Application Note: JN-AN-1001

Calculating JN516x Power Consumption

This Application Note describes how to calculate the power required by an NXP JN516x wireless microcontroller running an IEEE 802.15.4-based application. The formulae necessary to calculate the time required to complete common 802.15.4 network operations are presented in conjunction with the electrical current consumed by the JN516x devices during these operations. Additionally, timings and current consumption information are provided for specific tasks. An example is included that describes how the information presented can be used to determine the battery capacity requirements of an application.

IEEE 802.15.4 Network Operations

The timings of all 802.15.4 operations are based on one fundamental piece of information, the symbol rate. When operating in the 2.4-GHz band, the symbol rate is defined as 62500 symbols per second. Each symbol comprises four bits of information and, therefore, the over-air data rate is 250 kbps. Below are the formulae required to calculate the time taken to perform common 802.15.4 network operations, based on this data rate.

Energy Detection (ED) Scan

This is dependent on the number of channels to be scanned and the time for which each one is scanned, both of which are selected by the user.

NumberOfChannelsScanned = 1 to 16

ScanDuration = 0 to 14

ED Scan Period (ms) = [960 x ($2^{ScanDuration}$ + 1) / 62.5] x NumberOfChannelsScanned

Data Transmission

The maximum size of a data frame is 127 bytes, of which 6 bytes form the Physical layer header and 9 to 25 bytes form the MAC layer header. The size of the MAC layer header is dependent on the addressing mode used; see the table below for details.

Addressing Mode*	MAC Header Size (bytes)
No source or destination address (only PAN ID included)	9
16-bit source and destination addresses	13
64-bit source and destination addresses	25

* Further addressing mode combinations are available; see the IEEE 802.15.4 Standard for details.

The remaining 96 to 112 bytes can be used for the transmission of data. The total time taken to transmit a single data frame can be calculated as follows:

PayloadSize (bytes) = 1 to 112

HeaderSize (bytes) = 15 to 31

Data Frame Transmission Period (ms) = (HeaderSize + PayloadSize) x 8 / 250

Beacon Frame Transmission

The maximum size of a beacon frame is 127 bytes. This comprises a Physical layer header, a MAC layer header, beacon overhead and the beacon payload. The Physical layer header size is fixed at 6 bytes. The size of the MAC layer header is dependent on the addressing mode used; see the table below for details.

Addressing Mode	MAC Header Size (bytes)
16-bit short address	11
64-bit address	17

The beacon overhead depends on the number of GTSs used and the number of pending address fields. The beacon payload can take up the remaining bytes and is available for the transmission of application-specific data.

PHYHeaderSize = 6 bytes

MACHeaderSize = 11 or 17 bytes

BeaconOverhead = 3 to 52 bytes

BeaconPayload =

0 to (127 – PHYHeaderSize – MACHeaderSize – BeaconOverhead) bytes

Beacon Frame Transmission Period (ms) =

(PHYHeaderSize + MACHeaderSize + BeaconOverhead + BeaconPayload) x 8 / 250

CSMA/CA Channel Access

Access to the radio channel is gained using the CSMA/CA algorithm. Before a data frame is transmitted, a Clear Channel Assessment (CCA) is performed to determine if the channel is currently in use. The CCA takes 8 symbol periods (0.128 ms) to complete. The time at which the CCA is performed and how frequently it is repeated (if the channel is found to be busy) are determined by the type of CSMA/CA algorithm used.

If regular beacons are not used then data is transmitted using unslotted CSMA/CA. This algorithm is based on back-off periods, where one back-off period is equal to 20 symbol periods. Two variables are used by the algorithm:

- *NB* is the number of times the algorithm has been required to back off while attempting the current transmission; this is initialised to zero.
- *BE* is the back-off exponent and is related to how many back-off periods a device will wait before attempting to assess the channel – the maximum number of back-off periods is given by (2^{BE} – 1). *BE* is initialised to *macMinBE* (in the range 0 to 3) and has a default value of 3.

The unslotted CSMA/CA algorithm works as follows:

1. *NB* is set to 0, *BE* is set to *macMinBE*
2. Delay implemented for a random period in the range 0 to (2^{BE} – 1) back-off periods
3. CCA is performed
4. If the channel is idle then data is transmitted, otherwise algorithm continues to Step 5
5. *NB* is set to *NB* + 1, *BE* is set to min(*BE* + 1, *aMaxBE*)
6. If *NB* is greater than *macMaxCSMABackoffs* then the transmission has failed, otherwise algorithm returns to Step 1

The time between requesting that a data frame is transmitted and it actually being transmitted is therefore dependent on the back-off period and the result of the CCA. The time taken to complete a single back-off period can be calculated as follows:

$$\text{BackoffPeriod (ms)} = (2^{BE} - 1) \times 20 / 62.5$$

Data Reception

The time taken to receive a data frame is exactly the same as the time taken to transmit it. However, it should be noted that the receiver must be switched on at some point before the data frame is expected to arrive. How long before depends on the type of network being used (beacon or non-beacon enabled) and how the application uses the network. An examination of this is beyond the scope of this document.

JN516x Current Consumption

This section contains information regarding the current consumption of the JN516x while performing various operations. All quoted currents are typical values at 3V and 25°C.

Operation	Current (mA)	Notes
Active Processing		
CPU active (Radio off)	1.7 + 0.205/ MHz	CPU can run at 32, 16, 8, 4, 2 or 1 MHz. GPIOs enabled. When in CPU Doze mode, the current related to CPU speed is not consumed.
Radio transmitting	15.3	CPU in Doze mode
Radio receiving	17.0	CPU in Doze mode
The following current figures should be added to those above if the feature is being used		
ADC	0.555	Temperature sensor and battery measurements require ADC
Comparator	0.073/0.0008	Normal/Low power
UART	0.06	For each UART
Timer	0.021	For each timer
2-wire Serial Interface	0.046	
Sleep Modes		
Sleep with IO wake-up	0.00012	Waiting on IO event
Sleep with IO and 32-kHz RC oscillator timer wake-up	0.00064	
32-kHz Crystal Oscillator	0.0014	As an alternative sleep timer
The following current figures should be added to those above if the feature is being used		
RAM Retention	0.0009	
Comparator (low power mode)	0.0008	Reduced response time
Deep Sleep Mode		
Deep Sleep	0.00010	Waiting on chip RESET or IO event

Table 1: JN516x Current Consumption Summary

JN516x Wake-up Timings

This section contains information about the time taken by the JN516x to perform operations that are not related to the 802.15.4 protocol.

Parameter	Min	Typ	Max	Unit	Notes
Time for crystal to stabilise, ready to run CPU		0.74		ms	Reached oscillator amplitude threshold. Default bias current
Time for crystal to stabilise, ready for radio activity		1.0		ms	
Wake up from Deep Sleep or from Sleep		170		µs	Time to CPU release
Start-up time from reset (RESETN pin, BOR or SVM)		180		µs	Time to CPU release
Wake up from CPU Doze mode		0.2		µs	

Battery Capacity Calculation – Example for JN516x

The following example illustrates how to calculate the battery capacity required by a device in a non-beacon enabled star network that continuously performs the following actions:

1. Wakes from sleep
2. Reads data from a sensor connected to the two-wire serial interface
3. Performs a Clear Channel Assessment (CCA)
4. Transmits a data frame containing a payload of 64 bytes
5. Sleeps (without holding the memory contents) for 10 minutes before repeating Step 1

The durations and electrical currents corresponding to the above phases of device operation are determined below.

Waking from Sleep

If the contents of RAM have not been held, the time taken to wake from sleep mode can be calculated by adding the oscillator start-time to the application bootloader time (the bootloader initialises the .bss and .data RAM segments). Note that the JN516x devices start up using a fast RC oscillator and begin executing bootloader code at 24 MHz. After a further (740-170) µs, the faster 32-MHz crystal has stabilised and a glitch-less switchover occurs. The application code should wait until the crystal is stable for radio transmission, which occurs a further 430 µs later
(i.e. 1 ms from reset/wake event)

Time to wake from sleep = (1 – 0.170) = **0.830ms**

While the bootloader is initialising the C-runtime environment, the wireless transceiver will not be operating and therefore the current drawn is **4.98 mA** (CPU clock rate of 16 MHz).

Reading Data from Sensors

It is assumed that it takes **1 ms** to read data from the sensor attached to the two-wire serial interface. During this time, the current drawn by the JN516x is **5.04 mA**. No account is taken of the current drawn by the sensor.

Performing CCA

Before a data frame can be transmitted, the CSMA/CA algorithm is used to check that the channel is not being used. The time taken to execute the CSMA/CA algorithm in a non-beacon enabled network (assuming that the channel is found to be clear after the CCA and that the random back-off period is 2) can be calculated as follows:

BackoffPeriod (ms) = $(2^2 - 1) \times 20 / 62.5$ = **0.96 ms**

CCA Period (ms) = **0.128 ms**

During the back-off period, the application is running and the transceiver is on although it is not transmitting and receiving. The current drawn during this period is **12.3 mA**. During a CCA, the radio receiver is on and therefore the current drawn is **17.5 mA**.

Transmitting Data

Assuming that no source and destination data is included, the time taken to transmit a data frame can be calculated as follows:

Data Frame Transmission Period (ms) = (HeaderSize + PayloadSize) x 8 / 250

Data Frame Transmission Period (ms) = (15 + 64) x 8 / 250 = **2.528 ms**

The current drawn during this period is **15 mA**, the radio transmitting current.

While Sleeping

The current drawn during the (approximately) **10-minute** sleep period is **0.64 µA** (Sleep mode with I/O and RC oscillator timer wake-up)

Using the above time and current data, it is possible to calculate that the average current required by the application is approximately **0.740 µA** (see below for detailed calculation).

The device should therefore be capable of operating for over 32 years when powered by a 210-mAh button-cell.

The average current drawn by this application can be calculated by adding the electrical charge consumed during each phase and dividing by the total cycle time (10 minutes).

Phase	Current (mA) x Time (ms)	Charge (µC)
Wake from sleep	4.98mA x 0.830ms	4.1
Read sensors	5.04mA x 1ms	5.04
CCA back-off	12.3mA x 0.96ms	11.81
CCA	17mA x 0.128ms	2.17
Transmit data	15.3mA x 2.528ms	38.68
Sleep	0.00064mA x 600000ms	384
	Total	445.8

Therefore, 445.8 µC of charge is consumed over a total cycle time of 600 s (10 minutes):

Average Current = Total Charge Consumed / Time = 445.8 / 600 = **0.743 µA**

Note: This figure does not take into account the current drawn by the sensor or the self-discharge of the battery.

If the calculations are repeated for a range of sleep durations, the following graph of lifetime versus sleep duration is produced.

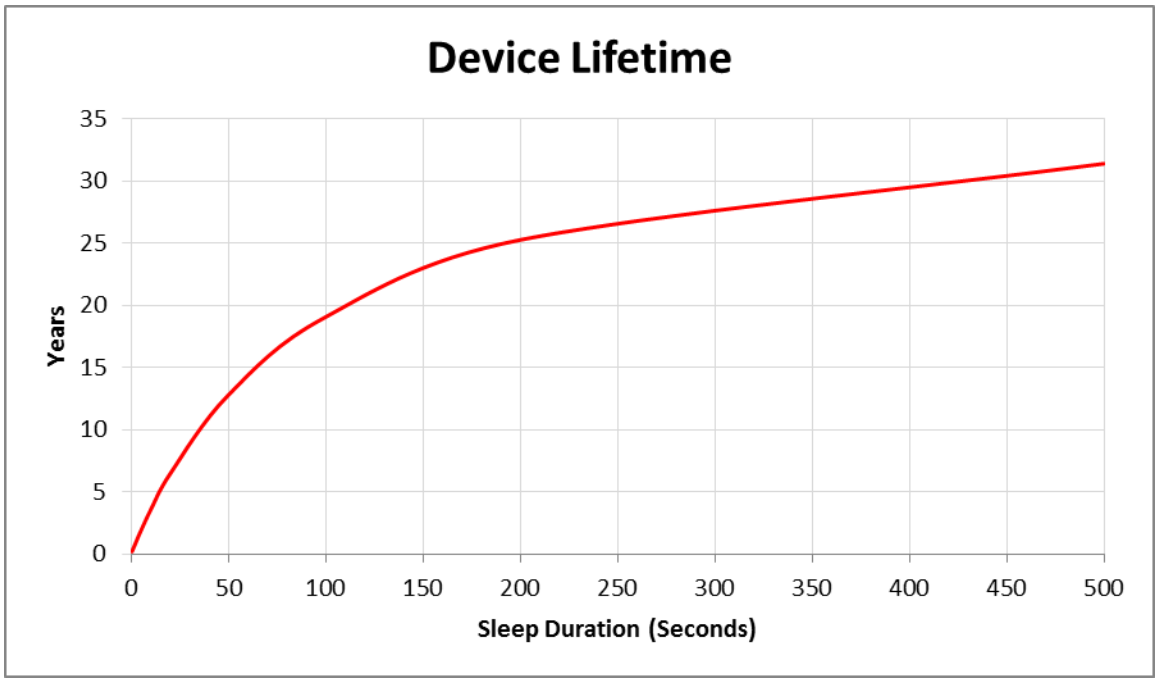

Figure 1: Lifetime vs Sleep Duration

Revision History

Version	Notes
1.0	Initial Release
1.1	Updated document template. Added section on beacon frame timings. Updated current consumption table.
1.2	Updated with latest current consumption figures and corrected average current figure in example. Added detailed calculation of average current figure in example.
1.3	Minor corrections
1.4	Updated document to remove references to JN5121 and replace with JN5139 and JN5148
1.5	Updated for JN516x devices and removed JN5139/48 information

Important Notice

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors 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. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

All trademarks are the property of their respective owners.

NXP Laboratories UK Ltd
(Formerly Jennic Ltd)
Furnival Street
Sheffield
S1 4QT
United Kingdom

Tel: +44 (0)114 281 2655
Fax: +44 (0)114 281 2951

www.nxp.com

