Power Management Using the Embedded Stand-alone Wake-up Mode Protocol. Rev. 2

AT86RF211 (TRX01) FSK Transceiver for ISM Radio Applications

Application Note

Why an Update of this Application Note?

The embedded wake-up mode of the AT86RF211 offers a lot of possibilities to users. The aim of this Application Note Revision 2 is to provide with a simple and effective way of implementing the wake-up mode, to save time and money.

All the wake-up functions are based on the Header recognition and analysis: the 10 bits sequence (1010100001) is searched before any wake-up, and the data rate "extracted" thanks to an embedded algorithm. In order for it to operate properly, the duration of the "0" and "1" must be close one to the other within a given tolerance.

Meeting this "duty cycle" requirement over all operating conditions is very easy at the transmitter side because any microcontroller is able to generate an output frame with an excellent precision (Timer Output Compare for instance). At the receiver side, it is made easier by using:

- a data rate $\leq$ 10 kbps: the longer the bits, the easier. Since it is possible to wake-up a device at 10 kbps and to transfer the data at 64 kbps afterwards, and thanks to the wake-up time based on RSSI checking, the additional power consumption is totally negligible.
- the "external" mode for the data slicer (the demodulated signal is compared to its average value stored into an external capacitor). The duty cycle remains excellent even if the external conditions have changed between two wake-up timeslots with no need for data slicer level adjustment. Consequences: lower requirements on the rest of the hardware (cost reduction) and shorter development time (simpler software).

Before using another implementation make sure to contact Atmel's FAE before.

Rev. 2186A–WIRE–08/02

Principle of the Embedded Wake-up Procedure

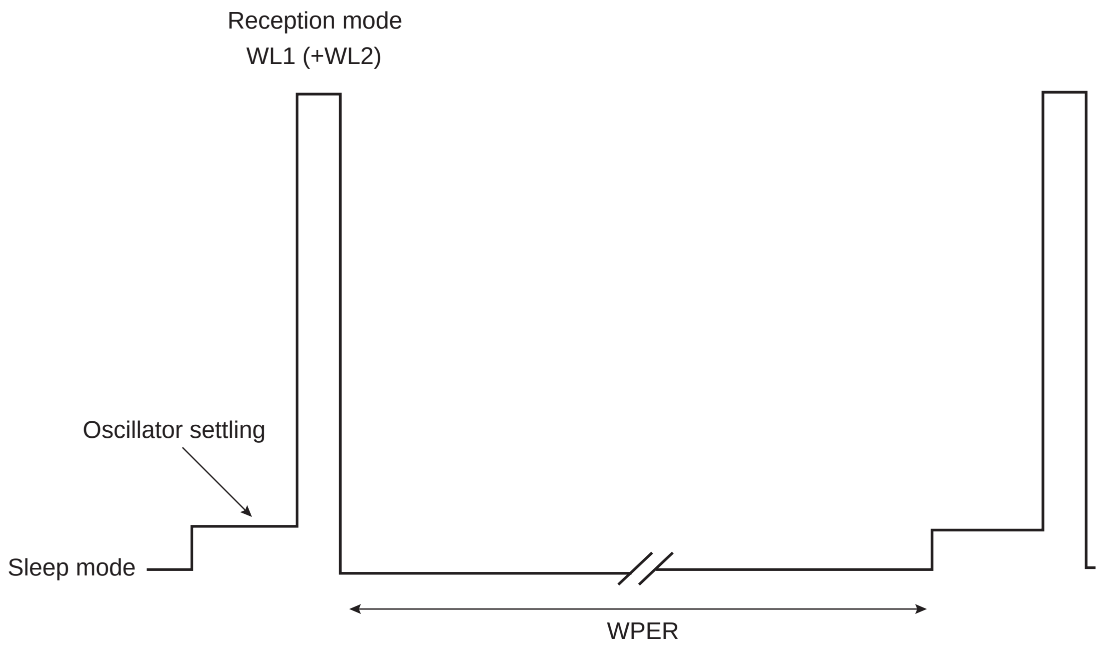
The "peak" consumption in Receive (Rx) or Transmit (Tx) mode of any high-performances transceiver makes it necessary to implement a power management protocol for long term battery operated systems.

The principle is generally as follow: the system is made of battery operated slaves (with highly critical power requirements) and a master. Slaves are in a very low power mode most of the time (i.e. 99.9%, the "sleep period"), wake-up periodically and wait for a dedicated message sent by the master. When such a message has been received, they make a relevant decision.

The two conditions to achieve a very low average power consumption are:

- very low sleep mode current (a few µA),
- reduction of reception window as much as possible (a few ms).

Usually, the microcontroller is in charge of managing such a protocol. The AT86RF211 features a *stand-alone wake-up function*. Once the chip has been set-up in this mode, it will periodically go into reception mode and try to catch an expected message (*stand alone operation*).



Sleep Mode = Very Low Power Mode

The sleep mode current of the AT86RF211 is typ. 3 µA. Because this is a stand-alone mode, the microcontroller is always in power-down mode (even during reception windows) with no timer running but only the possibility to be woken-up by an external interrupt given by the AT86RF211: this happens only when a valid message has been received.

Sleep Mode = Minimization of Reception Window

The reception window can be reduced to a minimum thanks to:

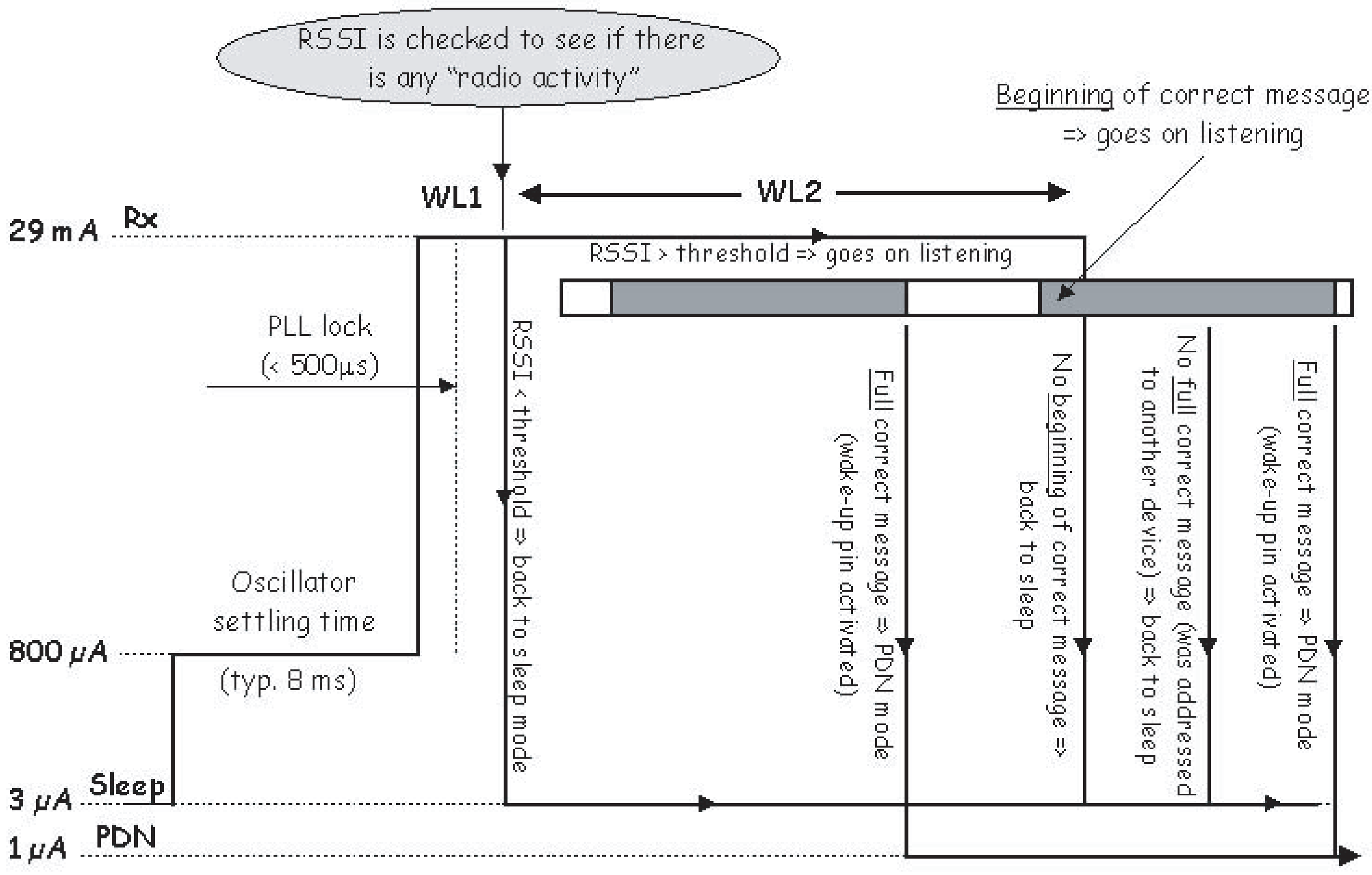
- its division into 2 time slots called WL1 & WL2: at the end of WL1, RSSI is checked and scan *goes on during WL2 only if there is a "radio activity", that is to say if RSSI > predefined threshold*. Choosing a very short value for WL1 will lead to a Σ I Δt as small as possible when the master is not sending anything (most of the time for most of the applications!).
- a built-in recognition of the received data
- a built-in check of RSSI

If no correct message has been received, the AT86RF211 is going back to sleep mode and try again after WPER. If a correct message is caught, then its Data field (if any) is stored into a register and the WAKEUP pin is activated to drive the microcontroller external interrupt (active state of this pin can be chosen). WAKEUP bit of STATUS register is also set.

Typical Sequence

As indicated before, Atmel suggests that the wake-up mode should be implemented with a very short WL1 (2 ms to make sure that PLL is locked & RSSI conversion completed). This way:

- the receiver will be in Rx mode during 2 ms only if no significant "radio activity" is detected, will go on listening only if it is worth doing so.
- a message is taken into account only if its level is > RSSI threshold (if WL1 is longer and a full correct message is received before its end, then it will be taken into account whatever the RSSI level may be).


RSSI is checked to see if there is any "radio activity"
Beginning of correct message => goes on listening
WL1
WL2
29 mA Rx
RSSI > threshold => goes on listening
PLL lock (< 500µs)
RSSI < threshold => back to sleep mode
Full correct message => PDN mode (wake-up pin activated)
No beginning of correct message => back to sleep
No full correct message (was addressed to another device) => back to sleep
Full correct message => PDN mode (wake-up pin activated)
Oscillator settling time (typ. 8 ms)
800 µA
3 µA Sleep
1 µA PDN

Notes:
1. 8 ms is a typical value that depends on the crystal's specifications (the user does not have to care about that: oscillator settling is automatically detected by AT86RF211).
2. Full correct message means: Header correct (mandatory) + Address matches (if any).
3. WL2 must be > 2 expected message duration (to make sure to catch it).
4. When a correct message has been received, the scan is stopped until the Data register WUD is read (it can be a simple access if the Data field is not used).
5. At the other side, the device that sends the wake-up message must do it during at least: [(1 ms + WPER + 8 ms + WL1) + 20%] to make sure that it will occur at least one time during the reception window (all time parameters have a tolerance of ± 20%).
6. The time parameters can be measured on the production bench and "calibrated" by software (the ±20% dispersion is reduced and all the devices have the same timeslots).

Practical Implementation of the Wake-up Mode

How is the Incoming Data Rate Determined?

The 10 bits Header '1010100001' is a recognition word (that is required to start any wake-up procedure), also used to extract the data rate value thanks to an embedded algorithm. In order for this algorithm to operate properly, the duration of the "0" and "1" must be close one to the other ("duty cycle") within a given tolerance:

- Between 1 kbps and 5 kbps: duty cycle $\leq \pm$ 3%.
- Between 5 kbps and 10 kbps: duty cycle $\leq \pm$ 2%.
- Between 10 kbps and 20 kbps: duty cycle $\leq \pm$ 1%.

Notes:
1. There is no requirement for the data rate accuracy itself (only the "duty cycle" must be within the tolerance).
2. This requirement only regards the Header (NOT the Address & Data fields)
3. This tolerance has NO relationship with the RATETOL value written into the WUR register, that is used for the rate check feature (see below)
4. This Header is set internally and can not be modified by the user

Practical Implementation

The following parameters have an influence on the duty cycle:

- Generation of the sequence by the Tx microcontroller
- Modulation of the RF by the Tx RF device.
- Demodulation of the received RF signal by the Rx RF device.
- Re-shape (data slicing) of the demodulated signal.

The 2nd and 3rd points depend on the RF component (internal architecture, loop filter of the PLL). Choosing the loop filter implementation given in chapter 2 ensures that the duty cycle will NOT be affected by the modulation/demodulation.

The 1st and 4th points depend on the application and have to meet the rules explained hereafter.

- **The following implementation is simple, and is a very good trade off simplicity/cost/performances. Atmel highly recommends to follow these rules.**
- **Should one wants to use another implementation, make sure to contact Atmel's FAE before.**

Data Rates

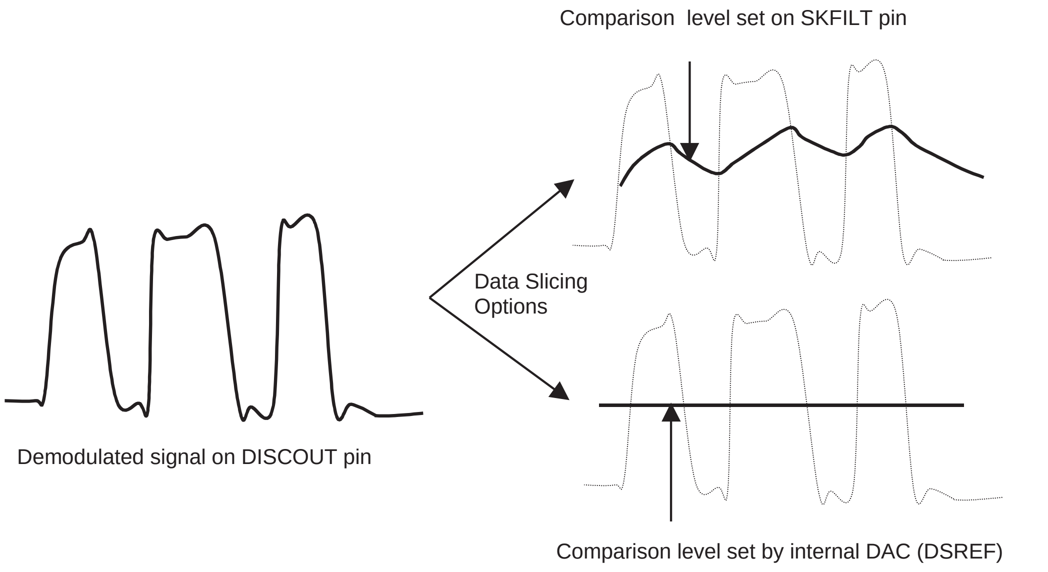
Atmel recommends:

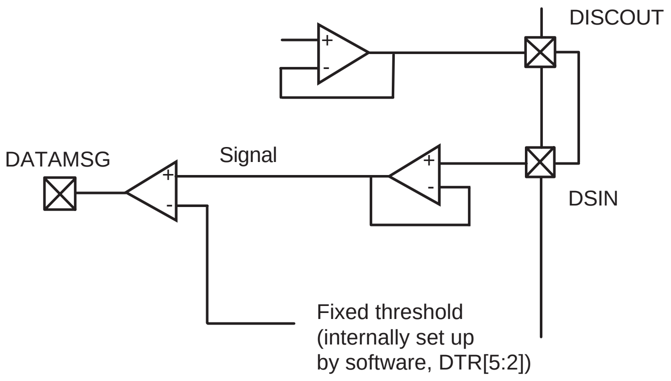
- a data rate $\leq$ 10 kbits/s
- a generation of the sequence by the Tx microcontroller $< \pm$ 1% (very easy to achieve with a microcontroller)

Note: It is possible to use a higher data rate, but it will harden the system requirements and bring no real advantage. Since it is possible to wake-up a device at 10 kbps and to transfer data at 64 kbps afterwards (see above), and thanks to the wake-up time based on RSSI checking, the additional power consumption is negligible. Please contact Atmel's FAE if application requires a higher data rate during the wake-up phase.

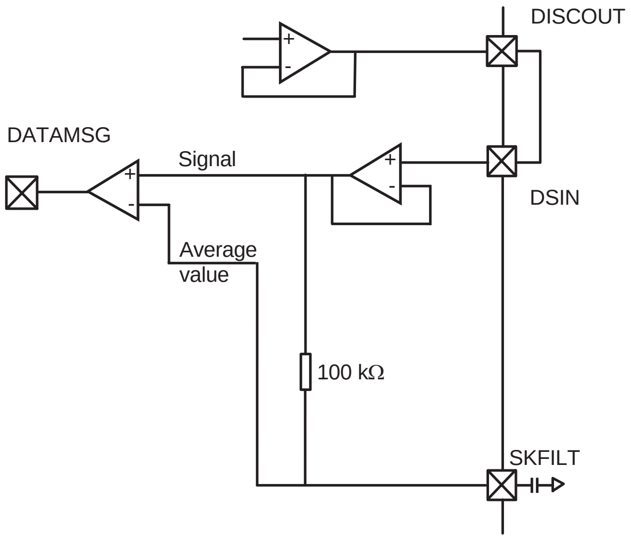
Mode for the Data Slicer

The data slicing is directly acting on the duty cyle of the received '0' and '1'. Two modes do exist: internal (comparison to a fixed threshold) and external (comparison to the average value of the signal).





Internal comparison mode: the signal is compared to a fixed level



External comparison mode: the signal is compared to its average value

Atmel recommends:

- **The "External" Mode for the Data Slicer:** since the demodulated signal is compared to its average value, stored into an external capacitor, the duty cycle remains naturally very close to 50% even if the external conditions (i.e. temperature) have changed between two wake-up timeslots, with no need for data slicer threshold re-adjustment.
- **A Value of SKFILT = 22 nF** to meet the duty cycle requirements from 1 to 10 kbits/s.

<u>About the Charge of the Capacitor</u>:

- the user must make sure that the SKFILT capacitor is charged before the recognition of the header starts (since the header recognition is based on re-shaped data). The maximum charging time is 15 ms (from 0V) with 22 nF.
- the SKFILT pin is set into a very high impedance state when the device is in the "sleep" phase. For SKFILT = 22 nF, the discharge is about 200 mV for WPER = 60 s, and 1V for WPER = 6 minutes. **For WPER < 1 min, there is no need to recharge the capacitor to be able to re-start the header recognition**. Above 1 minute, an additional relevant charging time (depending on WPER) must be taken into account within **<u>WL2</u>**.
- no need for the capacitor to be at the right value at the end of WL1: keeping WL1 as short as possible is important to keep the average power consumption as low as possible.
- WL1 = 2 ms and WL2 = 4 ms are the minimum values to make sure that the wake-up will operate properly.

Encoding Scheme for the Data/Address Fields

The duty cycle in the Address and the Data field can be as flexible as 30/70% or 70/30% (so meeting the header duty cycle requirement is OK for the rest of the message).

However, in order to keep the SKFILT capacitor charged at a correct level during the Address & Data fields (62 bits), one must not exceed a maximum number of equal consecutive bits, and "balance" the number of "0" and "1" as follow:

	Max. number of consecutive "0" or "1"	Any frame of this length must be "balanced" (= as many "0" as "1")
-2400 bits/s	5	10
-4800 bits/s	10	20
-9600 bits/s	20	40

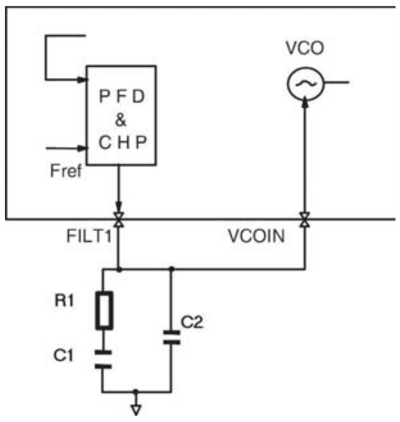
In any case, the discriminator of the AT86RF211 does NOT require a Manchester encoding.

Loop Filter

The loop filter of the PLL acts on the spectrum occupancy, the phase noise of the RF signal, the shape of the modulation, the rising and falling edges of the modulated signal of the transmitter (transitions between F0 and F1), the demodulated signal on the receiver (DISCOUT signal). In order to meet the duty cycle tolerance, the rising and falling edges of the demodulated signals must be sharp enough, especially when the data rate grows.

Atmel recommends using:

- [(C1 = 5.6 nF + R1 = 3.3 kΩ) // C2 = 560 pF] if the maximum data rate of the application (wake-up phase + data transfer afterwards) does not exceed 10 kbps.
- [(C1 = 1 nF + R1 = 10 kΩ) // C2 = 100 pF] if the data rate during the data transfer exceeds 10 kbits/s. This filter also works well during the wake-up phase (< 10 kbits/s) as the difference is mainly observed on the phase noise, not on the rising/falling edges of the signal.



Note: [(C1 = 2.2 nF + R1 = 14.7 kΩ) // C2 = 220 pF] can also be used for very narrow band applications.

Post Filter between DISCOUT and DSIN

The purpose of a post-filtering between DISCOUT and DSIN is to remove potential high-frequency noise on the demodulated signal. The filter also shapes the demodulated signal and smoothes the edges of the bits.

As the wake-up mode must be used with sharp transition levels, the cut-off frequency of the filter must not be too close to the equivalent transition frequency of the expected data rate. The rule-of-thumb, considering a 1st order filter, is: Freqcut-off (kHz) = Data rate (kbps). For instance, at 10 kbps, the post-filter cut-off frequency should not be lower than 10 kHz.

During the normal communication receive mode (after wake-up phase), it can be lower: Freqcut-off (kHz) = ½ Data rate (kbps), but not during the wake-up procedure.

Use of Different Data Rates After Wake-up (switch between 'external'/'internal' mode)

As indicated before, it is possible to run a higher data rate when the system has been woken up, if more data are to be exchanged between two modules. The AT86RF211 may be switched from the "external mode" to the "internal mode" by setting the CTRL[4] register of the AT86RF211 to '1', and setting up the DTR register accordingly (see DataSlicer application note for more information).

Use of RSSI Threshold and Hysteresis

At the end of WL1, RSSI threshold and hysteresis are used the same way as with the normal communication receive mode:

- defined in CTRL1[23-15].
- threshold levels = TRSSI-HRSSI and TRSSI+HRSSI.

➔ refer to the datasheet of the AT86RF211 for more information.

Clock Recovery of the Wake-up

This clock recovery process must not be confused with the clock recovery of the normal communication receive mode, giving an output clock on DATACLK pin. The wake-up clock recovery gives no output signal. It has only the purpose to trigger the received Address and Data fields. The data rate of the received Header is automatically extracted from a correct Header:

- if RATECHK (WUR register) is set to '1', the extracted data rate is compared to RATE (of WUR register) with a tolerance of ± RATETOL (WUR register). If the checking process succeeds, the message is considered valid. The advised tolerance for the data rate must be lower than 25% of the data rate (typically 10%).
- if RATECHK is set to '1' and the extracted data rate is different from the programmed one with the relevant tolerance, the detected Header is ignored and the chip searches for another message until the end of WL2 period. This option allows the designer to wake-up different devices with only the Header: no Address and no Data, but different data rates. It also helps to remove Header sequence which may occur randomly during scan period.

The RATECHK functionality MUST BE USED with the wake-up procedure. It is mandatory to ensure no detection of compatible random sequence in noise when no signal is present. The RSSI level allows to remove the effect and the RATECHK helps to get the best concerning sensitivity.

Error Detection

Error detection is implemented when MSGTST (of WUC register) is set to '1'. This is useful when debugging wake-up protocol.

After correct header readout, the reception of message is interrupted if the received Address does not fit or if an error is detected (in order to be analyzed):

- wrong Address value.
- wrong rate in RATECHK mode.
- no STOP field in variable data length mode.
- stuff error if four consecutive equal bits in the field (Address + Data) are detected.

Note: there is no direct access to the received Address in the chip. To help debug, set ADDL (of WUA) to 0 and DATL (of WUC) to Address length: the Address is then caught as a Data!

Message Format

Expected Message's Format

A wake-up message is made of 4 fields (only one is mandatory). They are defined by writing into the 4 dedicated registers. Please carefully read the notes and warnings which follow this table.

FIELD	HEADER	ADDRESS	DATA	STOP
Length	10 bits(1)	0 to 20 bits(2)	0 to 32 bits(3)	4 bits(4)
Value	1010100001b			1111b after 0 0000b after 1
Set-up		ADD of WUC (validation) ADDL of WUA (length) ADD of WUA (device's address)	DATA of WUC (validation) DATL of WUC (length)	STOP of WUC (validation)
Processing of received data	Header must be correct to go further	(if this field is active): - compared to ADD of WUA	(if this field is active): - received data is stored in WUD - extracted data length is stored into MSGDATL of STAT (variable length mode only)	
Usage	Mandatory	Option (active if ADD = 1)	Option (active if DATA = 1) The mode with no data can simply be used to wake-up the chip	Option (active if ADD = 1)
Make sure to meet chapter “Practical Implementation” on page 5 requirements for data encoding for all these fields!				

Notes:

1. One message is made with only one Header. Do NOT insert any additional header because the first recognition will inhibit detection mechanism until a message aborts or ends: the second header would be regarded as an address and the message lost. The whole message must be sent each time.
2. The Header is a predefined sequence of 10 bits that CANNOT be changed and acts like a synchronization. The data rate is measured during the Header and is used to sample the rest of the message. So the duration of all '0' or '1' bits must be identical in the whole Header field with the duty cycle requirement given in chapter “Practical Implementation of the Wake-up Mode” on page 5.

   Warning: the length of the last bit of the header (10th) must be reduced down to 50% of the other bits duration, in order to guarantee a good trigger of the data/address fields bits.
   For instance, if the expected data rate is 4800 bps, the bits duration is 0.208 ms for all the Header field but the last one which must be about 0.104 ms (no accuracy requirement concerning this bit).

   Warning: an extra bit '0' (required by the hardware) MUST be added at the end of the device's address (LSB position).
   Example: 3 devices in a Wake-up mode, address must be coded with 2 bits + 1 extra bit '0'. ADDL (length of address field) must correspond to 3 bits: '00010' (and not '00001').

   ➔ This extra bit must also be added in the wake-up message that will be sent, which means that devices will have the following address:
   ADD1 = '01' + '0'
   ADD2 = '10' + '0'
   ADD3 = '11' + '0'

   ➔ so **19 useful bits are available for the device's address**, the last is useless but needed (the 20 bits duration takes this extra bit "0" into account).
3. The DATA field can be used in two different ways:
   - fixed data length: data length is set by DATL of WUC register from 1 to 32 bits and STOP field is not used.
   - variable data length: data length is variable from 1 to 32 bits. STOP field must be used. Extracted data length (= data received in wake-up message before STOP field) is available into MSGDATL of STAT register. In this mode DATL of WUC register must be set to 32 bits.

The received Data (if any) is stored into WUD register and available for the microcontroller. The read of this data deactivates the wake-up pin (as well as WAKEUP flag of STATUS register).

Warning: the first data bit received (LSB bit: WUD[0] of WUD register) is the last bit appearing on the serial interface SDATA pin when WUD is read.

4. The STOP field is a predefined sequence, mandatory if variable data length mode is used. The sender must add this sequence after the last data bit:
 - '0000' if the last bit of the message is '1'
 - '1111' if the last bit of the message is '0'

 This prevents from sending 0000 or 1111 in the message itself. For this reason, a stuffing mechanism has been implemented.

Stuff Mechanism

In variable length mode, a stuff mechanism is used on the whole {Address + Data} field in order to make it impossible to have a stop sequence ('0000' or '1111') into the wanted message.

If three consecutive bits are equal, an extra opposite stuffing bit has to be inserted in the transmitted message by the transmitter and is automatically removed by the receiver in wake-up mode.

The stuff mechanism starts at the first bit after Header until the last bit of data. This mechanism is NOT used for Header and Stop fields.

Example:

{Address + Data} field	Emitted {Address + Data} field including stuff
00110101	00110101
01000101	01000**1**101
10111010	10111**0**010
01000010	01000**1**010
10111101	10111**0**101
0001010	000**1**1010
1110010	111**0**00**1**10
110101	110101

Note: Even if the variable data length mode is used, the stuff mechanism can be inhibited (using ISTU of WUC). The user must make sure that no stop sequence is contained into Address and Data field of the wake-up message.

Set-Up of a Transceiver and Initialization of a "Sleep" Procedure

Relevant Registers

All wake-up parameters are set-up as wanted at the beginning of the application by writing into the relevant registers. Following registers are used for set-up:

- Wake-up Control Register (WUC): definition of WPER, WL1, WL2, Address/Data fields properties + others
- Wake-up Data Rate Register (WUR): definition of wake-up message expected data rate + others
- Wake-up Address Register (WUA): definition of device's address

When a correct message is received, Data field (if used) is stored into Wake-up Data Register (WUD), and the STATUS register contents the relevant flags, length of received Data field (if variable) & measured data rate.

These registers are fully described in the datasheet.

Note: WPER is from 10 ms to 328s, WL1 from 1 ms to 1024 ms, WL2 from 0 to 31xWL1. These 3 parameters are process (and piece to piece) dependent (±20%). The time parameters can be measured on the production bench and "calibrated" by software (the ±20% dispersion is reduced and all the devices have the same timeslots).

The registers must be filled properly by the microcontroller.

Procedure to Put a Transceiver to Sleep

Once the AT86RF211 is set-up as wanted (examples will be given at chapter “How to Wake-up (a) Distant Device(s)?” on page 13), the microcontroller should:

- set-up the transceiver in PDN mode since sleep mode is a sub-mode of PDN mode, so CTRL1[31] = '0'.
- set WUEN = CTRL1[26] = '1'
- set WUE = WUC[31] = '1'
  - at this moment, the wake-up cycle begins
- go to sleep, waiting for an external interrupt: WAKEUP pin when valid message received by AT86RF211.

Starting a wake-up cycle will:

- deactivate WAKEUP pin and WAKEUP of STAT register,
- reset the message error flag MSGERR of STAT register in MSGTST,
- reset the extracted data rate MSGMRATE of STAT register,
- reset the wake-up data register WUD,
- reset the wake-up data message length MSGDATL of STAT register.

How to Wake-up (a) Distant Device(s)?

General Rules

As indicated in chapter 1-2:

- WL2 must be > 2 x message duration, to make sure to catch it
- wake-up message duration must be > (1 ms + WPER + 8 ms + WL1) + 20%
- when a correct message has been received, even with no Data field, the application MUST read at least one bit in the DATA register to acknowledge the wake-up (pin + STATUS bit) and enable other wake-up sequences.

Typical Applications

Individual Addressing

Each device has got its own address (the Address field must be long enough to allow that).

Example:

- a 4-device network: a 2-bit Address field is required (+ the extra bit '0').
- No Data needed
- Stuff mechanism inhibited
- WAKEUP pin active low
- Data rate = 10 kbits/s, i.e. 100 µs per bit, with tolerance = 10% for the Address field.
- Rate Check filtering is activated
- Wake-up test every second
- RSSI test after 2 ms, total test duration = 6 ms (WL1 = 2 ms + WL2 = 4 ms)
- Message duration = 1.3 ms (= (10+2+1) bits x 100 µs). WL2 is chosen to be twice the message length to make sure to catch it.

Values of registers (to set-up and put transceiver into sleep):

- CTRL1: WUEN = '1', PDN = '0'(so bit 26 = '1' and bit 31 = '0')

- WUR: WUOP = '00', RATECHK = '1', RATE = '0000111111', RATETOL = '00110' so **WUR = '00 1000 0111 1110 0110'**
RATETOL is set to 6 x 1.56 µs

- WUA: ADDL = '00010', ADD = '00000000000000000**xx**0' (2 bits in bold depending on the device: 00,01,10,11), so for the 4 devices:

*WUA = '0 0010 0000 0000 0000 0000 0**00**0'*

*WUA = '0 0010 0000 0000 0000 0000 0**01**0'*

*WUA = '0 0010 0000 0000 0000 0000 0**10**0'*

*WUA = '0 0010 0000 0000 0000 0000 0**11**0'*

Only the 3 last bits are significant for ADD. The 17 MSB bits are 'don't care', but the 20 bits of ADD **must be written**.

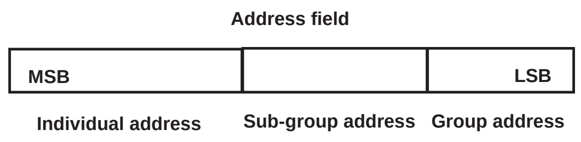
- WUC: WUE = '1', DATA = '0', STOP = '0', DATL = '00000', ADD = '1', MSGTST = '0', WPER = '0 0110 0011', WL1 = '000 0001', WL2 = '010', ISTU = '1' so **WUC = '1000 0000 1000 1100 0110 0000 0101 0100'**

The message to wake-up the device at address '01' is then: **'10 1010 0001'(Header) + '010'(Address)**

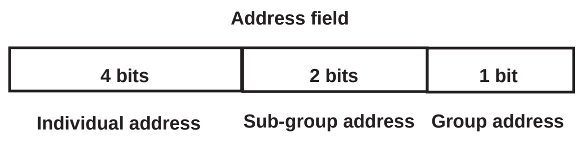
Group Addressing (no Data Field)

Each device has got its own address, but they are structured to make it possible to wake-up a whole group or sub-groups in one shot.

<u>Principle</u>: the address is structured as follow (common part = LSB & individual part = MSB), and the selection of devices to be woken up is made by choosing the length of significant Address field (ADDL value of register WUA).



For instance, to wake-up a whole group, ADDL (of WUA) will be chosen equal to group address length (so the other bits will be ignored). Once a group is woken-up, each device is reprogrammed to be again in wake-up mode but with individual addressing.



<u>Example</u>: same configuration as before except:

message duration = 1.8 ms [(10+7+1) bits x 100 µs], so WL2 = 4 ms remains correct.

- a 15-device network, divided into 2 groups and 4 sub-groups, 7 devices in group 1 (3 in sub-group 1 and 4 in sub-group 2), 8 devices in group 2 (4 in sub-group 3 and 4 in sub-group 4). The Address can be coded with 6 bits (+ 1 extra bit '0'):

- CTRL1: WUEN = '1', PDN = '0'(so bit 26 = '1' and bit 31 = '0')

- WUR register is unchanged as the data rate mode has not changed so WUR = **'00 1000 0111 1110 0110'**

- WUC register is unchanged, so WUC = **'1000 0000 1000 1100 0110 0000 0101 0100'**

- WUA: ADD = '0000 0000 0000 0xxx xxx0' (6 bits in bold depending on the device), ADDL = yyyyy depends on the target devices to be reached: '0 0110' (individual), '0 0011' (whole sub-group), '0 0010' (whole group)

So for all the 15 devices:

WUA = 'y yyyy 0000 0000 0000 0**001 001**0' number 1/group 1/sub-group 1

WUA = 'y yyyy 0000 0000 0000 0**010 001**0' number 2

WUA = 'y yyyy 0000 0000 0000 0**011 001**0' number 3

WUA = 'y yyyy 0000 0000 0000 0**001 101**0' number 1/group 1 /sub-group 2

WUA = 'y yyyy 0000 0000 0000 0**010 101**0' number 2

WUA = 'y yyyy 0000 0000 0000 0**011 101**0' number 3

WUA = 'y yyyy 0000 0000 0000 0**100 101**0' number 4

WUA = 'y yyyy 0000 0000 0000 0**001 010**0' number 1/group 2 /sub-group 3

WUA = 'y yyyy 0000 0000 0000 0**010 010**0' number 2

WUA = 'y yyyy 0000 0000 0000 0**011 010**0' number 3

WUA = 'y yyyy 0000 0000 0000 0**100 010**0' number 4

WUA = 'y yyyy 0000 0000 0000 0**001 110**0' number 1/group 2 /sub-group 4

WUA = 'y yyyy 0000 0000 0000 0**010 110**0' number 2

WUA = 'y yyyy 0000 0000 0000 0**011 110**0' number 3

WUA = 'y yyyy 0000 0000 0000 0**100 110**0' number 4

To carry out a group wake-up, ADDL = '0 0010' and the wake-up message will be:

for group 1: '**10 1010 0001**'(Header) + '**010**'(Address)

for group 2: '**10 1010 0001**'(Header) + '**100**'(Address)

To carry out an individual wake-up, ADDL = '0 0110' and the wake-up message will be:

for number 1/group 1/subgroup 1: '**10 1010 0001**'(Header) + '**001 0010**'(Address)

...

Group Addressing (with Data Field)

Example: same configuration as before except:

- 10 bits of data
  - ➔ maximum length of wake-up message = 10 + 7 (Address) + 10 (Data).

The wake-up message lasts (10 + 7 + 10) bits x 100 µs = 2.7 ms, so WL2 = 6 ms is correct.

WL1 = 2 ms and WL2 = 3 x WL1 = 6 ms.

- CTRL1: WUEN = '1', PDN = '0'(so bit 26 = '1' and bit 31 = '0')

- WUC register: WUE = '1', DATA = '1', STOP = '0', DATAL = '01001', ADD = '1', MSGTST = '0', WPER = '001100011', WL1 = '0000001', WL2 = '011'.

To carry out an individual wake-up, then ADDL = '0 0110' (in WUA) and the wake-up message will be:
for number 1/group 1/subgroup 1 with 0111000101 as data to be sent.
'**1010100001**'(Header) + '**001 0010**'(Address) + '**0111 00 0101**'(Data)

Notes:

- WL2 set to 6 ms does not impact the budget of the energy consumption as this period is active only if the RSSI is above the pre-defined value.
- The Address and Data fields of the message respect the minimum numbers of following bits set to the same value.

Addressing by "Data rate" (individual or group)

Another way to address the devices selectively is to use the bit rate. As indicated before, a clock is extracted from the header and (if relevant bits are selected) can be compared to a value that has been set-up (with a defined tolerance). Should the bit rate be different, the message is not recognized as a wake-up message. This way of addressing allows to minimize the length of the wake-up message since no address is needed.

Example:

- 5 receivers waiting for a wake-up message

- no Address needed

- no Data needed

- stuff mechanism inhibited.

- WAKEUP pin active low

- Rate check used for addressing with about 2% for the tolerance.

- Wake-up test every second

Choice of datarates:

- data rate 1 = 8 kbits/s, i.e. 125 µs per bit

- data rate 2 = 9 kbits/s, i.e. 111 µs per bit

- data rate 3 = 10 kbits/s, i.e. 100 µs per bit

- data rate 4 = 11 kbits/s, i.e. 90.9 µs per bit

- data rate 5 = 12 kbits/s, i.e. 83.3 µs per bit

Message duration (10 bits of Header) is between (10 x 83.3 µs = 833 µs) and (10 x 125 µs = 1.25 ms). In order to reduce Rx duration as much as possible, the configuration is set up with the minimum required WL1 and WL2. For all the devices, WL1 = 2 ms and WL2 = 4 ms, so total maximum test duration is 6 ms.

- CTRL1: WUEN = '1', PDN = '0', so bit 26 = '1' & bit 31 = '0'.

- WUR: WUOP = '00', RATECHK = '1', RATE = 'xx xxxx xxxx' depending on datarate, RATETOL = '00010'.

WUR = '00 1**000 1001 111**0 0010' for data rate = 8 kbits/s (corresponds to 80 x 1.56 µs)

WUR = '00 1**000 1000 110**0 0010' for data rate = 9 kbits/s (corresponds to 71 x 1.56 µs)

WUR = '00 1**000 0111 111**0 0010' for data rate = 10 kbits/s (corresponds to 64 x 1.56 µs)

WUR = '00 1**000 0111 001**0 0010' for data rate = 11 kbits/s (corresponds to 58 x 1.56 µs)

WUR = '00 1**000 0110 100**0 0010' for data rate = 12 kbits/s (corresponds to 53 x 1.56 µs)

RATETOL = 2 x 1.56 µs.

- WUC register: WUE = '1', DATA = '0', STOP = '0', DATAL = '00000', ADD = '0', MSGTST = '0', WPER = '001100011', WL1 = '0000001' for all data rates, WL2 '010', ISTU '1' so:

WUC = '1000 0000 0000 1100 0110 0000 0101 0100'.

- WUA is unused as no Address field is expected.

Calculation of Power consumption

Hypothesis & Method

The capacity of a battery is given in A.h that corresponds to an electric charge (Q = I . t). It is measured with given discharges conditions (discharge current up to a final voltage). Atmel assumes that in practical half of this value can be really used because of current constraints of the application (operating voltage > 2.4V, self discharge, T°...). Let's call Qb = Q/2.

During each wake-up cycle, a charge Qr is used:

- If no "radio activity": Qr = Σ I Δt < 8 ms x 0.8 mA + WL1 x 29 mA + [WPER] x 3 µA
- If "radio activity": Qr = Σ I Δt < 8 ms x 0.8 mA + WL1 x 29 mA + WL2 x 29 mA + [WPER] x 3 µA

When the device transmits data during WL3 (at 10 dBm), Qt is used:

Qt = WL3 x 48 mA

Application: Transceiver vs. Single Transmitter

Example:

- a wireless system must be able to answer maximum 5s after request, but is interrogated typically once an hour.
- the data to be sent = 16 bytes = [16 x 8] = 128 bits at 9600 bits/s => 13 ms
- 2 AA alkaline batteries (Q = 2.7 A.h) => Qb = 1.3 A.h = 1300 x 3600 x 1000 mA.ms = 4.7 10^9 mA.ms

- With the AT86RF211 transceiver.
  - the chip will wake-up periodically & send data only on request.
  - wake-up message (worst case with full Address and Data) = 62 bits => 6.5 ms

- – WL1 = 2 ms, WL2 = 16 ms, (1 ms + WPER + 8 ms +WL1) + 20% = 5 s, so the worst case to make sure that the device is ready to answer before 5s is: (1 ms + WPER + 8 ms + WL1) = 4166 ms => WPER = 4170 ms ('010101010').
- – as the WPER is short, there is no need to care about the discharge of the SKFILT capacitor: the level remains constant.
- – since "request & answer" occurs typically once an hour, the corresponding power consumption can be neglected.

Consumption is determined by sleep mode, oscillator settling & WL1.

During each 1610 ms cycle here are:

- – 8 ms: oscillator settling time, the consumption is 0.8 mA.
- – 0.5 ms for the PLL + WL1 = 2.5 ms in reception mode (rounded to 3 ms).
- – 3 µA during 4170 ms: sleep mode.

Qc = 8 x 0.8 + 3 x 29 + 4170 x 0.003 = 105.9 mA.ms

Number of 4181 ms cycles = 4.7 10^9/105.9 = 44 381 490 cycles of 4181 ms corresponding to 6 years ± 20% (± 1.2 year).

- With a single transmitter, the chip must send the data each time (so every 4.2 s)

  With the same parameters, during each 4.2 s cycle Atmel has:

  - – 8 ms: oscillator settling time, the consumption is 0.8 mA.
  - – 0.5 ms for the PLL transmission mode.
  - – 48 mA during 13 ms: transmission of the information.
  - – 3 µA during 4.2 s: sleep mode.

Qc = 8 x 0.8 + 13 x 48 + 4200 x 0.003 = 643 mA.ms

Number of 4.2s cycles = 4.7 10^9/643 = 7 309 489 cycles of 4.2s, corresponds to 12 months ± 1 month.

The conclusion is that the use of the transceiver with the feature of a stand-alone wake-up mode allows to save energy and improve batteries' life.

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 487-2600

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-76-58-30-00
FAX (33) 4-76-58-34-80

---

e-mail
literature@atmel.com

Web Site
http://www.atmel.com