2 serial interface accordingly, using this calculated baud rate.

An acknowledge byte (0x79) is returned to the host, which signals that the BlueNRG-1 or BlueNRG-2 is ready to receive commands.

**Figure 1. UART bootloader for BlueNRG-1 and BlueNRG-2 devices**



### 3 UART bootloader commands

The table below lists the supported commands, fully detailed in the following subsections.

**Table 1. UART bootloader commands**

| Command                          | Command code | Command description   |
|----------------------------------|--------------|---|
| Get List Command                 | 0x00         | Gets the version and the allowed commands supported by the current version of the bootloader            |
| Get Version                      | 0x01         | Gets the bootloader version   |
| Get ID                           | 0x02         | Gets the chip ID  |
| Read Memory                      | 0x11         | Reads up to 256 bytes of memory starting from an address specified by the application                   |
| Go                               | 0x21         | Jumps to user application code located in the internal Flash memory or in RAM                           |
| Write Memory                     | 0x31         | Writes up to 256 bytes to the RAM or Flash memory starting from an address specified by the application |
| Erase <sup>(1)</sup>             | 0x43         | Erases from one to all the Flash memory pages   |
| Readout Protect                  | 0x82         | Enables the read protection   |
| Readout Unprotect <sup>(2)</sup> | 0x92         | Disables the read protection  |

1. Mass erase command option is not supported on the BlueNRG-2N device.

2. Readout Unprotect is not supported on the BlueNRG-2N device.

All communications from the host to the BlueNRG-1 or BlueNRG-2 device are safe because they are verified by:

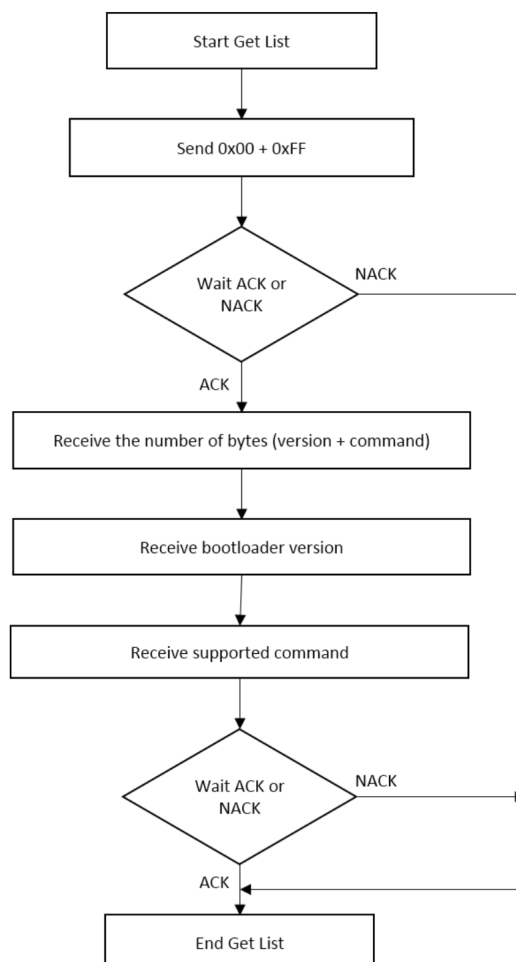
- Checksum: received blocks of data bytes are XORed. A byte containing the computed XOR of all previous bytes is added to the end of each communication (checksum byte). By XORing all received bytes, data + checksum, the result at the end of the packet must be 0x00
- For each bootloader command, the host sends a byte and its complement
- Each packet is either accepted (ACK answer) or discarded (NACK answer):
  - ACK = 0x79
  - NACK = 0x1F

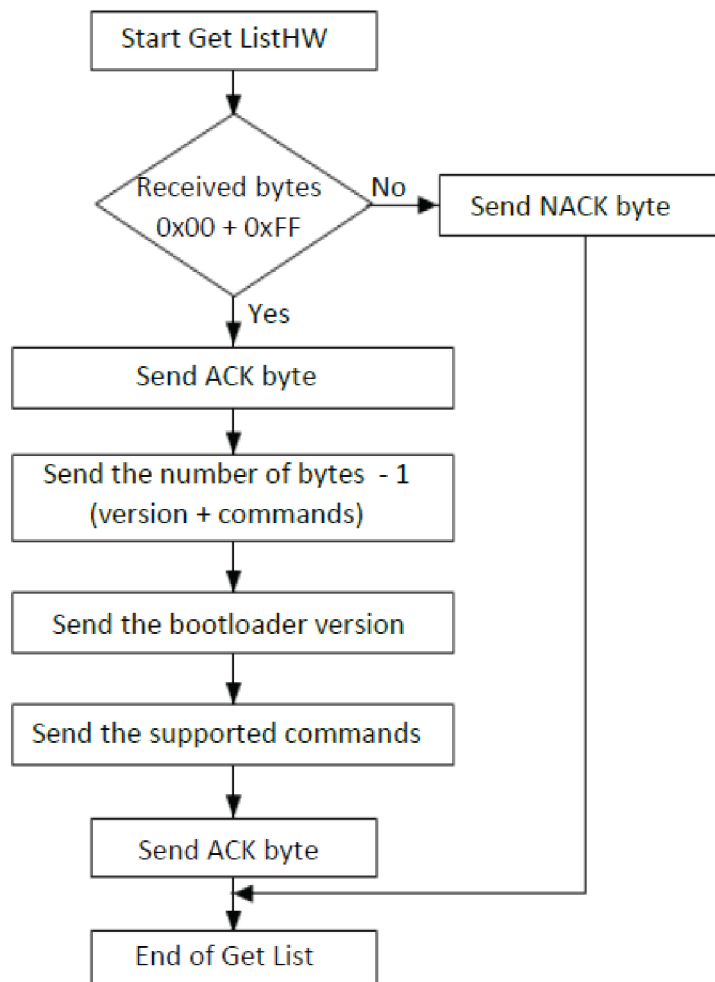
#### 3.1 Get List command

The Get List command allows getting the version of the bootloader and the supported commands.

When the BlueNRG-1 or BlueNRG-2 bootloader receives the Get List command, it transmits the bootloader version and the supported command codes to the host.

Figure 2. Get List command: host side



**Figure 3. Get list command: device side**


The BlueNRG-1 or BlueNRG-2 sends the bytes as follows:

Byte 1: ACK

Byte 2: N = 9 = the number of bytes to follow -1 except current and ACKs.

Byte 3: Bootloader version (0 < Version <= 255), example: 0x01 = Version 1.0

Byte 4: 0x00 – Get List command

Byte 5: 0x01 – Get Version command

Byte 6: 0x02 – Get ID command

Byte 7: 0x11 – Read Memory command

Byte 8: 0x21 – Go command

Byte 9: 0x31 – Write Memory command

Byte 10: 0x43 – Erase command

Byte 11: 0x82 – Readout Protect command

Byte 12: 0x92 – Readout Unprotect command

Byte 13: ACK. This is the last byte

## 3.2 Get Version command

The Get Version command is used to get the bootloader version. When the bootloader receives the command, it transmits the information shown below to the host.

**Figure 4. Get Version command: host side**

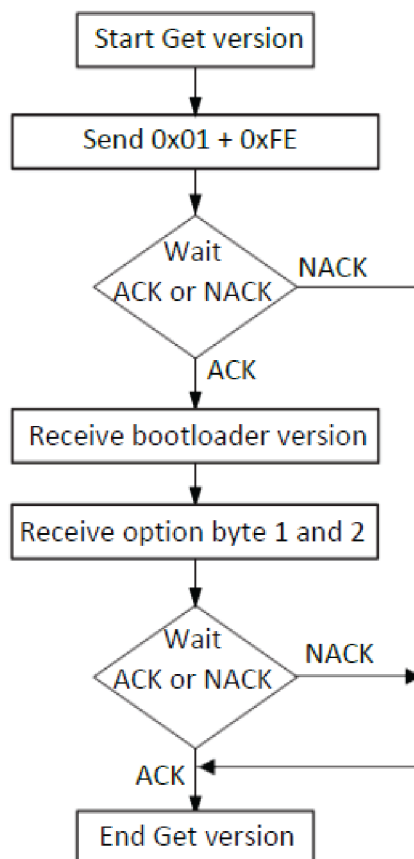
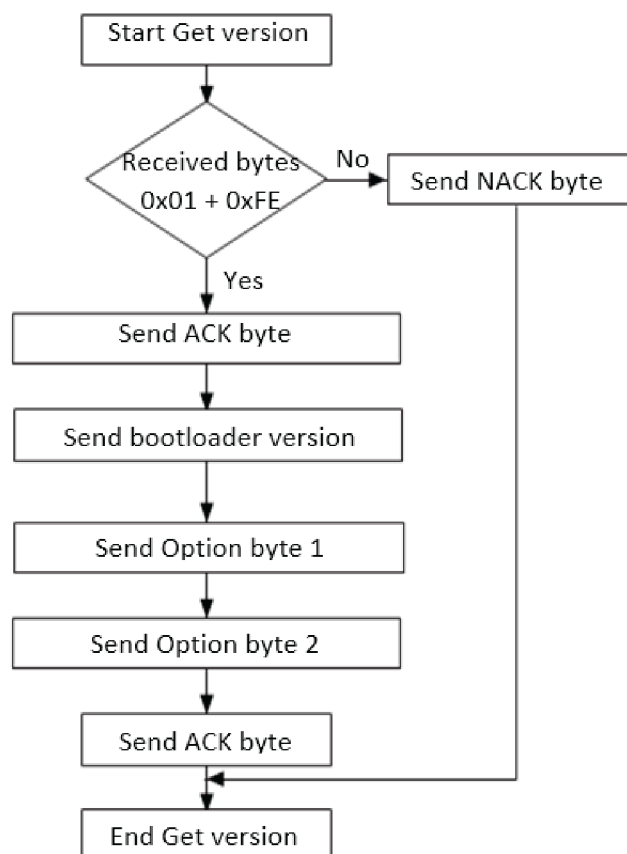


Figure 5. Get Version command: device side



The BlueNRG-1 or BlueNRG-2 device sends the bytes as follows:

Byte 1: ACK

Byte 2: Bootloader version ( $0 < \text{Version} \leq 255$ ), example: 0x01 = Version 1.0

Byte 3: Option byte 1: 0x00

Byte 4: Option byte 2: 0x00

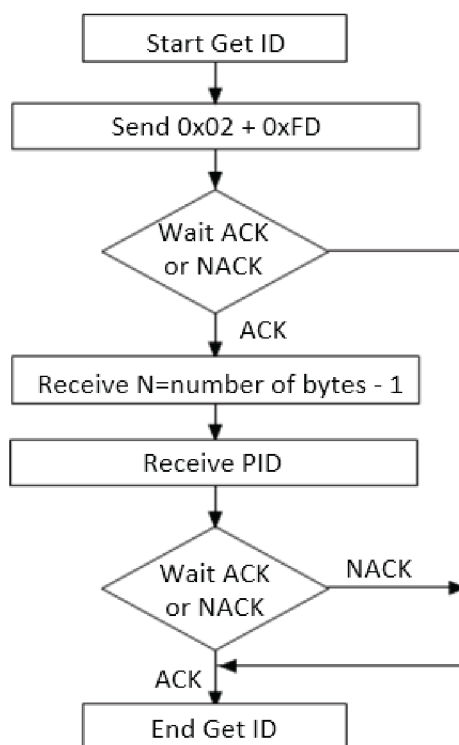
Byte 5: ACK

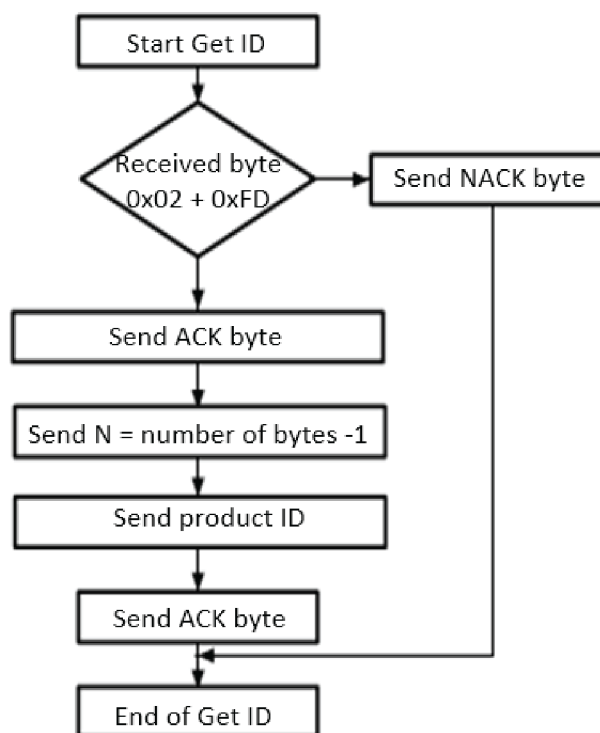


### 3.3 Get ID command

The Get IDcommand is used to get the version of the chip ID (identification). When the bootloader receives the command, it transmits the product ID to the host.

**Figure 6. Get ID command: host side**



**Figure 7. Get ID command: device side**


The BlueNRG-1 or BlueNRG-2 device sends the bytes as follows:

Byte 1: ACK

Byte 2: 0x02: the number of bytes -1, except for current byte and ACKs

Byte 3-5: PID

Byte 6: ACK

**Note:** The Get ID command format returns three bytes:

- **BYTE1:** Metal fix version (e.g. cut 1.0 will be 0)
- **BYTE2:** Mask set version (e.g cut 1.0 will be 1)
- **BYTE3:** is split in two nibbles: 0xHL:
  - *H* is the product ID:
    - 2 means BlueNRG-2
    - 0 means BlueNRG-1
  - *L* is the flash size code:
    - 3: 160 KB (for BlueNRG-1)
    - F: 256 KB (for BlueNRG-2)

### 3.4 Read Memory command

The Read Memory command is used to read data from any valid memory address in RAM and Flash memory. When the bootloader receives the Read Memory command, it transmits the ACK byte to the application. Afterwards, the bootloader waits for an address (4 bytes, byte 1 is the address MSB and byte 4 is the LSB) and a checksum byte. If the received address is valid and the checksum is correct, the bootloader transmits an ACK byte, otherwise it transmits a NACK byte and aborts the command.

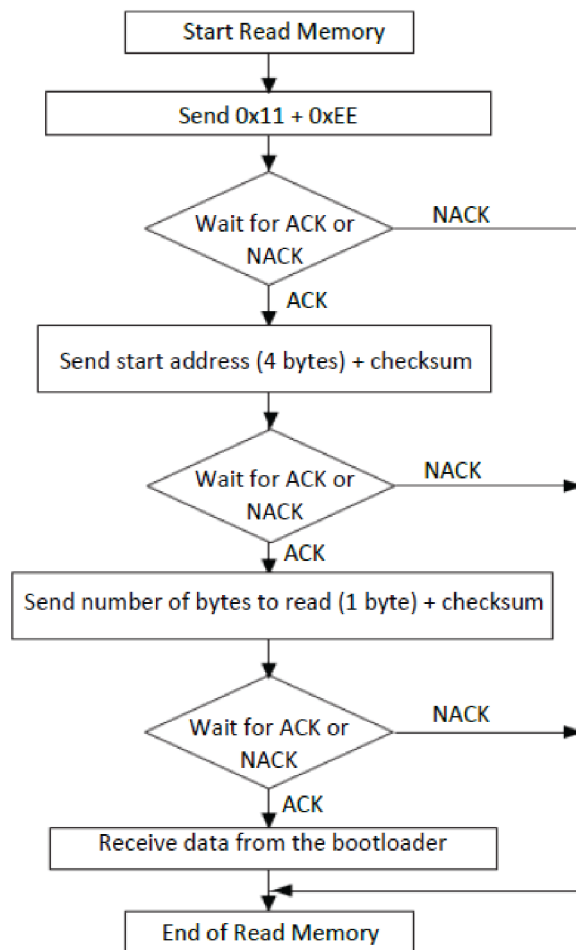
When the address is valid and the checksum is correct, the bootloader waits for N that is the number of bytes to be transmitted – 1 and for its complemented byte (checksum).

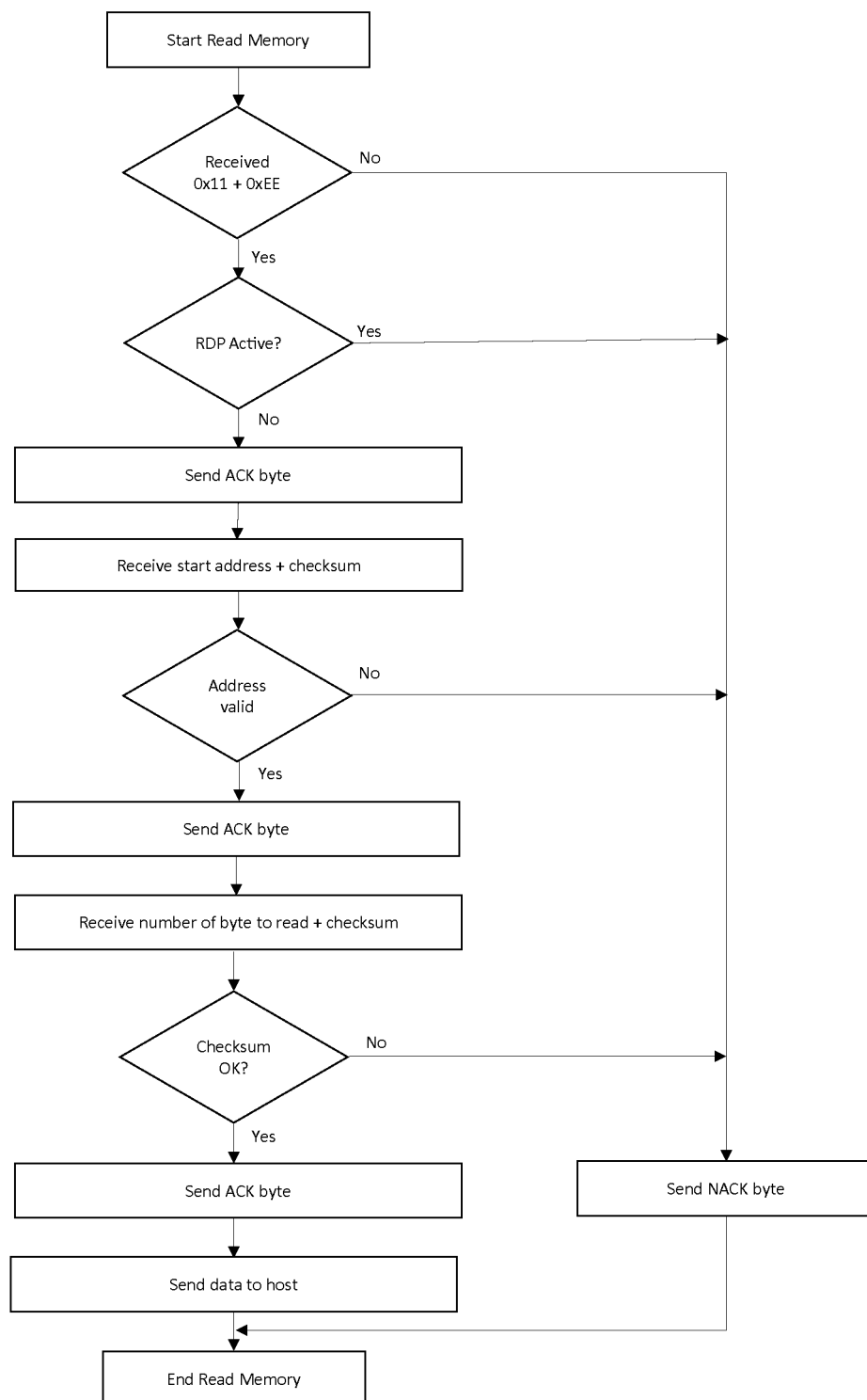
If the checksum is correct it transmits the needed data ((N + 1) bytes) to the application, starting from the received address.

If the checksum is not correct, it sends a NACK before aborting the command.

If the readout protection is active, a NACK byte is sent to the host when the Read Memory command is received.

Figure 8. Read Memory command: host side



**Figure 9. Read Memory command: device side**


The host sends bytes to the BlueNRG-1 or BlueNRG-2 as follows:

Byte 1: 0x11  
 Byte 2: 0xEE  
 Wait for ACK  
 Byte 3 to 6: start address
 

- Byte 3: MSB
- Byte 6: LSB

 Byte 7: Checksum: XOR of address bytes  
 Wait for ACK  
 Byte 8: Number of bytes to read -1 ( $0 < N \leq 255$ )  
 Byte 9: Checksum: XOR byte 8 (complement of byte 8)

### 3.5 Go command

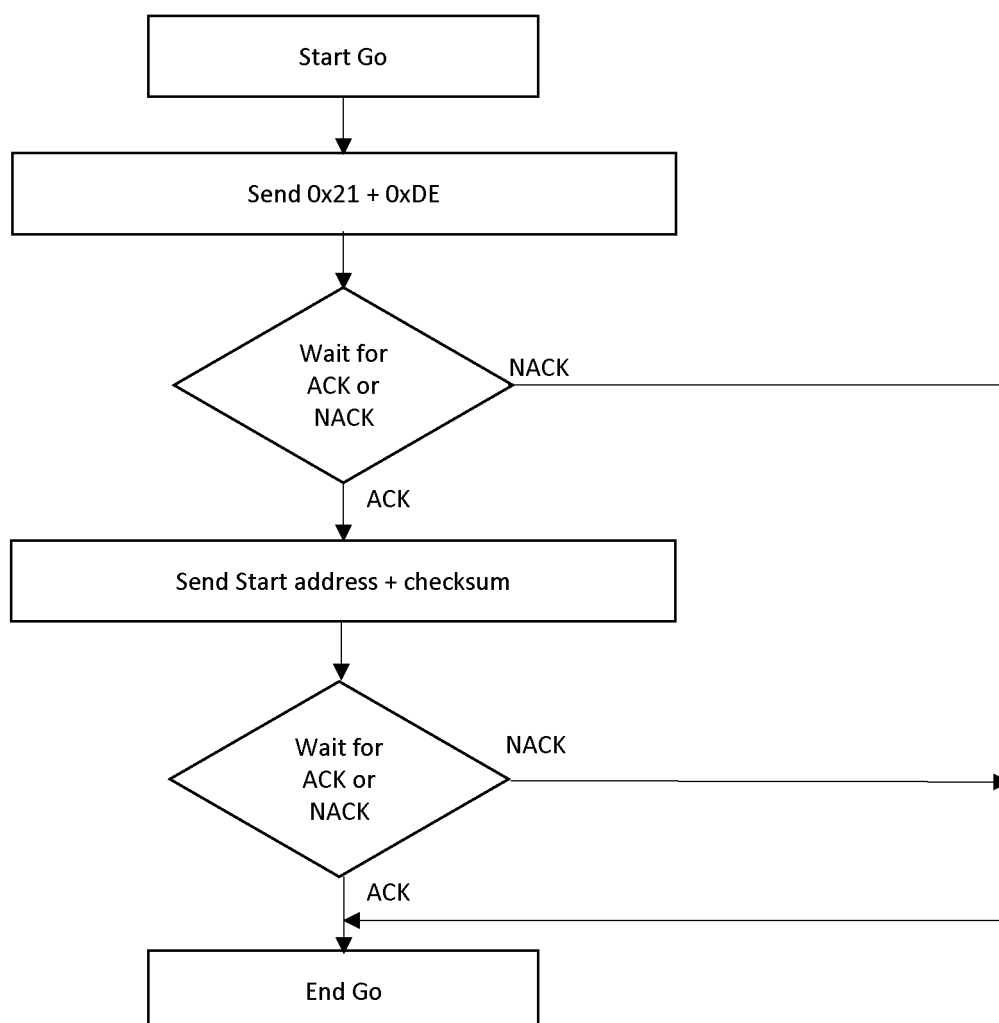
The Go command is used to execute the downloaded code or any other code by jumping to an address specified by the application. When the bootloader receives the Go command, it transmits the ACK byte to the application. Afterwards, the bootloader waits for an address (4 bytes, byte 1 is the address MSB and byte 4 is LSB) and a checksum byte.

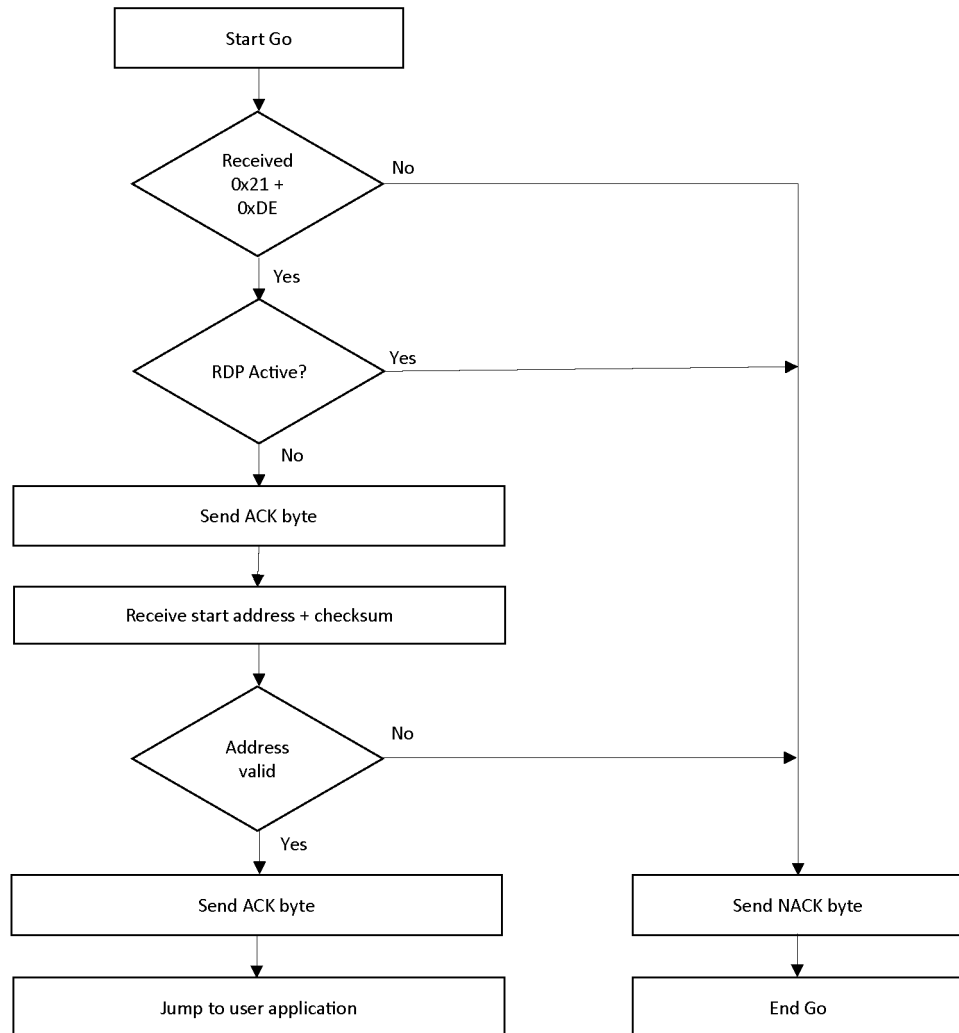
If the address is valid and the checksum is correct, the bootloader transmits an ACK byte, otherwise it transmits a NACK byte and aborts the command.

When the address is valid and the checksum is correct, the bootloader firmware performs the following actions:

- initializes the registers of the peripherals used by the bootloader to their default reset values
- initializes the user application main stack pointer
- jumps to the memory location programmed in the received 'address + 4' (which corresponds to the address of the application reset handler). For example, if the received address is 0x10040000, the bootloader jumps to the memory location programmed at address 0x10040004.

Figure 10. Go command: host side



**Figure 11. Go command: device side**


The host sends bytes to the BlueNRG-1 or BlueNRG-2 as follows:

Byte 1: 0x21

Byte 2: 0xDE

Wait for ACK

Byte 3 – 6: start application address

- Byte 3: MSB
- Byte 6: LSB

Byte 7: checksum: XOR of address bytes

Wait for ACK

### 3.6 Write Memory command

The Write Memory command is used to write data to any valid memory address RAM or Flash memory.

When the bootloader receives the Write Memory command, it transmits the ACK byte to the application.

Afterwards, the bootloader waits for an address (4 bytes, byte 1 is the address MSB and byte 4 is the LSB) and a checksum byte.

If the received address is valid and the checksum is correct, the bootloader transmits an ACK byte, otherwise it transmits a NACK byte and aborts the command.

When the address is valid and the checksum is correct, the bootloader performs these actions:

- gets a byte, N, which contains the number of data bytes to be received
- receives the user data ((N + 1) bytes) and the checksum (XOR of N and of all data bytes)
- programs the user data to memory starting from the received address

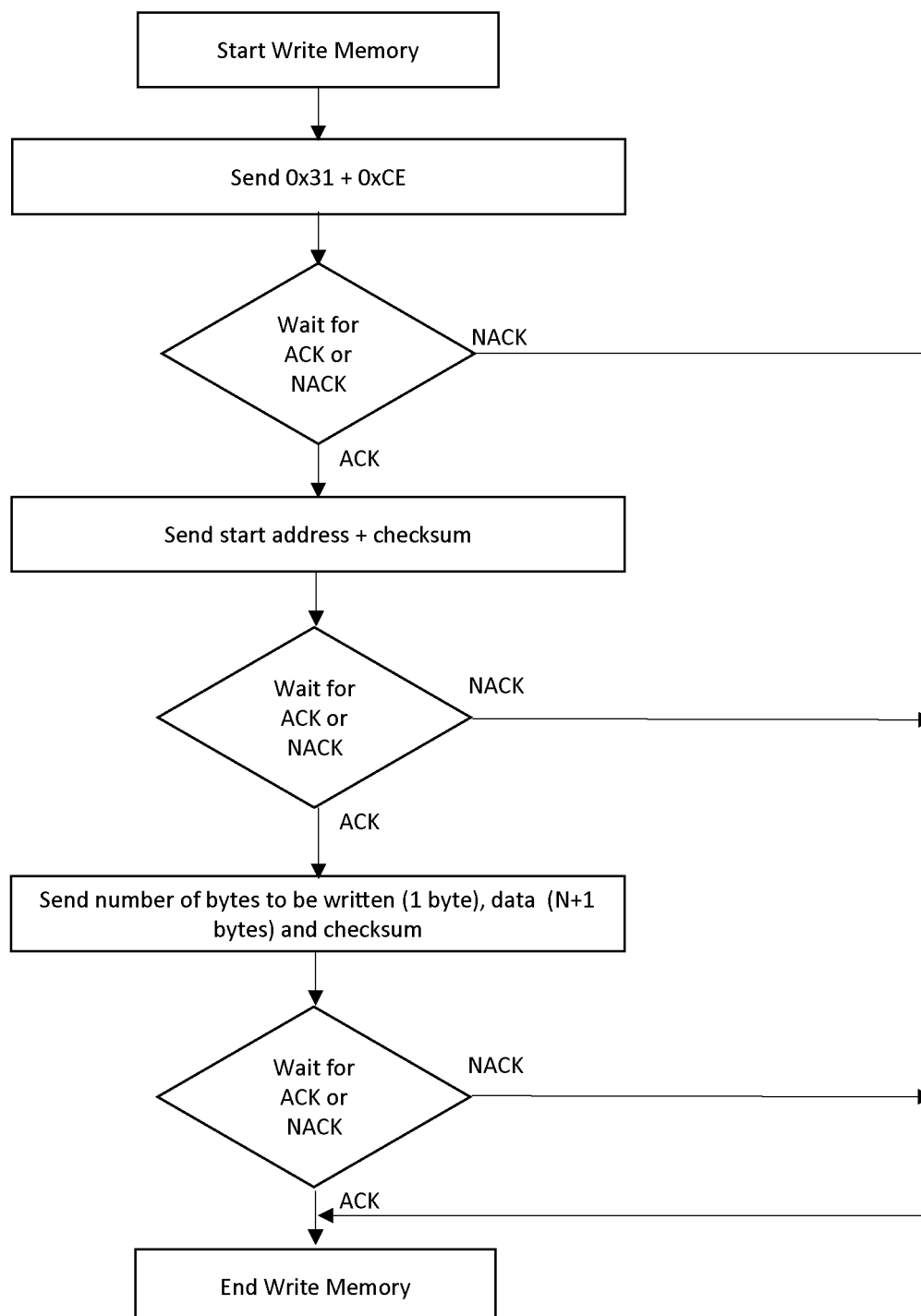
At the end of the command, if the write operation is successful, the bootloader transmits the ACK byte; otherwise it transmits a NACK byte to the application and aborts the command

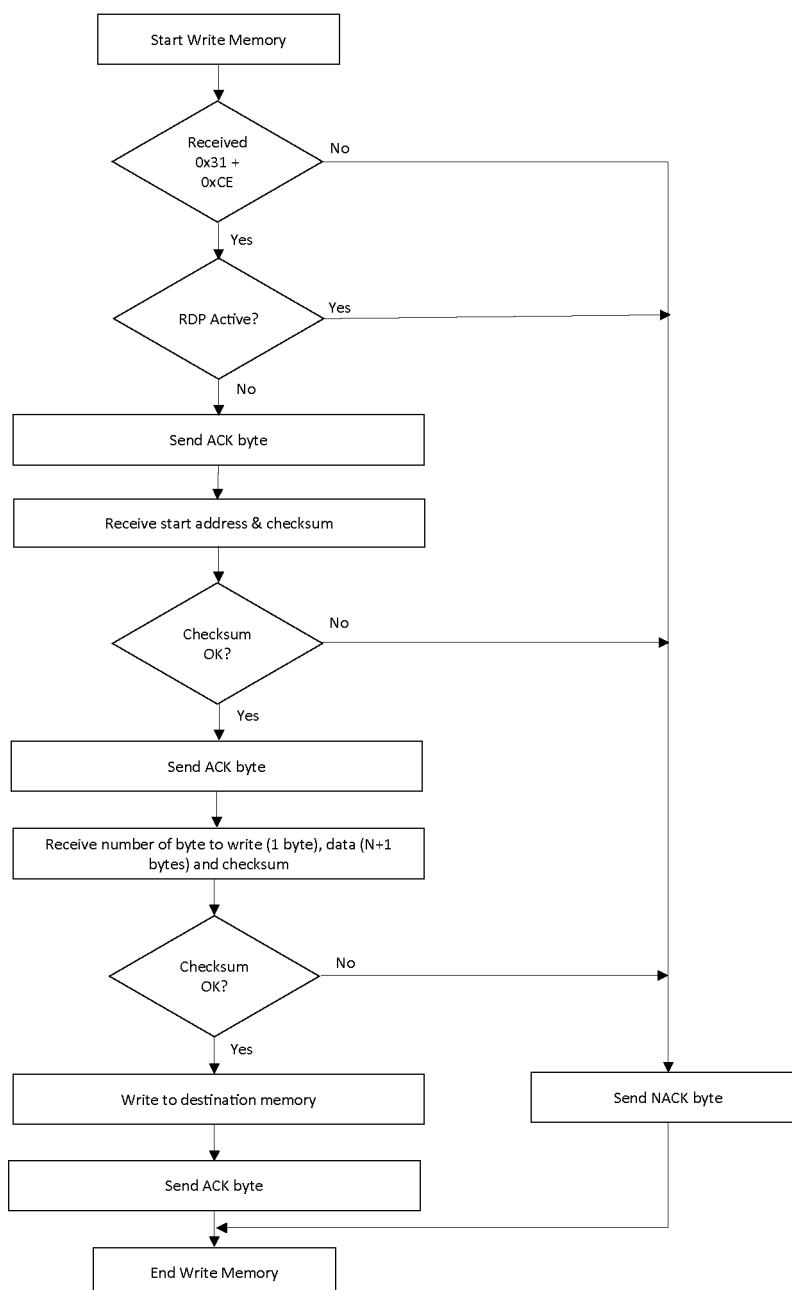
The maximum length of the block to be written for the BlueNRG-1 or BlueNRG-2 is 256 bytes.

If the readout protection is active, a NACK byte is sent to the host when the Write Memory command is received.



Figure 12. Write Memory command: host side



**Figure 13. Write Memory command: device side**


The host sends the bytes to the BlueNRG-1 or BlueNRG-2 as follows:

Byte 1: 0x31

Byte 2: 0xCE

Wait for ACK

Byte 3 to byte 6: start address

- Byte 3: MSB
- Byte 6: LSB

Byte 7: Checksum: XOR (Byte 3, Byte 4, Byte 5, Byte 6)

Wait for ACK

Byte 8: Number of bytes to be received ( $0 < N \leq 255$ )

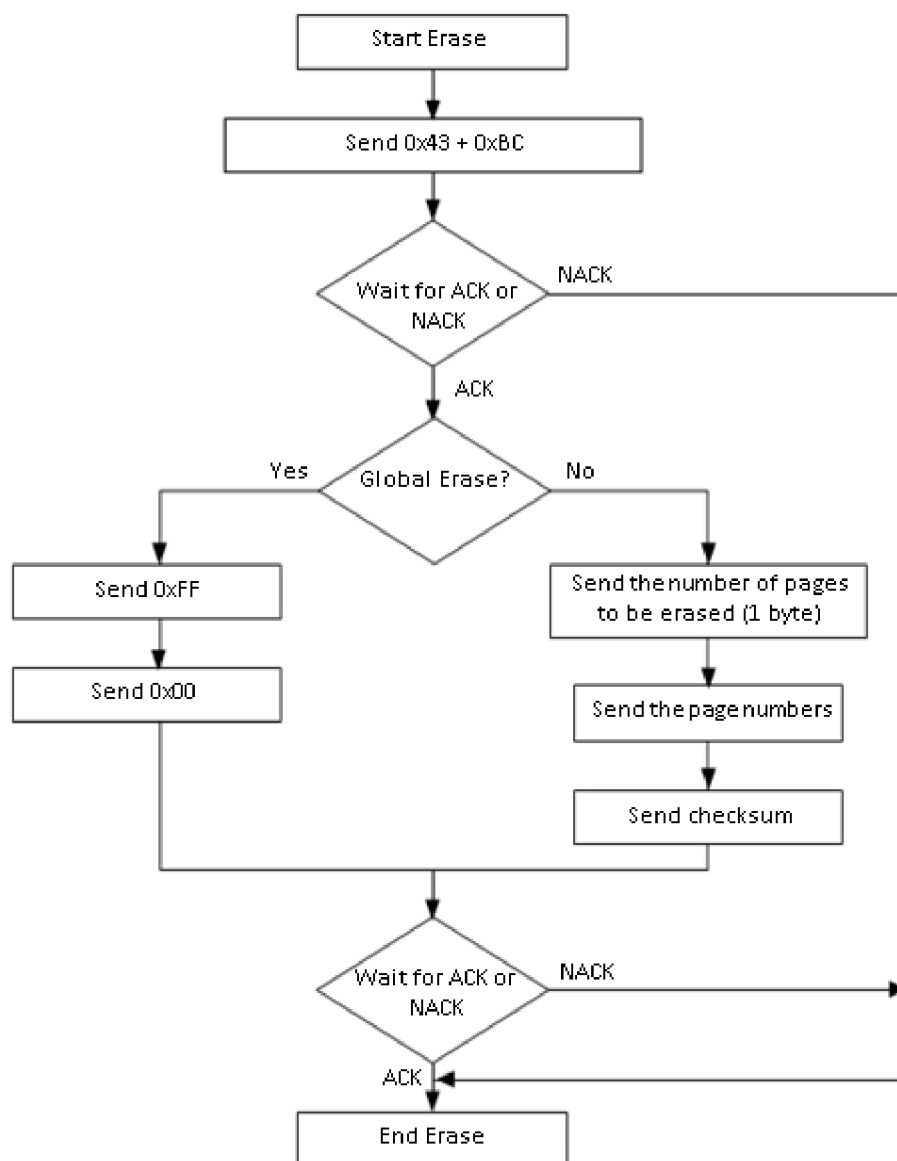
N + 1 data bytes: (Max 256 bytes)  
Checksum byte: XOR (N, N+1 data bytes)  
Wait for ACK

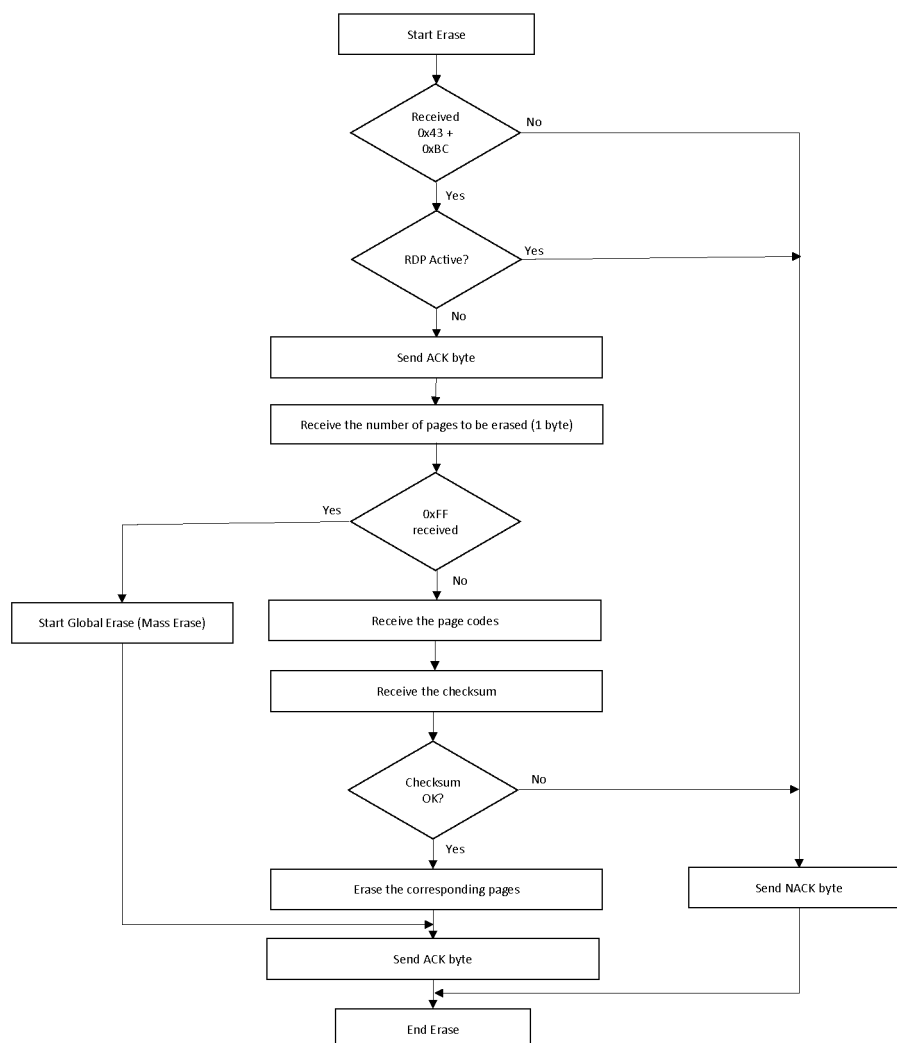
### 3.7 Erase Memory command

The Erase Memory command allows the host to erase Flash memory pages. When the bootloader receives the Erase Memory command, it transmits the ACK byte to the host. Afterwards, the bootloader receives one byte (number of pages to be erased), the Flash memory page codes and a checksum byte. If the checksum is correct then the bootloader erases the memory and sends an ACK byte to the host, otherwise it sends a NACK byte to the host and the command is aborted.

Erase Memory command specifications are:

1. the bootloader receives one byte that contains N, the number of pages to be erased – 1. N= 255 is reserved for mass erase request. For  $0 < N \leq 79$ , N + 1 pages are erased.
2. the bootloader receives (N + 1) bytes, each byte containing a page number

**Figure 14. Erase Memory command: host side**


**Figure 15. Erase Memory command: device side**


The host sends bytes to the BlueNRG-1 or BlueNRG-2 as follows:

Byte 1: 0x43

Byte 2: 0xBC

Wait for ACK

Byte 3: 0xFF or number of pages to be erased – 1 (0 ≤ N ≤ maximum number of pages)

Byte 4: 0x00 (in case of mass erase) or ((N+1 bytes (page numbers) and then checksum XOR(N, N+1 bytes))

Wait for ACK

### 3.8 Readout Protect command

The Readout Protect command is used to enable the Flash memory read protection. When the bootloader receives the Readout Protect command, it transmits the ACK byte to the host. Afterwards, the bootloader enables the read protection for the Flash memory.

At the end of the Readout Protect command, the bootloader transmits the ACK byte to signal the end of the command.

Figure 16. Readout Protect command: host side

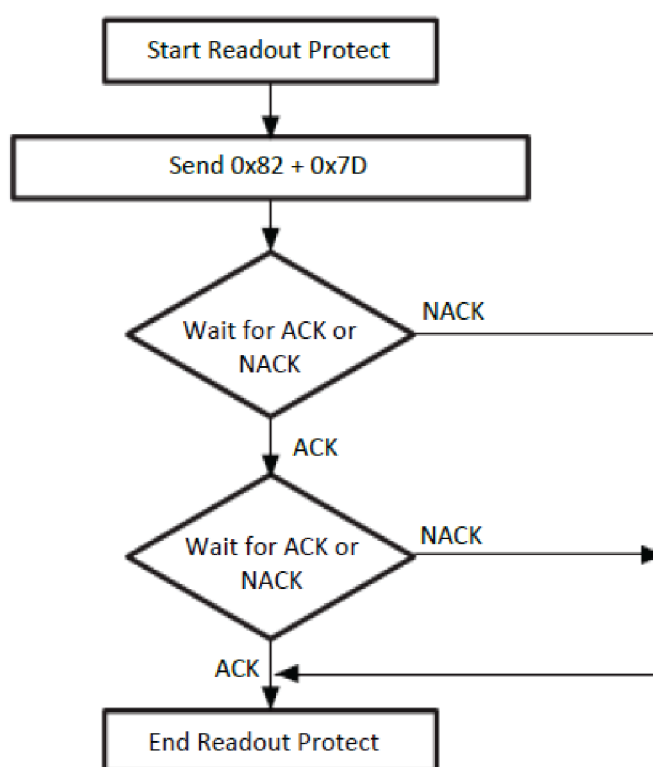
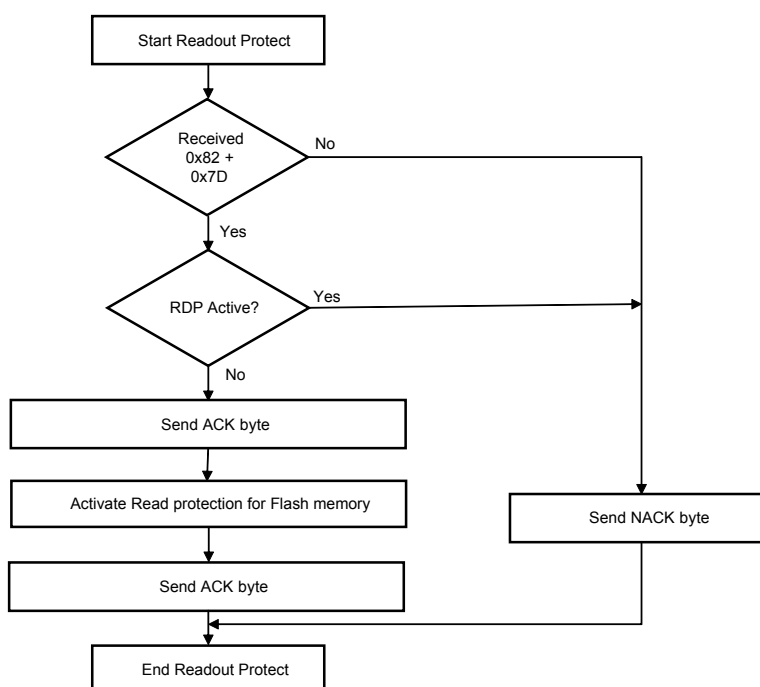


Figure 17. Readout Protect command: device side



### 3.9 Readout Unprotect command

The Readout Unprotect command is used to disable the Flash memory read protection. When the bootloader receives the Readout Unprotect command, it transmits the ACK byte to the host. Afterwards, the bootloader erases all the Flash memory sectors and it disables the read protection for the entire Flash memory.

If the erase operation is successful, the bootloader deactivates the Readout Protection. If the erase operation is unsuccessful, the bootloader transmits a NACK and the read protection remains active.

At the end of the Readout Unprotect command, the bootloader transmits an ACK and generates a system Reset. If other bootloader commands should be executed, the UART bootloader activation needs to be re-executed, that is HW reset plus DIO7 high, like described in the [Section 2 UART bootloader activation](#).

Figure 18. Readout Unprotect command: host side

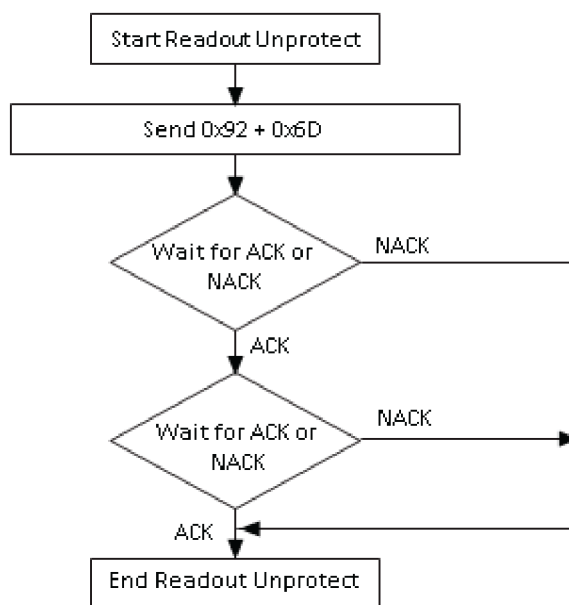
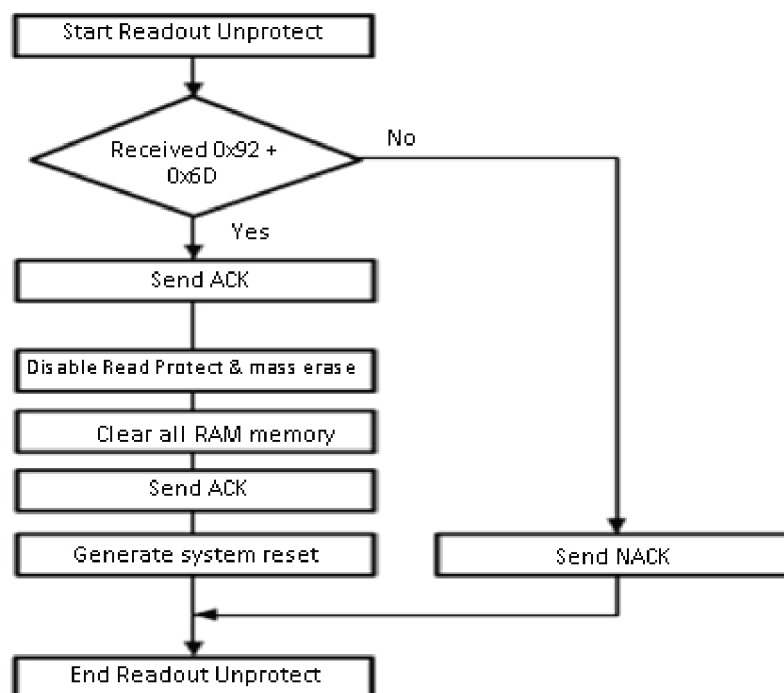


Figure 19. Readout Unprotect command: device side





## Revision history

**Table 2. Document revision history**

| Date        | Version | Changes  |
|-------------|---------|--|
| 06-Jul-2016 | 1       | Initial release.   |
| 06-Jul-2017 | 2       | Throughout text added references to BlueNRG-2.<br>Minor text edits.  |
| 18-Sep-2018 | 3       | Updated <a href="#">Section 3.9 Readout Unprotect command</a> and <a href="#">Section 2 UART bootloader activation</a> . |
| 18-Jan-2021 | 4       | Added reference to the BlueNRG-2N device.<br>Updated <a href="#">Table 1. UART bootloader commands</a> .                 |

## Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>UART bootloader configuration .....</b> | <b>2</b>  |
| <b>2</b> | <b>UART bootloader activation.....</b>     | <b>3</b>  |
| <b>3</b> | <b>UART bootloader commands .....</b>      | <b>4</b>  |
| 3.1      | Get List command .....                     | 4         |
| 3.2      | Get Version command .....                  | 7         |
| 3.3      | Get ID command .....                       | 9         |
| 3.4      | Read Memory command .....                  | 11        |
| 3.5      | Go command .....                           | 13        |
| 3.6      | Write Memory command .....                 | 16        |
| 3.7      | Erase Memory command .....                 | 19        |
| 3.8      | Readout Protect command .....              | 22        |
| 3.9      | Readout Unprotect command .....            | 23        |
|          | <b>Revision history .....</b>              | <b>25</b> |

## List of figures

|                   |   |    |
|-------------------|---|----|
| <b>Figure 1.</b>  | UART bootloader for BlueNRG-1 and BlueNRG-2 devices . . . . . | 3  |
| <b>Figure 2.</b>  | Get List command: host side . . . . .                         | 5  |
| <b>Figure 3.</b>  | Get list command: device side . . . . .                       | 6  |
| <b>Figure 4.</b>  | Get Version command: host side. . . . .                       | 7  |
| <b>Figure 5.</b>  | Get Version command: device side . . . . .                    | 8  |
| <b>Figure 6.</b>  | Get ID command: host side . . . . .                           | 9  |
| <b>Figure 7.</b>  | Get ID command: device side. . . . .                          | 10 |
| <b>Figure 8.</b>  | Read Memory command: host side . . . . .                      | 11 |
| <b>Figure 9.</b>  | Read Memory command: device side . . . . .                    | 12 |
| <b>Figure 10.</b> | Go command: host side . . . . .                               | 14 |
| <b>Figure 11.</b> | Go command: device side . . . . .                             | 15 |
| <b>Figure 12.</b> | Write Memory command: host side . . . . .                     | 17 |
| <b>Figure 13.</b> | Write Memory command: device side . . . . .                   | 18 |
| <b>Figure 14.</b> | Erase Memory command: host side. . . . .                      | 20 |
| <b>Figure 15.</b> | Erase Memory command: device side . . . . .                   | 21 |
| <b>Figure 16.</b> | Readout Protect command: host side . . . . .                  | 22 |
| <b>Figure 17.</b> | Readout Protect command: device side . . . . .                | 23 |
| <b>Figure 18.</b> | Readout Unprotect command: host side. . . . .                 | 24 |
| <b>Figure 19.</b> | Readout Unprotect command: device side . . . . .              | 24 |

**IMPORTANT NOTICE – PLEASE READ CAREFULLY**

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to [www.st.com/trademarks](http://www.st.com/trademarks). All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2021 STMicroelectronics – All rights reserved