Reducing downlink current drain of Sigfox devices by using discontinuous RX

Application note

Table of contents

1 Introduction

Energy saving is at the heart of wireless devices design. In most applications, radio transmission is the major contributor of electrical consumption.

Sigfox devices achieve very low consumption in uplink transmission, because no link establishement is required (device is not disciplined), and transceivers are more and more efficient regarding current drain. However, downlink transmission used to retrieve data from the network requires more power because the device has to open a receiving window, in which the downlink data is delivered.

This document describes the use of discontinuous RX technique to reduce current drain when receiving downlink data from Sigfox network.

2 Sigfox downlink mechanism

As a reminder, the following figure shows the Sigfox downlink principle. The device first sends an uplink message including a downlink request. The receiving windows starts $\mathbf{T_W}$ seconds after the end of the first uplink frame. Downlink data is delivered by Sigfox network within the $\mathbf{T_{RX}}$ seconds of the receiving window, when the data is available from the applicative server.

$\mathbf{T_W}$ and $\mathbf{T_{RX}}$ timings are designed so that the downlink mechanism is event-triggered. Sigfox backend has to recover data from the applicative server and select the base station that will transmit the downlink message (according to uplink reception parameters and base station TX duty cycle).

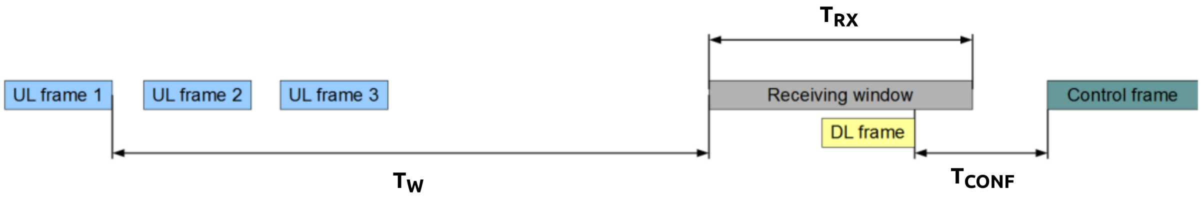

Figure 1 – *Downlink mechanism timing*

The next figure and table present the downlink frame structure which is sent by Sigfox network to the device.

See Sigfox connected objects radio specification for more details (reference **EP-SPECS V1.5**).

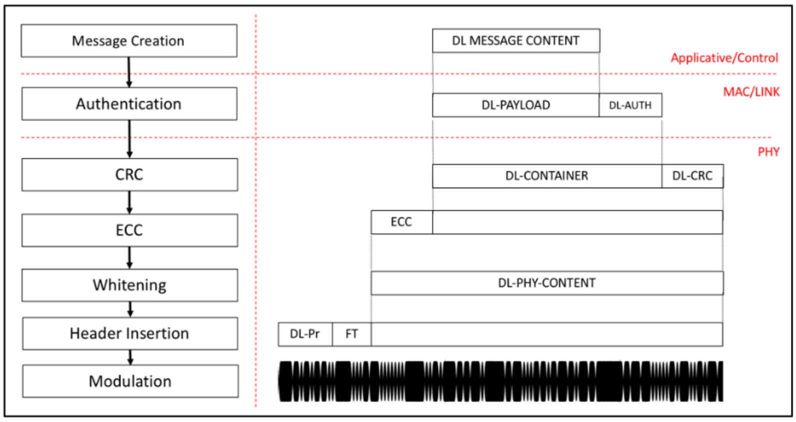

Figure 2 – *Downlink communication stack*

In practice implementation, **DL-Pr** and **DL-FT** are considered 88 and 16 bits respectively, so that the downlink frame can be structured as follow:

Field	Description	Length (bits)	Length (bytes)	Value
DL-Pr	Bit synchronization	88	11	0xAAAAAAAAAAAAAAAAAAAAAA
DL-FT	Frame type	16	2	0xB227
DL-ECC	Error correction code	32	4	Data dependent
DL-PAYLOAD	Downlink data	64	8	
DL-AUTH	Frame authentification	16	2	
DL-CRC	Cyclic redundary check	8	1	
TOTAL		**224**	**28**	

Figure 3 – *Downlink frame structure*

3 Downlink reception techniques

3.1 Continuous RX

After a downlink request, the device sleeps during $\mathbf{T_W}$, then wakes-up and configures the transceiver in continuous RX operation until the downlink data is received.

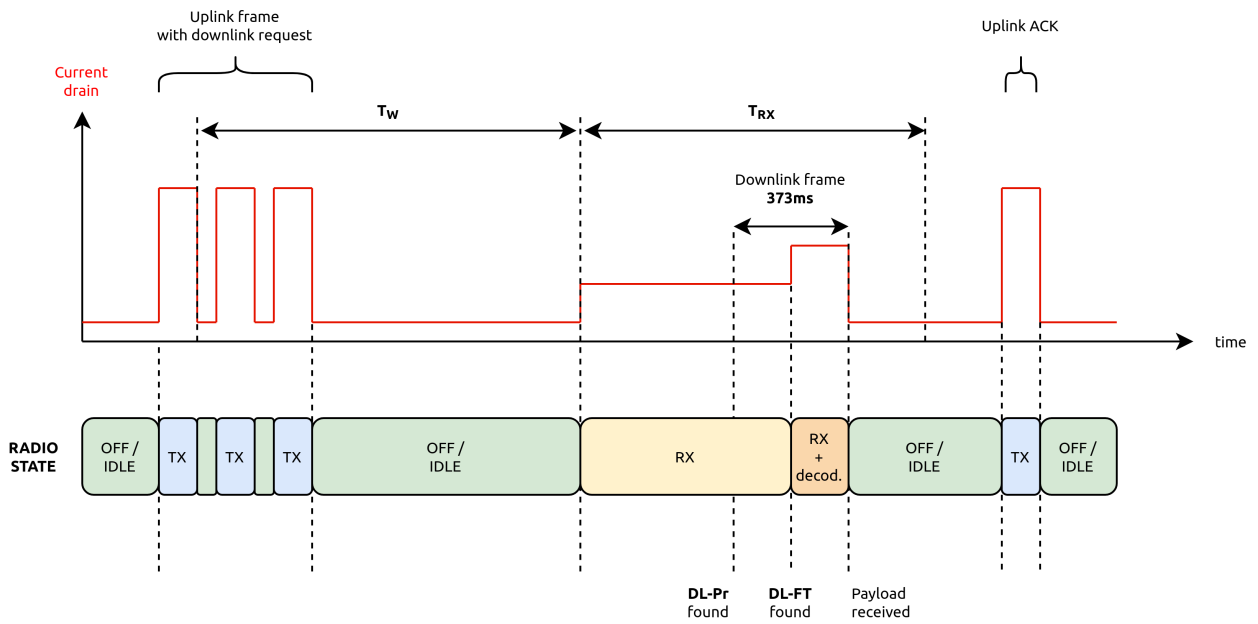

Figure 4 – *Continuous RX timing and consumption profile*

The downlink frame is sent at 600 bits/s and only lasts **373ms** within the $\mathbf{T_{RX}}$ window. Continuous RX implementation is thus sub-optimal in terms of energy when the device frequently asks for downlink data.

3.2 Discontinuous RX

To optimize current drain during the downlink window, a discontinuous RX operation can be performed by sampling the large downlink preamble **DL-Pr** (11 bytes at 600 bits/s last 146ms). Instead of continuously waiting for the frame signature, the device periodically detects for a part of the preamble with a duty-cycled reception.

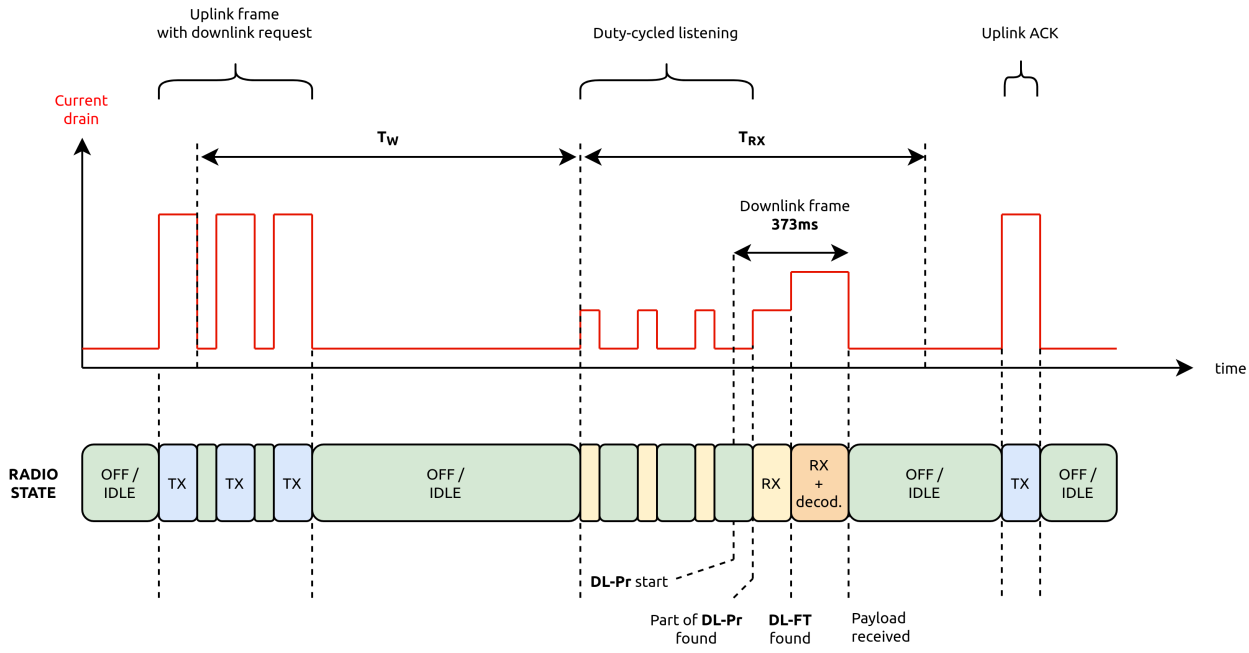

Figure 5 – *Discontinuous RX timing and consumption profile*

4 Discontinuous RX parameters

When discontinuous RX is used, the duty cycle should be set to ensure that the preamble is detected, regardless RX sub-windows start times and regardless the downlink frame arrival time.

4.1 Definitions

Signal preambles are used to align the receiver on symbol rate and frame start. To be easily identified, a preamble is composed of a **unit pattern** sent multiple times before the container starts. Discontinuous RX relies on the periodicity of the **whole preamble** and consists in periodically detecting a small part of it, that we designate the **sub-pattern**.

Parameter	Unit	Generic symbol
Unit pattern		*
Unit pattern length	bits	L_u
Whole pattern		***********
Whole pattern length	Number of units pattern(s)	L_w
Sub-pattern		***
Sub-pattern length	Number of units pattern(s)	L_s
Discontinuous RX cycle time (sub-windows periodicity)	Number of units pattern(s)	T_s

Figure 6 – *Discontinuous RX parameters*

Below is an illustrative example with $L_w = 11$, $L_s = 3$ and $T_s = 7$.

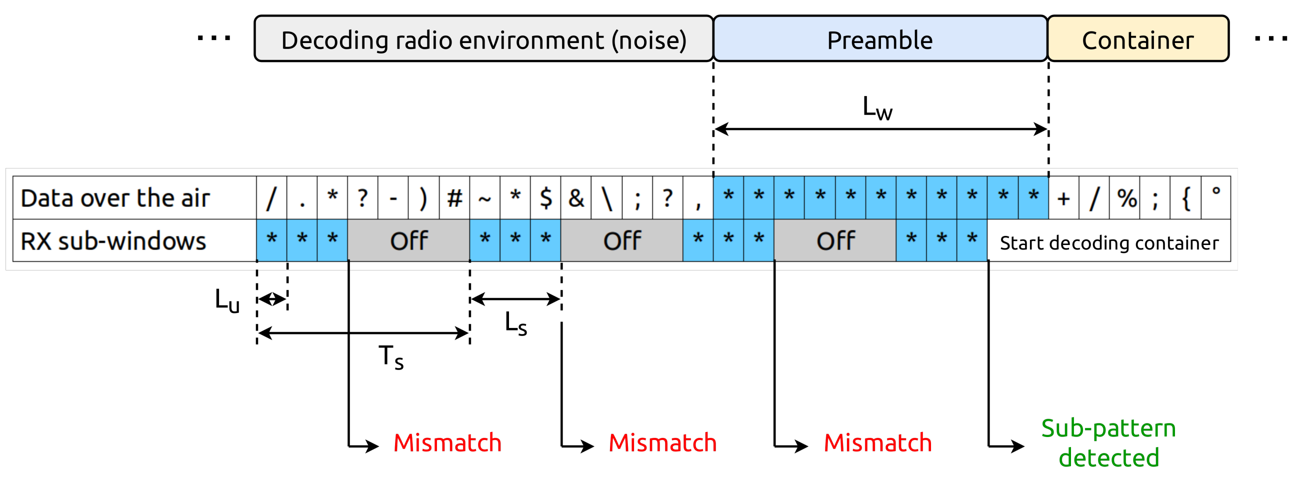

Figure 7 – *Discontinuous RX parameters example*

4.2 Cycle time sizing

The discontinuous RX cycle time is set to ensure that the sub-pattern is detected. The criterion relies on (T_s + L_s) , assuming T_s is always greater than L_s:

$$(T_s + L_s) \leq L_w$$

Then, T_s and L_s can be tuned according to the following criteria:

- **Minimizing the duty cycle (L_s / T_s)** to save as much energy as possible.
- **Reducing L_s** makes the system more sensitive to false detection.

4.3 Application to Sigfox downlink

Below is the patterns definition for the Sigfox downlink frame:

Parameter	Unit	Sigfox value
Unit pattern		0b10
Unit pattern length	bits	$L_u = 2$
Whole pattern		0xAAAAAAAAAAAAAAAAAAAAAA
Whole pattern length	Number of units pattern(s)	$L_w = 44$

Figure 8 – *Discontinuous RX parameters of Sigfox downlink frame*

A discontinuous RX can be performed with **T_s = 32** and **L_s = 8**. Those values comply to the previous criterion:

$$(T_s + L_s) = (32 + 8) = 40 < (L_w + 1) = (44 + 1) = 45$$

The duty-cycle of the preamble search, and thus of the current drain profile, is **25%** The following table recaps the results:

Parameter	Unit	Sigfox value
Sub-pattern		0xAAAA
Sub-pattern length	Number of units pattern(s)	$L_s = 8$
Discontinuous RX cycle time (sub-windows periodicity)	Number of units pattern(s)	$T_s = 32$

Figure 9 – *Discontinuous RX results*

Note: the 0x5555 pattern (0xAAAA shifted by 1 bit) can be searched as well, but it requires to set the preamble again to 0xAAAA when the sub-pattern is detected, in order to make the decoding process work. Indeed, if we concatenate **DL-Pr** and **DL-FT**, the effective value is 0xAAAAB227, not 0x5555B227.

5 Device implementation

5.1 Hardware

Since it is necessary to control the radio layer directly, discontinuous RX can only be performed in discrete front ends which implements Sigfox library, as shown in the figure below.

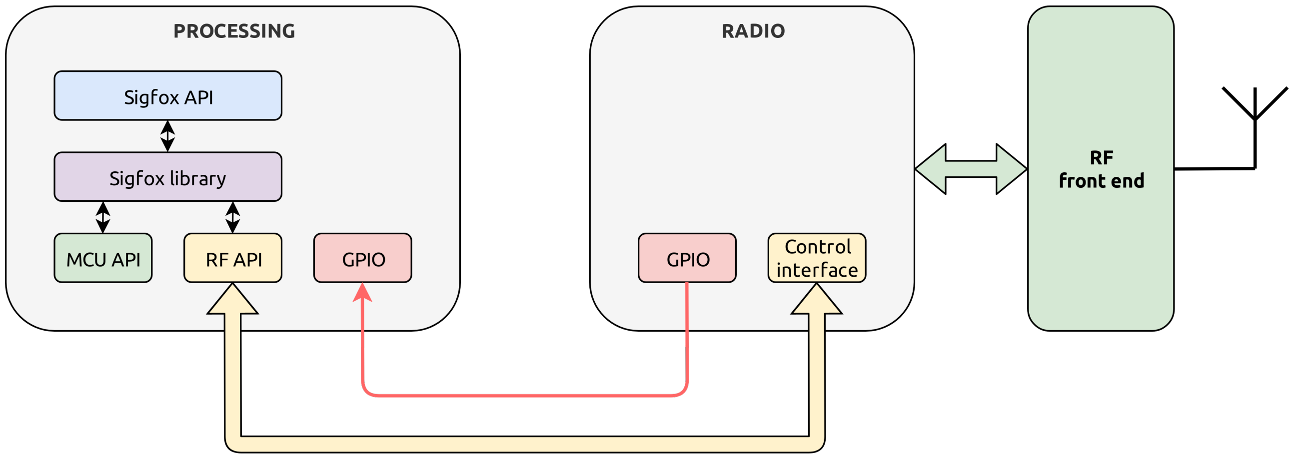

Figure 10 – *Discrete radio front-end architecture*

The GPIO link between processing and radio is not mandatory but is useful to save energy. For instance, the processing part is woken-up by interrupt instead of a continuously polling the control interface.

5.2 Radio configuration

The duty-cycled reception implementation depends on the transceiver capabilities and output options:

- In addition to physical radio parameters, most of transceivers can be programmed to detect and parse a specific pattern, composed of a preamble, a synchronization word and a payload of a given length. For a Sigfox downlink frame, parameters should be set as follow:
    - **Frequency:** computed internally by Sigfox library.
    - **Modulation:** GFSK.
    - **Bit rate** = 600 bits/s.
    - **Preamble** = 0xAAAA (see **section 4.3**).
    - **Synchronization word** = 0xB227.
    - **Payload length** = 15 (DL-ECC + DL-PAYLOAD + DL-AUTH + DL-CRC).
- Most of transceivers feature a configurable GPIO which can trigger an interrupt when the demodulator detects the programmed preamble.

5.3 Sigfox library (V2.10.0)

Whatever the reception technique, radio configuration is performed in the **RF_API_init()** and **RF_API_change_frequency()** functions of Sigfox library (**rf_api.c** source file).

Discontinuous RX operation should be implemented in the **RF_API_wait_frame()** function , which is called when waiting for the downlink frame and is responsible for the timeout $\mathbf{T_{RX}}$.

The cycle time has to take into account an **additional delay** L_t [seconds] required to wake-up the radio part, to re-configure it if needed, and to wait for local oscillator warm-up. This parameter is transceiver dependent, and should be evaluated on the target hardware.

Once T_s and L_s are set according to the previous criteria (see **section 4.2**), the corresponding durations T and L [seconds] depends on the bit rate D_r [bits/s]:

$$T=\frac{T_s \times L_u}{D_r} \qquad L=\frac{L_s \times L_u}{D_r}+L_t$$

Device implementation (radio configuration, T and L selection) has to be validated against frame error rate (**FER**) and downlink **sensitivity** target.

Below is an example of practical implementation:

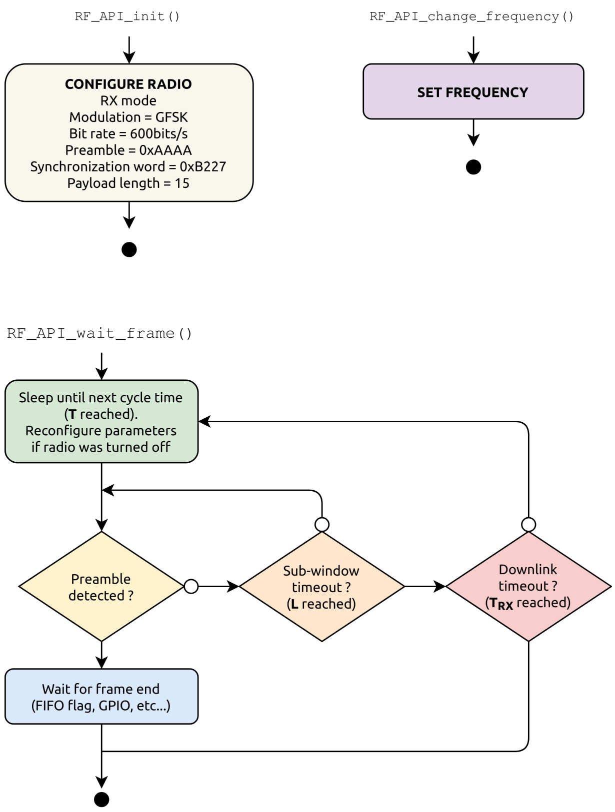

Figure 11 – *Discontinuous RX implementation example in Sigfox library*

Notes