Application Note: JN-AN-1003

Boot Loader Operation

All Jennic JN51xx wireless microcontrollers have a ROM boot loader. This boot loader is executed by the CPU following a reset, on waking from sleep or deep sleep mode, and when power is applied. It is designed to operate with external serial memory (normally Flash memory) connected to the SPI interface, loads the user application into RAM and starts program execution.

This Application Note describes the functionality of the boot loader, covering the following topics:

- **Operation of the ROM boot loader after a cold start**
- **Operation of the ROM boot loader after a warm start**
- **The Flash image application header**
- **The serial interface used to communicate with the boot loader**

Boot Loader Operation

When started, the boot loader performs one of the following operations

- Starts execution of the application, already present in RAM
- Copies an application from the external Flash memory device (starting at Flash memory location 0) to RAM and starts application execution
- Enters programming mode

The application must contain two entry points from which execution may begin - a wake-up point and a reset point. The locations of these points are stored in the header information contained within Flash memory.

If the application preserves the contents of the internal RAM when the device goes to sleep via the **vAHI_Sleep()** function, passing either E_AHI_SLEEP_OSCON_RAMON or E_AHI_SLEEP_OSCOFF_RAMON, then the boot loader will restart the application from the wake-up entry point rather than the reset entry point.

If the application is not present in RAM, the boot loader will attempt to load the application from the external Flash memory. If the boot loader finds a valid application at address 0 in Flash memory, this application will be copied into RAM.

Alternatively, the boot loader may enter programming mode if the SPI MISO line is held low when the device is reset, or if no valid application image is found in the external Flash memory.

JN5121 Operation

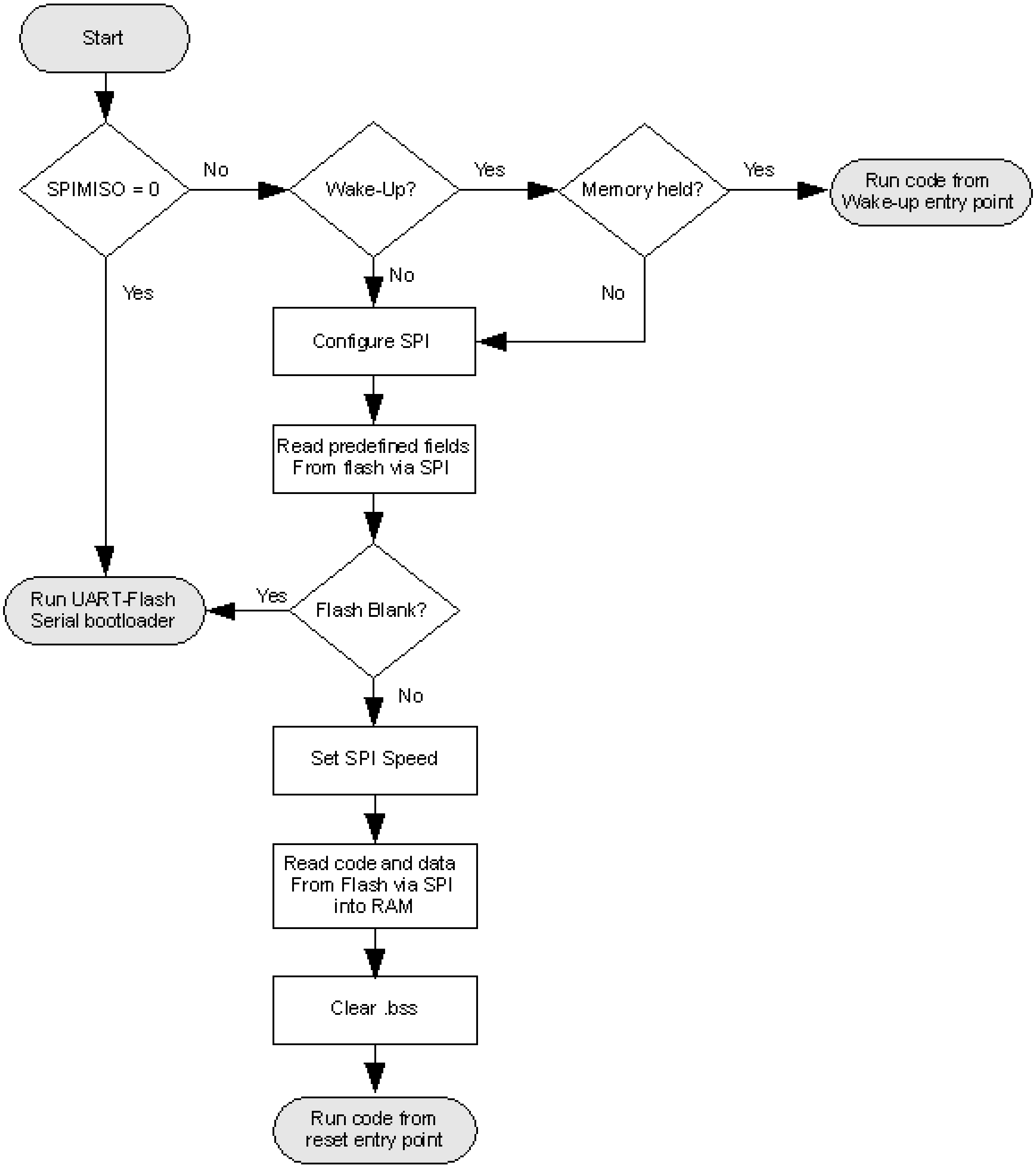

Figure 1: JN5121 Boot Loader Operation

JN5139 Operation

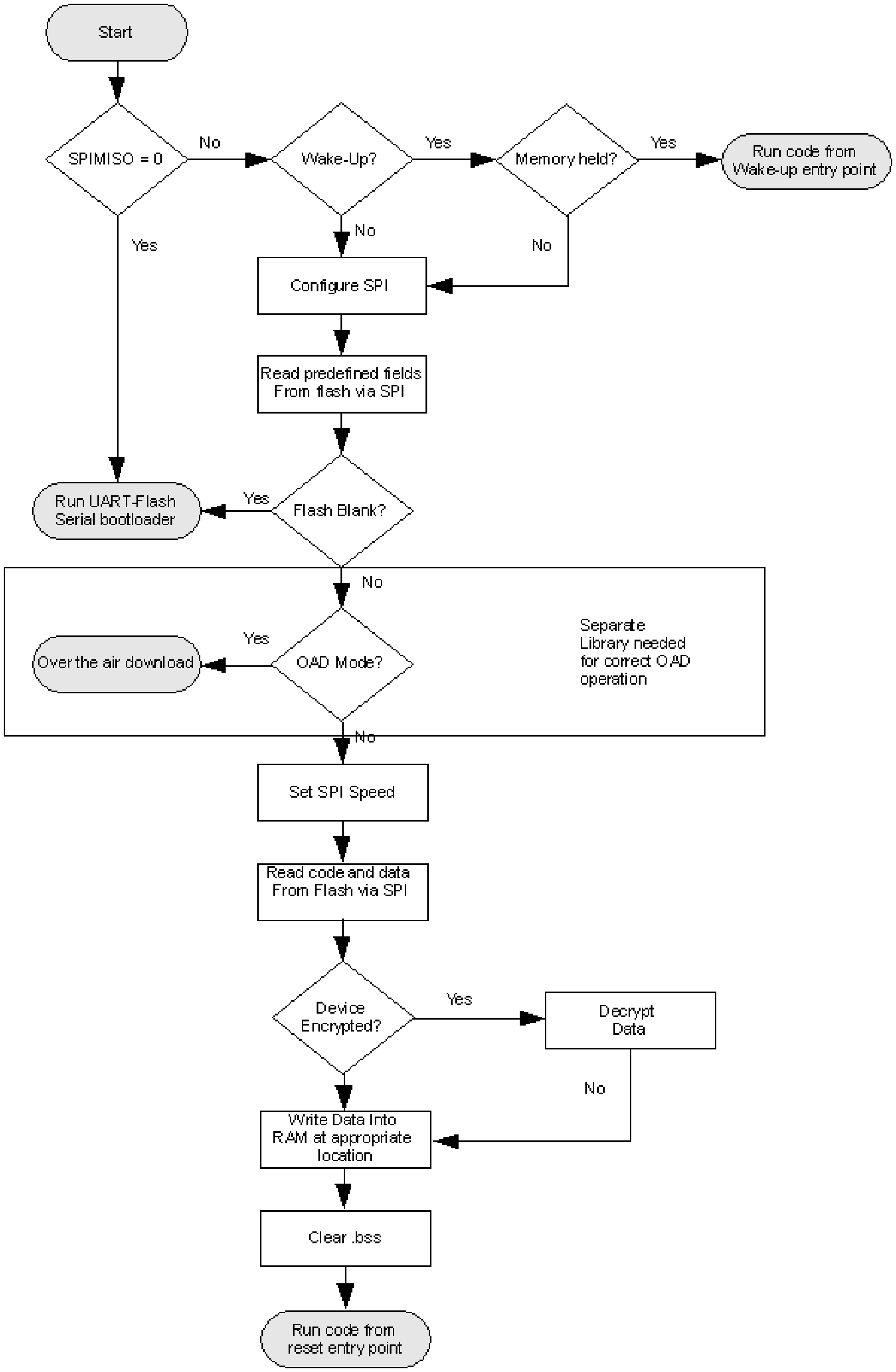

Figure 2: JN5139 Boot Loader Operation

JN5148 Operation

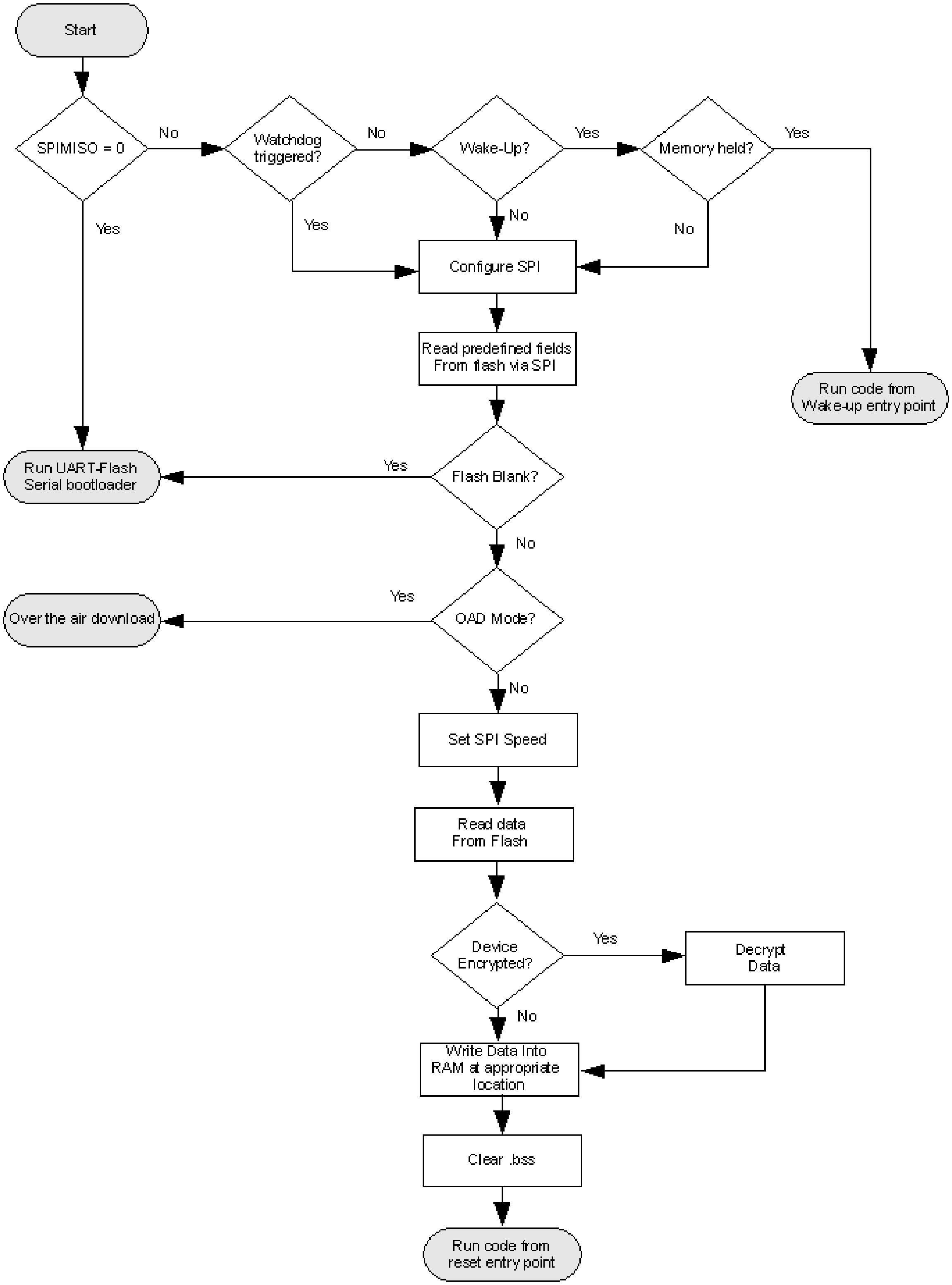

Figure 3: JN5148 Boot Loader Operation

Flash Header

Any user application programmed into the Flash memory device must include a header that contains information about the program and the device in which it is stored. The format of this header is detailed below:

JN5121

Bytes	Word	Contents
0x0000 to 0x0003	0	Configuration bytes: 0xAABBCCDD AA - Reserved (always set to 7) BB - Reserved (AA repeat, always 7) CC - Flash access configuration byte (see below for details) DD - Flash access configuration byte (repeated)
0x0004 to 0x0007	1	The start address in RAM where the application data needs to be copied to (word-aligned). The data copied to this address will start from Flash address 0x24
0x0008 to 0x000B	2	Length of data to copy to RAM from Flash address 0x30 (includes sections: .mac, .rodata, .data, .text, .bss)
0x000C to 0x000F	3	Chip version for which the binary was compiled
0x0010 to 0x0013	4	Must be 0
0x0014 to 0x0017	5	Start address for .bss segment in RAM (word-aligned)
0x0018 to 0x001B	6	Length of .bss segment in RAM
0x001C to 0x001F	7	Wake-up entry point (word-aligned) (address of **AppWarmStart()**)
0x0020 to 0x0023	8	Reset entry point (word-aligned) (address of **AppColdStart()**)
0x0024 to End		Application data The first 36 bytes contain an 8-byte MAC address and 28 bytes for ZigBee use

Table 1: JN5121 Flash Header

The Flash header is defined at compile/link time by the chip-specific linker file located in the SDK directory at **Chip\JN5121**.

Flash Access Configuration Byte

If the first Flash access configuration byte is 0xFF, this indicates a blank Flash memory. Otherwise, the Flash access configuration byte is interpreted as follows:

Bit	Contents
7:6	Always set to 3
5	Address field supported: 0 for 16-bit address 1 for 24-bit address
4:0	SPI clock divider value to be used. This will be used to set the SPI divider setting in the SPI master, padded with a 0 as the MSB of the 6-bit divider field (0x1F is not a valid setting for this field)

The first four bytes will be read from Flash memory with a SPI clock speed of 1 MHz. The boot loader will use the SPI clock divider from the Flash access configuration byte to set the SPI clock for the remaining reads.

JN5139

Bytes	Word	Contents
0x0000 to 0x0003	0	Configuration bytes: 0xAABBCCDD AA - Reserved (always Set to 7) BB - Reserved (AA repeat, always 7) CC - Flash access configuration byte (see below for details) DD - Flash access configuration byte (repeated)
0x0004 to 0x0007	1	The start address in RAM where the application data needs to be copied to (word-aligned). The data copied to this address will start from Flash address 0x30
0x0008 to 0x000B	2	Length of data to copy to RAM from Flash address 0x30 (includes sections: .mac, .rodata, .data, .text, .bss)
0x000C to 0x000F	3	Chip ROM version for which the binary was compiled
0x0010 to 0x0013	4	Must be 0
0x0014 to 0x0017	5	Start address for .bss segment in RAM (word-aligned)
0x0018 to 0x001B	6	Length of .bss segment in RAM
0x001C to 0x001F	7	Wake-up entry point (word-aligned) (address of **AppWarmStart()**)
0x0020 to 0x0023	8	Reset entry point (word-aligned) (address of **AppColdStart()**)
0x0024 to 0x0027	9	OAD channel scan
0x0028 to 0x0029	10	OAD channel scan bitmap
0x002a to 0x002b		OAD application/server identification
0x002c to 0x002d	11	Unused
0x002e		Revert and valid flags for OAD
0x002f		Invalid flags for OAD
0x0030 to End		Application data The first 36 bytes contain an 8-byte MAC address and 28 bytes for ZigBee use

Table 2: JN5139 and JN5148 Flash Header

The Flash header is defined at compile/link time by the chip-specific linker file located in the SDK directory at **Chip\JN513x\Build**.

Flash Access Configuration Byte

If the first Flash access configuration byte is 0xFF, this indicates a blank Flash memory. Otherwise, the Flash access configuration byte is interpreted as follows:

Bit	Contents
7:6	Always set to 3
5	Address field supported: 0 for 16-bit address 1 for 24-bit address
4:0	SPI clock divider value to be used. This will be used to set the SPI divider setting in the SPI master, padded with a 0 as the MSB of the 6-bit divider field (0x1F is not a valid setting for this field)

The first four bytes will be read from Flash memory with a SPI clock speed of 1 MHz. The boot loader will then use the SPI clock divider from the Flash access configuration byte to set the SPI clock for the remaining reads.

JN5148

The JN5148 Flash header is the same as for the JN5139 except for the reserved fields at Flash addresses 0x00 and 0x01, which should always be 0. For the header details, refer to Table 2.

The Flash header is defined at compile/link time by the chip-specific linker file located in the SDK directory at **Chip\JN5148\Build**.

External (Flash) Memory

The external serial memory must be connected to the SPI port using the SPISEL0 select line.

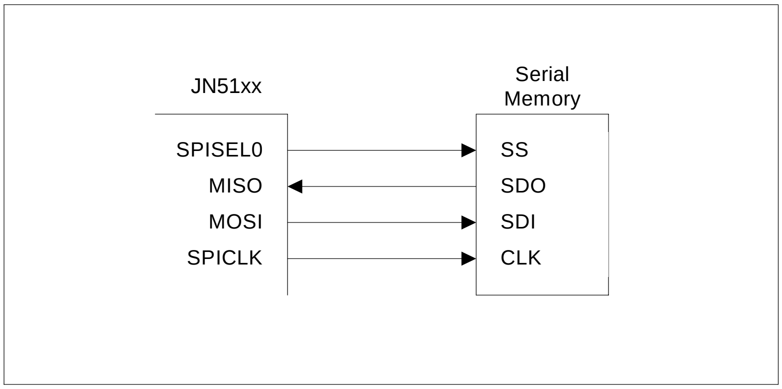

Figure 4: Connecting External Serial Memory

JN5121 Supported Devices

The JN5121 only supports the 128-Kbyte ST M25P10-A Flash device.

JN5139 Supported Devices

The JN5139 supports the following devices:

- SST 25VF0101 (128 Kbytes)
- ATMEL AT25F512 (64 Kbytes)
- ST M25P10 A (128 Kbytes)

JN5148 Supported Devices

The JN5148 supports the following devices:

- SST 25VF0101 (128 Kbytes)
- ATMEL AT25F512 (64 Kbytes)
- ST M25P10 A (128 Kbytes)
- ST M25P40 (512 Kbytes)

Memory Map

The JN5121 and JN5139 have 96 Kbytes of RAM, and the JN5148 has 128 Kbytes of RAM.

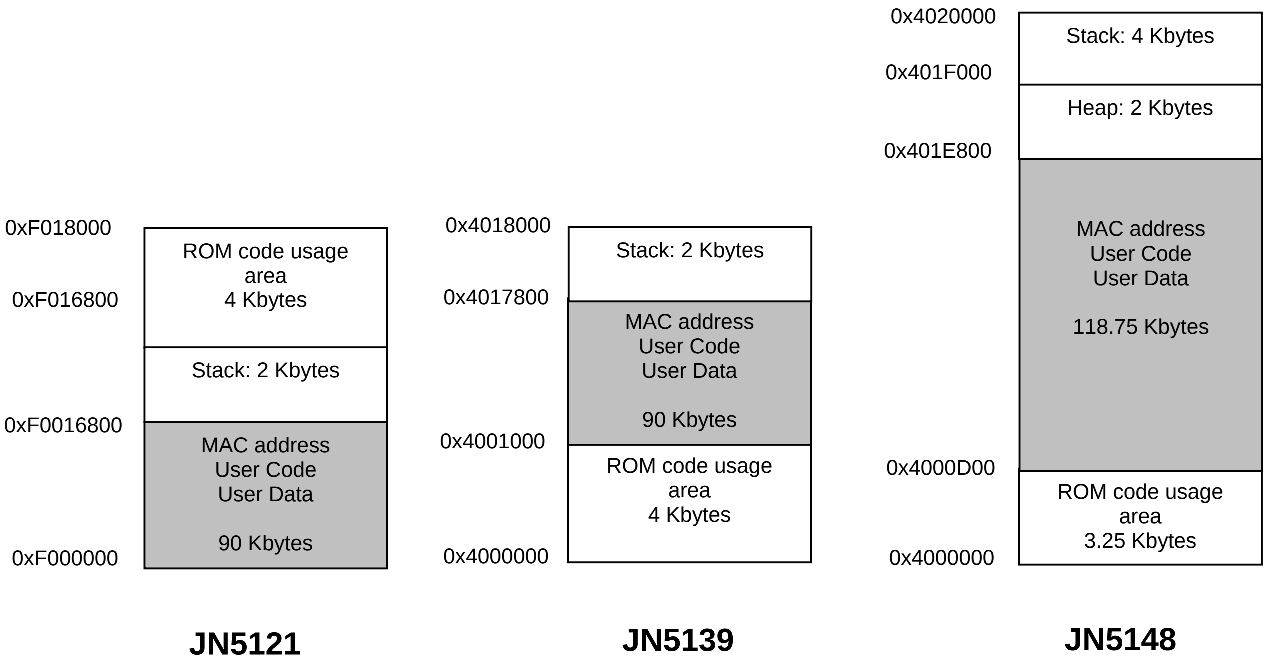

Figure 5: JN51xx Memory Maps

Serial Protocol

The protocol defines a message structure and set of messages that allow a client to communicate with the JN51xx chip, and hence perform operations on the external Flash memory. Serial communication between the PC and boot loader is implemented as 8,N,1,38K4 from the JN51xx UART 0.

The Jennic device will respond to the serial protocol when it has entered programming mode, either due to a reset when the MISO line is asserted of if the Flash chip is deemed to be blank.

Message Structure

A message contains four fields and is structured as follows:

Field	**1**	**2**	**3**	**4**
Bytes	1	1	(x)	1
Data	Length	Message Type	Message Data	Checksum

Table 3: Message Structure

Length The number of bytes in the rest of the message, including the checksum. Note that the maximum total message size is 255.

Message Type Defined below.

Message Data Variable length and dependent on message type.
May be zero-length (i.e. not present)

Note: For Flash/RAM Read/Write requests, a maximum of 128 bytes should be read or written.

Checksum Calculated over the preceding part of the message, including the Length field, by implementing a logical Exclusive-OR operation on the previous bytes.

Byte Transfer Timeout

A timeout is applied to the reception of each individual byte of a message. This timeout depends on the CPU clock frequency. When using the 16-MHz clock, in your download program you are advised not to allow more than 5 seconds between bytes transmitted.

Message Types

Message types are as follows (N/A means that an optional field is not present):

Message Type	Meaning	Originator	Supported Devices	# Parms	Message Parameters	Response Message Type
0x00 to 0x06	Reserved	N/A	N/A	N/A	N/A	N/A
0x07	Flash erase request (All Sectors)	PC	JN5121 JN5139 JN5148	0	N/A	0x08
0x08	Flash erase response	JN51xx	JN5121 JN5139 JN5148	1	#1: Status (1 byte): 0x00: Erase successful 0xFF: Parameter error 0xF7: Auth error	N/A
0x09	Flash program request	PC	JN5121 JN5139 JN5148	2	#1: Flash address (4 bytes), LSB first #2: Up to 128 bytes of data **Note:** Data should not span page write boundary as defined by the Flash specification.	0x0A
0x0A	Flash program response	JN51xx	JN5121 JN5139 JN5148	1	#1: Status (1 byte): 0x00: Program successful 0xFF: Readback verify failed 0xF7: Auth error	N/A
0x0B	Flash read request	PC	JN5121 JN5139 JN5148	2	#1: Flash address (4 bytes), LSB first #2: Length (2 bytes), LSB first (Length should not be greater than 128 bytes)	0x0C
0x0C	Flash read response	JN51xx	JN5121 JN5139 JN5148	2	#1: Status (1 byte): 0x00 (success) 0xF7 (Auth error) *#2:* Bytes read from Flash	N/A
0x0D	Sector erase request	PC	JN5121 JN5139 JN5148	1	#1: Sector to erase (1 Byte) **Note:** Ensure the write protection is disabled by writing 0 to the status register using message 0x0F.	0x0E
0x0E	Sector erase response	JN51xx	JN5121 JN5139 JN5148	1	#1: Status (1 byte): 0x00: Erase successful 0xFF: Erase error 0xF7: Auth error	N/A
0x0F	Write SR (status register) request	PC	JN5121 JN5139 JN5148	1	#1: SR Value (1 byte)	0x10
0x10	Write SR response	JN51xx	JN5121 JN5139 JN5148	1	#1: Status (1 byte): 0x00: Write successful 0xFF: Write error 0xF7: Auth error	N/A

Message Type	Meaning	Originator	Supported Devices	# Parms	Message Parameters	Response Message Type
0x1D	RAM write request	PC	JN5121 JN5139 JN5148	2	#1: RAM address (4 bytes), LSB first *#2:* Bytes to write into RAM (Up to 128 bytes)	0x1E
0x1E	RAM write response	JN51xx	JN5121 JN5139 JN5148	1	#1: Status (1 byte): 0x00 (success) 0xF7 (Auth error)	N/A
0x1F	RAM read request	PC	JN5121 JN5139 JN5148	2	#1: RAM address (4 bytes), LSB first #2: No. of bytes to read (2 bytes), LSB first (should not be greater than 128 bytes)	0x20
0x20	RAM read response	JN51xx	JN5121 JN5139 JN5148	2	#1: Status (1 byte): 0x00 (success) 0xF7 (Auth error) *#2: x* bytes read from RAM	N/A
0x21	Run request	PC	JN5121 JN5139 JN5148	1	#1: Address to jump to (4 bytes), LSB first	0x22
0x22	Run response	JN51xx	JN5121 JN5139 JN5148	1	#1: Status (1 byte): 0x00 (success) 0xF7 (Auth error)	N/A
0x25	Read Flash ID request	PC	JN5139 JN5148	0	N/A	0x26
0x26	Read Flash ID response	JN51xx	JN5139 JN5148	3	#1: Status (1 byte): 0x00 (success) 0xF7 (Auth error) #2: Flash manufacturer ID (1 byte) #3: Flash device ID (1 byte)	N/A
0x27	Change Baud Rate Request	PC	JN5139 JN5148	1	#1: Serial Clock Divisor 1 = 1000000 2 = 500000 9 = 115200 (approx.) 26 = 38400 (approx.)	N/A
0x28	Change Baud Rate Response	JN51xx	JN5139 JN5148	1	#1: Status (1 byte): 0x00 (success)	0x27
0x2C	Select Flash type	PC	JN5139 JN5148	2	#1 : Flash type (1 byte) (see table below for mapping) #2: Custom programming jump address (4 bytes), LSB first (set to 0000),	0x2D
0x2D	Select Flash type response	JN51xx	JN5139 JN5148	1	#1 byte: Status	N/A
0x32	Get Chip ID Request	PC	JN5148	0	N/A **Note:** The Chip ID for the JN5121 and JN5139 can be obtained by a RAM read on address 0x100000FC	N/A

Message Type	Meaning	Originator	Supported Devices	# Parms	Message Parameters	Response Message Type
0x33	Get Chip ID Response	JN51xx	JN5148	2	#1: Status (1 byte): 0x00 (success) #2: Chip ID (4 bytes) (MSB first): JN5121: 0x20000000 JN5139R: 0x00001000 JN5139(R1): 0x10002000 JN5148R1: 0x10404686 **Note:** The Chip ID for the JN5121 and JN5139 can be obtained by a RAM read of 4 bytes from address 0x100000FC	0x32

Table 4: Message Types

Mapping Flash IDs to Flash Device Types

The JN5139 and JN5148 devices support multiple Flash device types. It is the responsibility of the PC application to instruct the boot loader which Flash device to use. To determine the Flash device type connected, issue a Read Flash ID request (0x25). The returned Manufacture ID and Device ID can then be mapped to a Jennic ID, according to the table below.

Manufacturer ID	Device ID	Type	Jennic ID
0x10	0x10	ST M25P10-A	0
0xBF	0x49	SST 25VF010A	1
0x1F	0x60	Atmel 25F512	2
0x12	0x12	ST M25P40	3

Table 5: Flash Mappings

Response Status Code

Status Type	ID
OK	0
Not supported	0xFF
Write fail	0xFE
Invalid response	0xFD
CRC error	0xFC
Assert fail	0xFB
User interrupt	0xFA
Read fail	0xF9
TST error	0xF8
Auth error	0xF7
No response	0xF6

Table 6: Response Status Codes

Binary Version

When a binary file is built, the ROM version for which it was built is stored at offset 0x0C in the binary file. This is configured in the linker configuration file stored in the chip build directory, i.e. **Chip\JN513x\Build\AppBuild_JN5139.ld** or **Chip\JN5148\Build \AppBuild_JN5148.ld**.

To verify that the binary file is correct for the target device, the boot loader should perform a RAM read of 4 bytes from address 0x00000004. If the two versions do not match then the binary file has been built for the wrong target.

Example Programming Sequence

On the JN5139 and JN5148 devices, the PC application must dynamically determine which Flash device is connected to the wireless microcontroller. This is achieved by issuing the FL_READ_ID_REQ message. The response will give the Manufacturer ID and Device ID of the Flash device attached.

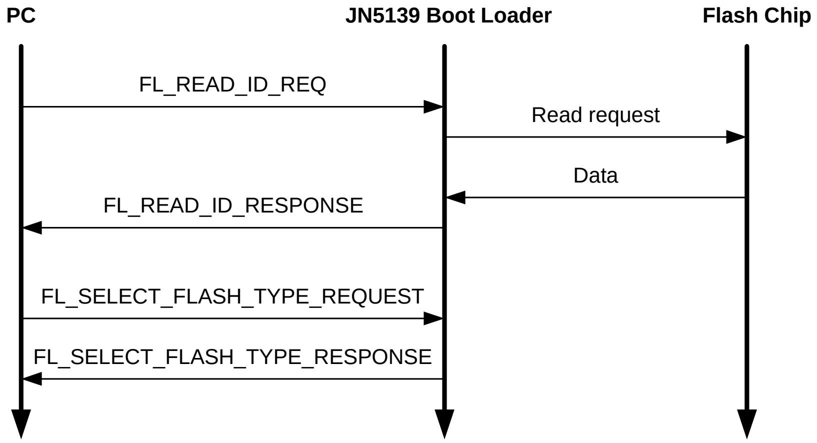

Figure 6: Flash Device Type Selection Sequence

Once the Flash device type is known, the type can be set within the boot loader by sending a FL_SELECT_FLASH_TYPE_REQUEST.

After the Flash type has been selected, the programming sequence is the same for all JN51xx devices, as illustrated below.

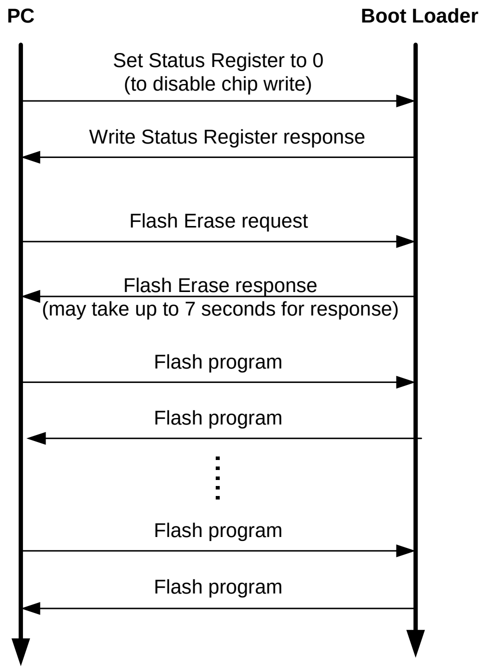

Figure 7: Flash Programming Sequence

Revision History

Version	Notes
1.0	Initial Release
1.1	Corrected Flowchart
1.2	Added Boot Loader Timing Parameters
1.3	Re-templated
1.4	Corrected Flowchart
1.5	- New format - Corrected JN5121 details - Added information for the JN5139 and JN5148 devices - Merged with JN-AN-1007 “Boot Loader Serial Protocol”

Important Notice

Jennic reserves the right to make corrections, modifications, enhancements, improvements and other changes to its products and services at any time, and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders, and should verify that such information is current and complete. All products are sold subject to Jennic’s terms and conditions of sale, supplied at the time of order acknowledgment. Information relating to device applications, and the like, is intended as suggestion only and may be superseded by updates. It is the customer’s responsibility to ensure that their application meets their own specifications. Jennic makes no representation and gives no warranty relating to advice, support or customer product design.

Jennic assumes no responsibility or liability for the use of any of its products, conveys no license or title under any patent, copyright or mask work rights to these products, and makes no representations or warranties that these products are free from patent, copyright or mask work infringement, unless otherwise specified.

Jennic products are not intended for use in life support systems/appliances or any systems where product malfunction can reasonably be expected to result in personal injury, death, severe property damage or environmental damage. Jennic customers using or selling Jennic products for use in such applications do so at their own risk and agree to fully indemnify Jennic for any damages resulting from such use.

All trademarks are the property of their respective owners.

Jennic Ltd
Furnival Street
Sheffield
S1 4QT
United Kingdom

Tel: +44 (0)114 281 2655
Fax: +44 (0)114 281 2951
E-mail: info@jennic.com

For the contact details of your local Jennic office or distributor, refer to the Jennic web site:

www.Jennic.com
TECHNOLOGY FOR A CHANGING WORLD