Fraunhofer
IIS

FRAUNHOFER INSTITUTE FOR INTEGRATED CIRCUITS IIS

RFicient®
ULTRA-LOW POWER WAKE-UP RECEIVER DATASHEET

Confidential
Preliminary

Table of contents

Revision History:

Rev.	Changed content	Author	Date
0.1	initial draft	frs	20170808
0.2	fluent text functional description	mls	20170810
0.3	detailed register description	mls	20170822
0.3.2	Preliminary note	mls	20170922
0.4.	electrical characteristics added	mls	20171006
0.5	Pinlist, Pin- and Package Drawings, Operating conditions for Dune 3	epp	20170706
0.6	Application Circuit Drawing	frs	20170731
0.7	register description updated for Dune 3	mls	20180808
0.8	Updated pinout diagram for future release of Dune 4	epp	20181109
0.81	Updated pinout diagram and pin description table	epp	20281129
0.82	Added XTAL oscillator section	epp	20181214
0.83	Added LDO description	mls	20181214
0.9	Added new block diagram, updated abs. max ratings; VDD1V5 inconsistency removed	olr	20181217
0.91	Max. Temp 85°C in table 3	epp	20190131
0.92	PinOut Update	epp	20190507
0.94	PinOut Update; Changed VDD3V3 to VDDIO and VDD1P5 to VDDC	epp	20190529
0.95	FDD timeout specs added, table 6	mls	20190605
0.96	Sections 3, 3.1,3.2, 3.3, 3.4, 4 updated	mls, frs	20190628
0.97	Section 3.5 and register set updated	mls	20190701
0.98	Register set updated	mls	20190701
0.99	Register set updated	mls	20190704

1 General description ULP receiver technology RFicient®

The integrated ULP receiver technology RFicient® was developed for ISM frequency bands and built in standard CMOS technology. The receiver operates Multi-band sub-1-GHz, at 433 MHz, 868 MHz and 2.4 GHz and achieves a receiver sensitivity of -80 dBm. When operating in the standard configuration at a data rate of 1 kbit/s, the energy consumption is 3 µA at 1.5 V with a response time of only 32 ms.

The integrated ULP receiver RFicient® operates without the use of a microcontroller and recognizes two separate wake-up patterns. After receiving a specific wake-up pattern, a digital control signal is generated to activate any application hardware like a MCU.

Apart from the pure wake-up mode of operation, a selective activation using a 16-bit wide address range and reception of coded data streams is possible.

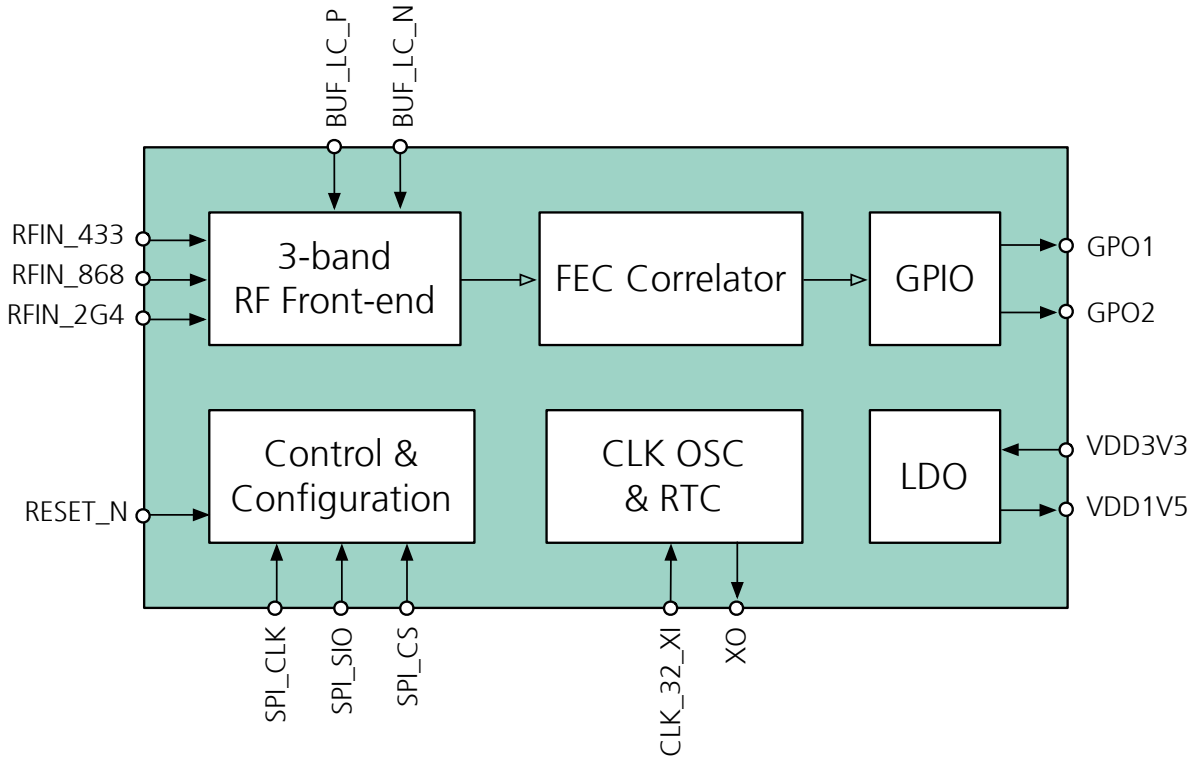


Figure 1: IP-Level Block Diagram

General description of the RFicient receiver

The RFicient Wake-Up receiver is a tri-band receiver for simultaneous reception of OOK modulated signals in the frequency bands 433 MHz, 868/915 MHz and 2.4 GHz. RF band selection is done by adjustment of the RX frequency and in addition with an external SAW filter. The receiver offers separated RF single-ended inputs for each frequency band at a high grade of flexibility. Applications can run with single, double or multiband antennas depending on the individual requirements.

The datarate for each RF band can be adjusted individually. The user can choose single-band, two-band or tri-band among operating modes during runtime. All configurations are set via the SPI interface. The tri-band receiver works as RF monitoring circuit that delivers IRQ events due to the incoming RF telegrams. Furthermore, user specific data packets can be received and read out in the FIFO memory for each frequency band simultaneously. An external 32.768 kHz clock source is needed or can be provided from the built-in sub-1-µA crystal oscillator with an external clock crystal.

Selective wake-up: Built-in address decoder

The RFicient receivers scans the three RF bands for incoming wake-up telegrams. These can be interpreted from the receiver as selective wake-up address. Hence, individual RF modules can be addressed without the need of a wireless sensor network. Moreover, addressing groups is also available. From the network prospective, a particular broadcast address that is assigned to a certain radio network can be used to address all items of one common network.

Data packets can be received by the RFicient receiver and stored in three built-in FIFO buffers. This can be combined with selective wake-up addressing. All of these data events can trigger an IRQ signal for external circuitry in order to indicate available data. Thus, peripheral components can be run in deep-sleep operating modes and achieve ultra-low values of the total power consumption.

Fast Data Decoder

The RFicient receiver can be run at a very low current consumption if the regular data rates are set at 1 kbps or lower. Recognizing a certain wake-up preamble, the receiver automatically switches the data rate to a predefined higher value (“fast” data rate), e.g. 8.192 kbps. As soon as the data packet is fully received at the higher data rate, the receiver reduces the data rate to the original lower value (“slow” data rate). By this, the average current consumption remains constant for rare data events. This benefits the user to combine ultra-low power radio reception with low latencies. The collected decoded bits are stored into a FIFO buffer with possible lenghts between 16 and 40 bits. If the FIFO buffer is filled, the receiver can generate an IRQ signal automatically. Thus, the user’s MCU can pick up the collected data packet at once.

Fault-tolerant preamble detection

Applying fault-tolerant data decoding, the RFicient receiver provides a robust unidirectional data communication. This is suitable for remote control or remote sensor applications. The On-Off-Keying (OOK) modulation was chosen for power consumption reasons. Moreover, OOK schemes require fault-tolerant data decoding without bidirectional communication. Thus, the RFicient communication uses binary correlators to detect predefined 31-bit preambles. Applying appropriate sequences with excellent autocorrelation and crosscorrelation attributes, the preamble detection can tolerate strong co-channel interferers with bit error rates up to 16%. A pair of 31-bit correlators are continuously fed with incoming OOK symbols provided from the ultra-low power analog front-end for each frequency band. Within the RFicient receiver, six binary correlators are implemented to provide simultaneous and independent tri-band data reception. The specified values for the receiver current consumption comprise both the analog front-end and the digital processing of the correlators, address and data decoders.

Built-in calibration circuits

The receive frequencies and internal comparator thresholds mostly suffer from variations from power supply and ambient temperature. For convenience, several calibrations are integrated within the RFicient receiver. Avoiding the need for external further reference signals, the built-in calibrations use the crystal frequency from the 32.768 kHz clock source. The user can initiate frequency and threshold calibrations at any desired point of time. Doing this during the receiver runtime, the reception is interrupted for less than ten milliseconds. Moreover, a third calibration is implemented and optimizes internal timing sequences.

Fast register access via SPI

Reading and writing the RFicient receiver configuration settings, data buffers, interrupt settings, event buffers and calibration data is carried out via SPI. Enabling very short operating times for peripheral MCUs, the SPI clock frequency is designed to be as high as 10 MHz. Thus, each register transfer may need only a microsecond. Wireless tags which require very small reaction times benefit from this.

IRQ generation: Data Events and RTC Timer Events

The user can choose between several possible events for automatic IRQ signal generation: Wake-up sequences A or B, FIFO buffer filled, FIFO buffer overflow, ID match. The SPI_SO/IRQ signal is set high if a specified event occurs. A 4-bit SPI register indicates the event type. Several built-in user-defined timers help to implement complex and accurate data protocols.

2 Pinning

2.1 QFN16 pinout

The RFicient® Wake-Up receiver pinning is shown in Figure 2. See section 6 for package outline dimensions.

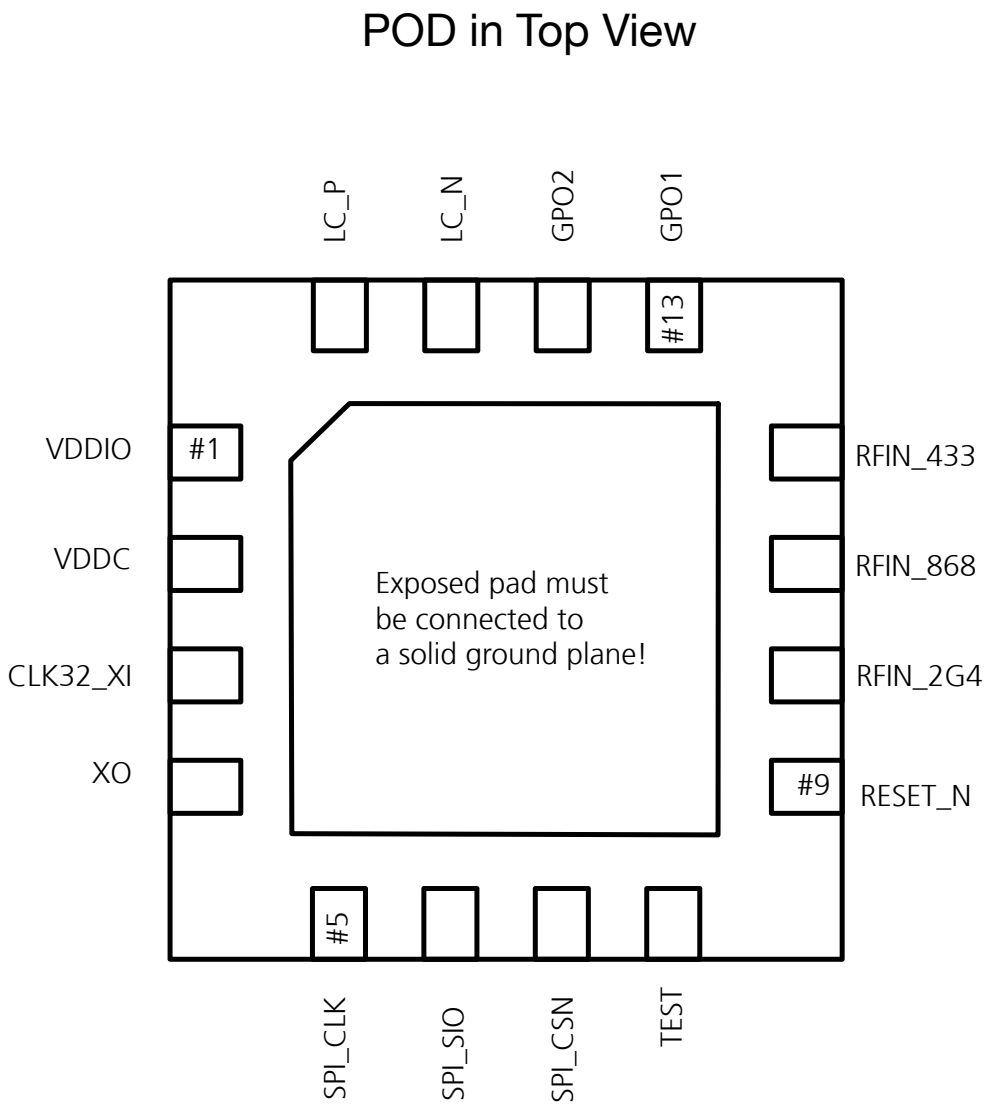


Figure 2: IC Pinout Top View

2.2 Pin description

Pin	Name	Type	Description
1	VDDC	Core Supply Voltage/ LDO Output	Connect 47-µF capacitor C3 for usage of internal LDO (see section 3.2) or forced external voltage supply, 1.5 V
2	VDDIO	IO Supply	Power Supply 1.8 to 3.3 V. Connect 47-µF capacitor C8 and 100 nF as C9 for usage of internal LDO.
3	CLK32_XI	Input	Crystal connector pin for operation with internal low power clock or system clock input, typically 32768 Hz for external clock operation
4	XO	Output	Crystal connector pin for operation with internal low power clock.
5	SPI_CLK	Input	SPI Clock Signal
6	SPI_SIO	In/Out	SPI Slave Input and Output. Note: This is a bidirectional signal.
7	SPI_CSN	Input	Spit Chip Select, Low active
8	TEST	Input	Test Enable, production Scan Test only. TEST input must be connected to GND.
9	RESET_N	Input	Reset to chip default settings, pull down for $t_{reset} = 4 \cdot t_{CLK\_32}$
10	RFIN_2G4	RF-Input	LNA input, 2.4-GHz band
11	RFIN_868	RF-Input	LNA input, 868-MHz band
12	RFIN_433	RF-Input	LNA input, 433-MHz band
13	GPO2	Output	General Purpose Output 2
14	GPO1	Output	General Purpose Output 1 (see Section xxx)
15	LC_N	Analog	Connect External LC-Filter. Use 220 nH as L200.
16	LC_P	Analog	Connect External LC-Filter. Use 72 pF as C200.
EP	GND	Supply	Connect exposed pad to Ground

Table 1: IC Pinout Overview

3 Typical application circuit

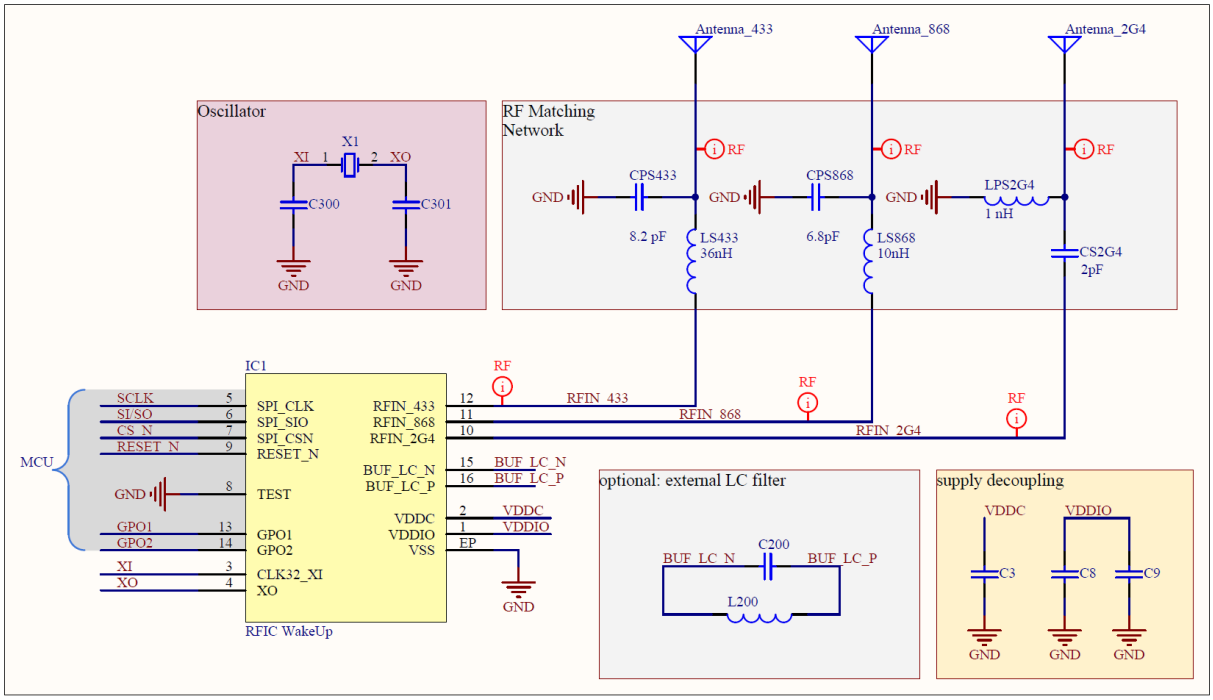


Figure 3: Typical application circuit for UHF triband operation

3.1 Crystal Oscillator

Features

- Built-in 32.768 kHz Real Time Clock crystal oscillator
- Characterized with crystal loading capacitors ranging from 4 pF to 20 pF
- Low power design (< 1.2 µW typical) operates on core VDD only
  - I_{AVG} < 0.8 µA (typical model, 25°C)
  - P_{LEAK} ≈ 75 nW (typical model, 25°C)
- Oscillator enable/disable via SPI register
- Oscillator tuning via SPI register

The product is equipped with an ultra low power oscillator suitable for 32768-Hz clock quartz crystals. Pins CLK32_XI and XO have to be connected to the quartz crystal for oscillator operation. Table 2 lists recommended quartz crystals. Make sure that two additional capacitors C300 and C301 are needed due to C_{Load} specifications.

The crystal oscillator enable controlled via register setting. When disabled an external system clock can be fed to the CLK32_XI pin. XO is left open in this case, i.e. the bypass mode. The external clock signal at pin CLK32_XI must not exceed the core power (VDDC) voltage level.

In a later version, the crystal oscillator is digitally tunable via register setting to corrected clock-drift of the real time clock counter (RTC). For operating instructions of the RTC see section 5, description of registers.

Recommended values for C300 and C301 are C300 = C301 ≈ 2· (C_{Load} – C_{Shunt} – 1 pF).

C_{Shunt} depends on the chosen crystal part.

No.	Part Number	Type	Drive Level	C_{Load}	C_{Shunt}	Form factor
1	ECX-.327-CDX-1293 SMD CRYSTAL	SMD	100 nW, max. 1 µW	12.5 pF	1.05 pF	3.2 x 1.5 x 0.9 mm^3
2	LFXTAL009709 Bulk	SMD	max. 1 µW	7.00 pF	1.05 pF	3.2 x 1.5 x 0.5 mm^3
3	LFXTAL002995 Bulk	Cylinder	max. 1 µW	12.50 pF	2.5 pF	∅3.0 x 8.0mm

Table 2: Recommended quartz crystals for built-in clock oscillator

Parameter	Description	Min.	Typ.	Max.	Units
VDDC	Core supply voltage	1.35	1.5	1.65	V
VDDD	I/O supply voltage Options	2.97	3.3	3.63	V
		2.25	2.5	2.75	V
		1.62	1.8	1.98	V
Tj	Junction temperature	-40	25	85	°V
VPAD	Voltage at CLK32_XI	0		VDDC	V

Table 3: Recommended operating conditions for oscillator

3.2 Low-Drop-Out Regulator

Features

- ❖ Built-in low-drop out voltage regulator
- ❖ two external 47-µF SMD capacitors are needed
- ❖ very low leakage current

An built-in LDO provides the opportunity to apply higher supply voltages between 1.8 V and 3.3 V e.g. from a battery. A regulated 1.5-V core voltage is generated by the LDO. An internal bandgap circuit operates as voltage reference for the regulator. The LDO adjusts the core voltage periodically in pulse operation. The residual voltage deviation is kept below 10 mV for 1.50 V typical core voltage. An external 47-µF capacitor is needed for stabilization. The equivalent series resistance of the capacitor must be lower than 0,1 Ω for frequencies above 1 kHz. Lithium type button cells are recommended as voltage supply.

The internal resistance of the applied battery shall not exceed 10 Ω. The LDO also manages power-up procedures of the receiver and of the built-in crystal oscillator. Stable operation after settling processes is ensured.

3.3 Extended Interrupt Generator

Features

- Eight independent interrupt events selectable by user
- Built-in ID address decoders for wake-up events
- individual wake-up using 16-bit node ID,
- groupwise wake-up or general ("broadcast") wake-up.
- status of RX data buffer ("FIFO") can trigger interrupt events
- four built-in RTC timers can trigger alarm interrupts for user-defined timings within protocols
- programmable cyclic timer for periodical interrupts
- Built-in 40-Bit clock counter

Event Type No.	Event Name	Remarks
0	ID match	16-Bit-ID (Reg. 0x35, 0x36: ID_HI & ID_LO) in FDD mode fits to received Fast Data Packet
1	FIFO overflow	indicates FIFO overflow if FDD data bits extend the specified FIFO length (Reg. 0x56): 16/24/32/40 bits.
2	FIFO buffer filled	indicates that FIFO is filled with FDD data bits according to the specified FIFO length (Reg. 0x56): 16/24/32/40 bits.
3	Code A or B detected	Correlation sequences A or B detected (no matter if FDD or Slow Mode is active)
4	ID match & FIFO filled	event type 0 occurred within the last <FIFO_length +2> bit cycles and event type 2 occurred after event type 0 ("telegram received")
5	ID match & SLOW	event type 0 occurred within the current FDD data reception and transition from FAST to SLOW mode occurred after event type 0 ("end of telegram")
6	RTC Timer Alarm	one of the 4 RTC timers (RTCSH0, RTCSH1, RTCLG0, RTCLG1) have reached the user-defined targets and caused alarm. Register RTC_EVENTS indicates which alarm rings.
7	Cyclic Timer Alarm	the cyclic timer has reached the user-defined target and caused alarm

Table 4: Available interrupt event types

3.4 Real-Time Clock and Built-in Timers

Features

- User-defined timing schemes can be triggered by five built-in timers:
- Two short timers up 2 seconds
- Two short timers up to 1 year 23 days
- One cyclic alarm timer up to 8 min 30 s
- Built-in clock offers unique time stamps for one full year

Each IC offers a 40-bit clock counter as a built-in system clock that comprises 1 year 23 days before overflow. Derived from that, four user-defined RTC timers are provided with a granularity of 30.517578 µs (i.e. 1 clock cycle):

Name	Function	Description
RTCSH0	16-Bit Short-term timer	Single Timer with a max. duration of 1.99996495 s (i.e. 65535 clock cycles)
RTCSH1	16-Bit Short-term timer	Single Timer with a max. duration of 1.99996495 s
RTCLG0	40-Bit Short-term timer	Single Timer with a max. duration of 33554432 s (i.e. 388 d 8h 40 min 32 s or 2^{40} -1 clock cycles)
RTCLG1	40-Bit Short-term timer	Single Timer with a max. duration of 33554432 s
CYCL	16-Bit cyclic timer	Provides periodic alarms out of the system clock divided by M, M can be 1 until 255. Thus, a maximum alarm period of 8 min 29.9922 s (i.e. 16711425 clock cycles) can be adjusted.

Table 5: Available RTC timers

3.5 Multipurpose Signal Outputs

The RFicient® receiver offers numerous internal signals that can be selected and provided at two output pins GPO1 and GPO2. Writing register 0x75, the user can choose according to this table:

MUX_D_OUT_SEL	"GPO1" Pin 13	"GPO2" Pin 14	Description
0	FAST_CLK_2G4	FAST_DATA_2G4	RX data stream at 2.4 GHz
1	FAST_CLK_868	FAST_DATA_868	RX data stream at 868 MHz
2	FAST_CLK_433	FAST_DATA_433	RX data stream at 433 MHz
3	WUP_A_2G4	WUP_B_2G4	WakeUp signals at 2.4 GHz
4	KOMP_OUT_W	RX_ACTIVE	Test signals
5	KOMP_OUT_M	KOMP_OUT_S	Test signals
6	ID_MATCH	WUP_A_433	General ID match, WakeUp signal
7	WUP_A_868	WUP_B_868	WakeUp signals at 868 MHz
8	WUP_A_433	RX_ACTIVE	WakeUp signal, trigger signal
9	WUP_A_868	RX_ACTIVE	WakeUp signal, trigger signal
10	WUP_A_2G4	RX_ACTIVE	WakeUp signal, trigger signal
11	KOMP_OUT_M	RX_ACTIVE	Test signals
12	KOMP_OUT_S	RX_ACTIVE	Test signals
13	KOMP_OUT_W	WUP_A_868	Test signal, WakeUp signal
14	CLK32	IRQ_EVENT	System clock, Interrupt signal
15 (chip default)	IRQ_EVENT	CLK32	Interrupt signal, system clock

Table 6: Selection table for GPO1/GPO2 signals

3.6 RFicient® Protocol: Reception of ID and User Data

RFicient® offers ultra-low values of current consumption (<3 µA) at a low latency. Many wireless applications can be distinguished between "RF listening" and "RF data reception". Spontaneous wake-up events require a low system latency (e.g. <32 ms) at a low current consumption (e.g. <3 µA). Consequently, long-term battery-driven applications achieve several years of operation with a single battery and keep RF connected continuously. It seems likely that a RF wake-up event is followed by a short radio telegram containing an ID, remote instructions and config parameters. This user data should be processed very quickly, i.e. at a high data rate in order to occupy the radio channel only for a short time. RFicient® may provides both functions:

a) wake-up reception and

b) ID and data reception.

It is very useful for wireless networks when wake-up calls offer a kind of selectivity among numerous other RF nodes. Thus, RFicient® features three kinds of selective wake-up methods: individual wake-up, groupwise wake-up and broadcast wake-up.

The above mentioned features require a predefined protocol scheme that is the two-stage RFicient® protocol:

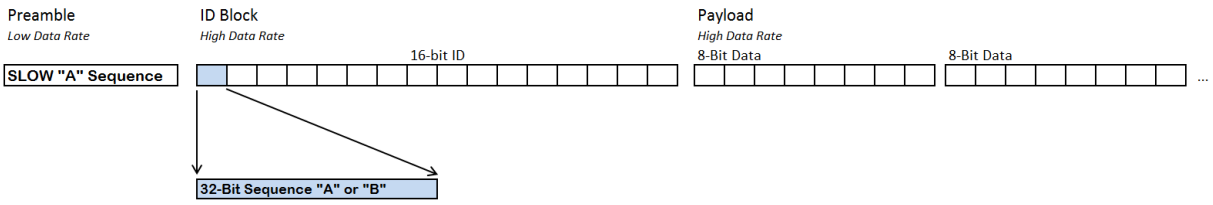


Figure 4: Two-stage RFicient® wireless protocol

The user can enable the fast data decoder (FDD) for each frequency band using register FDD_ENABLE. Consequently, the RFicient receiver interprets incoming Code A sequences at the slow data rate as initiator for FDD packets. During fast data rate mode, the receiver interprets incoming fast Code A sequences as logical "0" and fast Code B sequences as logical "1". This ensures a bit error tolerant reception of data packets for a robust unidirectional data communication. Slow Code A or B sequences are ignored in fast mode, as well as Code A or B sequences in data rates other than the FAST data rate. Having properly received the 16-bit ID, the receiver compares this with the local ID of the receiver node (see registers 0x35, 0x36). In case of an ID match, an interrupt event "ID match" might be triggered and further incoming "fast mode" data bits are stored in the FIFO data buffer for each frequency band separately. If no further fast data mode bits are being received, the receiver quits fast data mode and switches back to slow data rate. The user can choose if this end-of-data event may cause an interrupt.

Example:
Assuming the low data rate with 1024 bps and 32768 bps as high data rate, the preamble needs 31.25 ms. Each fast user bit is represented by a fast code sequence (“A” or “B”) and needs 0.977 ms each. Thus, the 16-bit ID is transmitted within 15.625 ms. A further 8-bit data packet takes 7.813 ms. An entire frame starting with slow “A”, having a 16-bit ID plus two 8-bit data packets exhibits a total duration of 62.5 ms.

4 Electrical specifications

4.1 Absolute Maximum Ratings

Parameter	Min.	Max.	Unit	Condition
Supply Voltage Core VDDC	-0.5	1.65	V	
Supply Voltage IO VDDIO	-0.5	3,63	V	
Voltage on any pin	-0.5	VDDIO + 0,5	V	
Input RF level		0	dBm	
Storage temperature	-40	125	°C	
Reflow solder temperature		tbd	°C	

Table 7: Absolute maximum ratings

4.2 Operating Conditions

Parameter	Min.	Typ.	Max.	Unit	Condition
Supply Voltage Core VDDC	1.35	1.5	1.65	V	
Supply Voltage VDDIO	1.8	3.0	3.63	V	
Operating ambient temperature	-40	27	85	°C	

Table 8: Operating conditions

4.3 Electrical characteristics

Parameter	Min.	Typ.	Max.	Unit	Condition
Supply Current					
1 kbps 868 MHz single band	1.8	2.1		µA	typ. operation: Supply voltage 1.5 V, ambient temp. 27°C, TSP 1
1 kbps 2.4 GHz single band	1.8	2.1		µA	
1 kbps 433 MHz single band	1.8	2.0		µA	
1 kbps 868 MHz/2.4 GHz dual band	2.3	2.6		µA	
1 kbps 433 MHz/868 MHz dual band	2.3	2.7		µA	
1 kbps 433 MHz/2.4 GHz dual band	2.3	2.7		µA	
1 kbps 433 MHz/868 MHz/2.4 GHz triple band	3.0	3.3		µA	
16 kbps 868 MHz single band		13.2		µA	
8 kbps 868 MHz single band		7.3		µA	
4 kbps 868 MHz single band		4.4		µA	
2 kbps 868 MHz single band		2.9		µA	
1 kbps 868 MHz single band		2.1		µA	
512 bps 868 MHz single band		1.8		µA	
256 bps 868 MHz single band		1.6		µA	
RF sensitivity @ 433 MHz		-70		dBm	preliminary
RF sensitivity @ 868 MHz		-70		dBm	preliminary
RF sensitivity @ 2.4 GHz		-70		dBm	preliminary

Table 9: Electrical characteristics

4 The 3-Wire SPI Interface

The Wake-Up IC can be configured and read out by a 3-Wire SPI interface, where the RFicient® IC acts as slave. Any SPI transaction start with a header byte containing a (R/$\overline{W}$) bit and 7 address bits A6..A0 followed by the 8-bit data byte. SPI read and write access is shown in Figure 5. Each SPI access is initiated by falling edge and stop by rising edge of SPI_CS_N pin. SPI_CS_N must be low during the SPI read/write process.

Note: For read access, the SPI_SIO pin changes its data direction to output after eight SPI_CLK clocks, in order to avoid cross-currents at the SPI_SIO pin the SPI Master should change its data direction within the time t_{sad}. If the master can not guarantee this then a serial resistor should be used between master and slave.

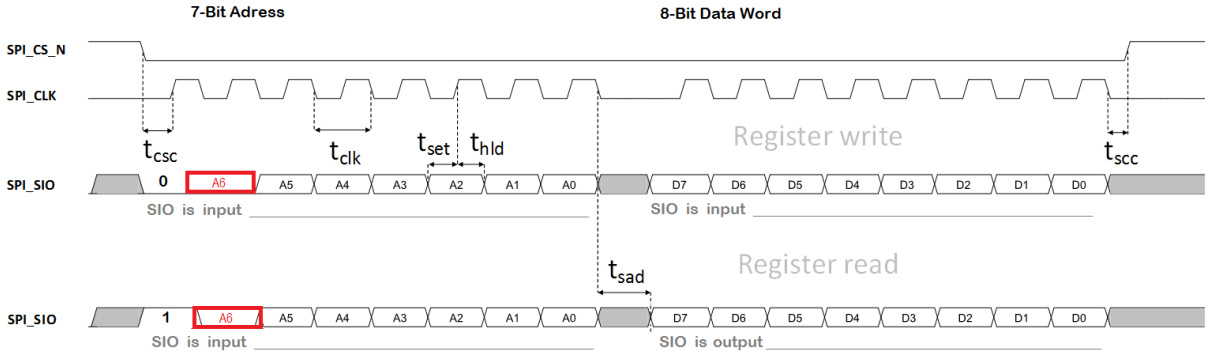


Figure 5: SPI register read/write operation

Parameter	Description	Min.	Max.	Units
t_{clk}	SPI_CLK period	100	-	ns
t_{csc}	Time SPI_CS_N falling edge to first SPI_CLK rising edge	5	-	ns
t_{set}	SPI_SI setup, data must be stable before SPI_CLK rising edge	15	-	ns
t_{hld}	SPI_SI data hold	5	-	ns
t_{sad}	SPI_SO data setup	5	10	ns
t_{scc}	Last SPI_CLK falling edge to SPI_CS_N rising edge	5	-	ns

Table 10: SPI timing requirements

5 Register Set

		128	64	32	16	8	4	2	1	Hex-	wer schreibt?	Chip-
Adresse	Registername	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Adresse	User/Chip/both	Default
0	NFA433_SLOW						NFA433_SLOW<2>	NFA433_SLOW<1>	NFA433_SLOW<0>	0x00	User	0x05
1	NFA433_FAST						NFA433_FAST<2>	NFA433_FAST<1>	NFA433_FAST<0>	0x01	User	0x00
2	NFA868_SLOW						NFA868_SLOW<2>	NFA868_SLOW<1>	NFA868_SLOW<0>	0x02	User	0x05
3	NFA868_FAST						NFA868_FAST<2>	NFA868_FAST<1>	NFA868_FAST<0>	0x03	User	0x00
4	NFA2G4_SLOW						NFA2G4_SLOW<2>	NFA2G4_SLOW<1>	NFA2G4_SLOW<0>	0x04	User	0x05
5	NFA2G4_FAST						NFA2G4_FAST<2>	NFA2G4_FAST<1>	NFA2G4_FAST<0>	0x05	User	0x00
6	CALIB_STATUS					OFFSET_CAL_IN_PROG	SPG_CAL_IN_PROG	LCO_CAL_IN_PROG	CAL_IN_PROG	0x06	Chip	0x00
7	CALIB_CTRL					OFFSET_CAL	SPG_CAL	LCO_CAL	CAL_START	0x07	both	0x0E
8	TSP_CTRL			CORNER_CTRL<1>	CORNER_CTRL<0>	SPG_TSP<3>	SPG_TSP<2>	SPG_TSP<1>	SPG_TSP<0>	0x08	both	0x08
9	N_SPG_TARGET	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x09	User	0x31
10	SPG_FREQ	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x0A	Chip	0x00
11	N_LCO_TARGET_433				<12>	<11>	<10>	<9>	<8>	0x0B	User	0x0E
12	N_LCO_TARGET_433	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x0C	User	0x20
13	N_LCO_TARGET_868				<12>	<11>	<10>	<9>	<8>	0x0D	User	0x0D
14	N_LCO_TARGET_868	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x0E	User	0x87
15	N_LCO_TARGET_2G4				<12>	<11>	<10>	<9>	<8>	0x0F	User	0x12
16	N_LCO_TARGET_2G4	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x10	User	0xCE
17	LCO_RANGE_433			<5>	<4>	<3>	<2>	<1>	<0>	0x11	both	0x20
18	LCO_RANGE_868			<5>	<4>	<3>	<2>	<1>	<0>	0x12	both	0x21
19	LCO_RANGE_2G4			<5>	<4>	<3>	<2>	<1>	<0>	0x13	both	0x22
20	LCO_FREQ_433				<12>	<11>	<10>	<9>	<8>	0x14	Chip	0x00
21	LCO_FREQ_433	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x15	Chip	0x00
22	LCO_FREQ_868				<12>	<11>	<10>	<9>	<8>	0x16	Chip	0x00
23	LCO_FREQ_868	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x17	Chip	0x00
24	LCO_FREQ_2G4				<12>	<11>	<10>	<9>	<8>	0x18	Chip	0x00
25	LCO_FREQ_2G4	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x19	Chip	0x00
26	COMPREF_W_433		<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x1A	both	0x3A
27	COMPREF_W_868		<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x1B	both	0x3A
28	COMPREF_W_2G4		<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x1C	both	0x3A
29	COMPREF_M_433		<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x1D	both	0x3A
30	COMPREF_M_868		<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x1E	both	0x3A
31	COMPREF_M_2G4		<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x1F	both	0x3A
32	COMPREF_S_433		<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x20	both	0x3A
33	COMPREF_S_868		<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x21	both	0x3A
34	COMPREF_S_2G4		<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x22	both	0x3A
35	D_CORNER_CTRL			CRN_S_ENA_SPG	CRN_F_ENA_SPG	CRN_S_ENA_MUXMIX	CRN_F_ENA_MUXMIX	CRN_S_ENA_LCO	CRN_F_ENA_LCO	0x23	User	0x00
36	BAND_BRANCH_CTRL		2G4	868	433		STRONG	MEDIUM	WEAK	0x24	User	0x77
37	TESTBUF_CTRL	DEBUG_EN		DEMS_TESTBUF_ENABLE	MM_TESTBUF_ENABLE	MW_TESTBUF_ENABLE	F1_TESTBUF_ENABLE	F2_TESTBUF_ENABLE	F3_TESTBUF_ENABLE	0x25	User	0x00
38	IFAMP_GAIN_CTRL			<5>	<4>	<3>	<2>	<1>	<0>	0x26	User	0x15
39	RX_ACTIVE_SELECT							BAND_ACTIVE<1>	BAND_ACTIVE<0>	0x27	both	0x03
40	CODE_SELECT	CODE_B<3>	CODE_B<2>	CODE_B<1>	CODE_B<0>	CODE_A<3>	CODE_A<2>	CODE_A<1>	CODE_A<0>	0x28	User	0x10
41	KORREL_THRESH_A				<4>	<3>	<2>	<1>	<0>	0x29	both	0x1A
42	KORREL_THRESH_B				<4>	<3>	<2>	<1>	<0>	0x2A	both	0x1A
43	KORREL_STATE	BAND<1>	BAND<0>	B_STRONG	B_MEDIUM	B_WEAK	A_STRONG	A_MEDIUM	A_WEAK	0x2B	Chip	0x00
44	KORREL_VAL	KORREL_VAL_B<3>	KORREL_VAL_B<2>	KORREL_VAL_B<1>	KORREL_VAL_B<0>	KORREL_VAL_A<3>	KORREL_VAL_A<2>	KORREL_VAL_A<1>	KORREL_VAL_A<0>	0x2C	Chip	0x00
45	FDD_ENABLE						FDD_2G4	FDD_868	FDD_433	0x2D	User	0x07
46	FDD_ACTIVE						FDD_ACTIVE_433	FDD_ACTIVE_868	FDD_ACTIVE_2G4	0x2E	Chip	0x00
47	FO_QUIT						FO_QUIT_2G4	FO_QUIT_868	FO_QUIT_433	0x2F	User	0x00
48	FDD_EXIT_COND			<5>	<4>	<3>	<2>	<1>	<0>	0x30	Chip	0x00
49	IRQ_SELECT	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x31	User	0x01
50	IRQ_STATUS	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x32	Chip	0x00
51	IRQ_CLR	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x33	User	0x00
52	IRQ_SET	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x34	User	0x00
53	ID_HI	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x35	User	0x7D
54	ID_LO	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x36	User	0xA8
55	IDM_ENABLE						IDM_2G4	IDM_868	IDM_433	0x37	User	0x07
56	IDM_CTRL							IDM_CTRL<1>	IDM_CTRL<0>	0x38	User	0x00
57	IDM_CLR								IDM_CLR_STATUS	0x39	User	0x00
58	IDM_BAND							IDM_BAND<1>	IDM_BAND<0>	0x3A	Chip	0x00
59	IDM_REASON							IDM_REASON<1>	IDM_REASON<0>	0x3B	Chip	0x00
60	RTC_SELECT				<4>	<3>	<2>	<1>	<0>	0x3C	User	0x00
61	RTC_STATUS					<3>	<2>	<1>	<0>	0x3D	User	0x00
62	RTC_CLR					<3>	<2>	<1>	<0>	0x3E	User	0x00
63	RTCSH0_THRESH	<15>	<14>	<13>	<12>	<11>	<10>	<9>	<8>	0x3F	User	0x00
64	RTCSH0_THRESH	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x40	User	0x00
65	RTCSH1_THRESH	<15>	<14>	<13>	<12>	<11>	<10>	<9>	<8>	0x41	User	0x00
66	RTCSH1_THRESH	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x42	User	0x00
67	RTCLG0_THRESH	<39>	<38>	<37>	<36>	<35>	<34>	<33	<32>	0x43	User	0x00
68	RTCLG0_THRESH	<31>	<30>	<29>	<28>	<27>	<26>	<25>	<24>	0x44	User	0x00
69	RTCLG0_THRESH	<23>	<22>	<21>	<20>	<19>	<18>	<17>	<16>	0x45	User	0x00
70	RTCLG0_THRESH	<15>	<14>	<13>	<12>	<11>	<10>	<9>	<8>	0x46	User	0x00
71	RTCLG0_THRESH	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x47	User	0x00
72	RTCLG1_THRESH	<39>	<38>	<37>	<36>	<35>	<34>	<33	<32>	0x48	User	0x00
73	RTCLG1_THRESH	<31>	<30>	<29>	<28>	<27>	<26>	<25>	<24>	0x49	User	0x00
74	RTCLG1_THRESH	<23>	<22>	<21>	<20>	<19>	<18>	<17>	<16>	0x4A	User	0x00
75	RTCLG1_THRESH	<15>	<14>	<13>	<12>	<11>	<10>	<9>	<8>	0x4B	User	0x00
76	RTCLG1_THRESH	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x4C	User	0x00
77	CYCLPRESC	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x4D	User	0x00
78	CYCLTOP	<15>	<14>	<13>	<12>	<11>	<10>	<9>	<8>	0x4E	User	0x00
79	CYCLTOP	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x4F	User	0x00
80	CNTR40	<39>	<38>	<37>	<36>	<35>	<34>	<33	<32>	0x50	Chip	0x00
81	CNTR40	<31>	<30>	<29>	<28>	<27>	<26>	<25>	<24>	0x51	Chip	0x00
82	CNTR40	<23>	<22>	<21>	<20>	<19>	<18>	<17>	<16>	0x52	Chip	0x00
83	CNTR40	<15>	<14>	<13>	<12>	<11>	<10>	<9>	<8>	0x53	Chip	0x00
84	CNTR40	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x54	Chip	0x00
85	CNTR40_CLR								<0>	0x55	User	0x00
86	FIFO_LENGTH			<5>	<4>	<3>	<2>	<1>	<0>	0x56	User	0x14
87	FIFO_COUNT_433			<5>	<4>	<3>	<2>	<1>	<0>	0x57	both	0x00
88	FIFO_COUNT_868			<5>	<4>	<3>	<2>	<1>	<0>	0x58	both	0x00
89	FIFO_COUNT_2G4			<5>	<4>	<3>	<2>	<1>	<0>	0x59	both	0x00

		128	64	32	16	8	4	2	1	Hex-	wer schreibt?	Chip-
Adresse	Registername	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Adresse	User/Chip/both	Default
90	RX_FIFO_5_433	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x5A	Chip	0x00
91	RX_FIFO_4_433	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x5B	Chip	0x00
92	RX_FIFO_3_433	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x5C	Chip	0x00
93	RX_FIFO_2_433	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x5D	Chip	0x00
94	RX_FIFO_1_433	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x5E	Chip	0x00
95	RX_FIFO_0_433	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x5F	Chip	0x00
96	RX_FIFO_5_868	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x60	Chip	0x00
97	RX_FIFO_4_868	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x61	Chip	0x00
98	RX_FIFO_3_868	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x62	Chip	0x00
99	RX_FIFO_2_868	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x63	Chip	0x00
100	RX_FIFO_1_868	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x64	Chip	0x00
101	RX_FIFO_0_868	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x65	Chip	0x00
102	RX_FIFO_5_2G4	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x66	Chip	0x00
103	RX_FIFO_4_2G4	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x67	Chip	0x00
104	RX_FIFO_3_2G4	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x68	Chip	0x00
105	RX_FIFO_2_2G4	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x69	Chip	0x00
106	RX_FIFO_1_2G4	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x6A	Chip	0x00
107	RX_FIFO_0_2G4	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x6B	Chip	0x00
108	ACTUAL_NFA_433		NFA433_FAST<2>	NFA433_FAST<1>	NFA433_FAST<0>		NFA433_SLOW<2>	NFA433_SLOW<1>	NFA433_SLOW<0>	0x6C	Chip	0x25
109	ACTUAL_NFA_868		NFA868_FAST<2>	NFA868_FAST<1>	NFA868_FAST<0>		NFA868_SLOW<2>	NFA868_SLOW<1>	NFA868_SLOW<0>	0x6D	Chip	0x15
110	ACTUAL_NFA_2G4		NFA2G4_FAST<2>	NFA2G4_FAST<1>	NFA2G4_FAST<0>		NFA2G4_SLOW<2>	NFA2G4_SLOW<1>	NFA2G4_SLOW<0>	0x6E	Chip	0x25
111	ACTUAL_BANDSELECT						2G4	868	433	0x6F	Chip	0x07
112	GENPURP_0	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x70	User	0x00
113	GENPURP_1	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x71	User	0x00
114	GENPURP_2	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x72	User	0x0F
115	XTAL_OSC_CTRL								<0>	0x73	User	0x01
116	XTAL_OSC_TRIM	<7>	<6>	<5>	<4>	<3>	**<2>**	**<1>**	**<0>**	0x74	User	0x04
117	MUX_D_OUT_SEL					<3>	<2>	<1>	<0>	0x75	User	0x0F
118	LC_TG_ENA								<0>	0x76	User	0x01
119	XTAL_GOOD								<0>	0x77	Chip	0x00
120	**COMP_THRESH_W**		<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x78	User	0x04
121	**COMP_THRESH_M**		<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x79	User	0x14
122	**COMP_THRESH_S**		<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x7A	User	0x14
123	**OFFSETCAL_CTRL**	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x7B	User	
124	**KORREL_SV_CLEAR**								<0>	0x7C	User	
125										0x7D		
126										0x7E		
127	VERSION	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	0x7F	Chip	0x41

Table 11: RFicient® register set

Detailed description of registers

Note: Not all registers are described in detail yet in this document version.

Note: Registers marked with * are for internal use only and not recommended for user access.

Register 0x00: NFA433_SLOW
Bits 2..0: NFA433_SLOW sets the sample rate for slow RX operation in the 433-MHz band. Table 4 shows the meaning of all NFA settings. Higher sample rates allow lower reaction times due to lower wake-up sequence durations. However, the current consumption of the receiver increases with higher sample rates. For ultra-low power operation achieving lowest current consumptions, the uses should choose NFA = 111_2. Thus, a 32-bit wake-up sequence has a duration of 125 ms.

NFA<2:0>	Sample Rate	Wake-Up Sequence Duration	Unit	Comment
000	32768 Hz	0.977	ms	fastest for mono-band
001	16384 Hz	1.953	ms	fastest for dual-band, fastest for 868-MHz band in tri-band mode
010	8192 Hz	3.906	ms	fastest for 433-MHz and 2.4-GHz band in tri-band mode
011	4096 Hz	7.813	ms	
100	2048 Hz	15.625	ms	
101	1024 Hz	31.25	ms	typical
110	512 Hz	62.5	ms	
111	256 Hz	125.00	ms	slowest

Table 12: Sampling modes for various NFA settings

In dual-band or tri-band mode the user can check which data rates are actually effective, see registers 0x6C..0x6F ACTUAL_NFA and ACTUAL_BANDSELECT.

Register 0x01: NFA433_FAST
Bits 2..0: NFA433_FAST sets the sample rate for fast RX operation within enabled FDD mode for the 433-MHz band, according to table 12.

Register 0x02: NFA868_SLOW
Bits 2..0: NFA868_SLOW sets the sample rate for slow RX operation in the 868-MHz band.

Register 0x03: NFA868_FAST
Bits 2..0: NFA868_FAST sets the sample rate for fast RX operation within enabled FDD mode for the 868-MHz band.

Register 0x04: NFA2G4_SLOW
Bits 2..0: NFA2G4_SLOW sets the sample rate for slow RX operation in the 2.4-GHz band.

Register 0x05: NFA2G4_FAST
Bits 2..0: NFA2G4_FAST sets the sample rate for fast RX operation within enabled FDD mode for the 2.4-GHz band.

Register 0x06 CALIB_STATUS
Bit 0 (CAL_IN_PROG) is 0 if no calibration is in progress. The receiver automatically sets and clears this bit during calibration. The user can check this bit before accessing registers.
Bits 3:1 select calibration methods.

Register 0x07 CALIB_CTRL
If the user sets bit 0 to "1", the selected calibration methods are being initiated. Bits 3:1 select each calibration method.

Register 0x08 TSP_CTRL *
This register determines the pulse operation of the receiver. Write access is not recommended.

Register 0x09 N_SPG_TARGET *
This register is reserved for proper pulse timing. Write access is not recommended.

Register 0x0A SPG_FREQ *
This register can be used to check pulse timing.

Register 0x0B N_LCO_TARGET_433, Register 0x0C N_LCO_TARGET_433, Register 0x0D N_LCO_TARGET_868, Register 0x0E N_LCO_TARGET_868, Register 0x0F N_LCO_TARGET_2G4, Register 0x10 N_LCO_TARGET_2G4 *:
These registers are reserved for internal frequency adjustment. Write access is not recommended.

Register 0x11 LCO_RANGE_433, Register 0x12 LCO_RANGE_868, Register 0x13 LCO_RANGE_2G4 *:
These registers are reserved for internal frequency adjustment. Write access is not recommended.

Register 0x14 LCO_FREQ_433, Register 0x15 LCO_FREQ_433, Register 0x16 LCO_FREQ_868, Register 0x17 LCO_FREQ_868, Register 0x18 LCO_FREQ_2G4, Register 0x19 LCO_FREQ_2G4 *:
These registers are reserved for internal frequency adjustment and are used to check internal frequencies.

Register 0x1A COMPREF_W_433, Register 0x1B COMPREF_W_868, Register 0x1C COMPREF_W_2G4, Register 0x1D COMPREF_M_433, Register 0x1E COMPREF_M_868, Register 0x1F COMPREF_M_2G4, Register 0x20 COMPREF_S_433, Register 0x21 COMPREF_S_868, Register 0x22 COMPREF_S_2G4 *:
These registers are reserved for internal offset adjustment. Write access is not recommended.

Register 0x23 D_CORNER_CTRL *

This register is reserved for internal adjustment. Write access is not recommended.

Register 0x24 BAND_BRANCH_CTRL

Bits 2:0 select the RX branches for various RF sensitivities. Write "1" into bit 0 for reception of weak RF signals, bit 1: for medium RF signals, bit 2: for strong RF signals (e.g. very short distance). Chip default is "111", i.e. all RX branches are activated.

Bits 6:4 select the desired RF frequency bands for wakeUp reception. Note: simultaneous reception at two or three frequency bands is possible!
Bit 4 selects the 433-MHz band, bit 5 selects the 868-MHz band, bit 6 selects the 2.4-GHz band.

BAND_BRANCH_CTRL<6:4>	2G4	868	433	Comment
000	-	-	-	no band selected
001	-	-	active	mono-band
010	-	active	-	mono-band
011	-	active	active	dual-band
100	active	-	-	mono-band
101	active	-	active	dual-band
110	active	active	-	dual-band
111 (chip default)	active	active	active	tri-band

Table 13: Modes for multi-band operation

Register 0x25 TESTBUF_CTRL *

For internal use only. Write access is not recommended.

Register 0x26 IFAMP_GAIN_CTRL *

For amplifier adjustment. This register is set automatically. Write access not recommended.

Register 0x27 RX_ACTIVE_SELECT *

For test reasons. Write access is not recommended.

Register 0x28 CODE_SELECT

Bits 3..0 define the Code number<3:0> for the Code A sequence. MSBs are transmitted first.
Bits 7..4 define the Code number<3:0> for the Code B sequence. MSBs are transmitted first.

Code Number	Binary 32-Bit Sequence	Decimal Value	Name
0	0110 1010 1100 1101 0110 0111 0100 1111	1791846223	Code A
1	0110 1101 0011 1000 1001 0111 0111 0011	1832425331	Code B
2	0111 0010 0001 0011 0110 1010 0001 0110	1913874966	Code C
3	0000 1101 1110 1100 1001 0101 1100 0111	233608647	Code D
4	0001 0101 0011 0010 1001 1000 1011 0000	355637424	A inv
5	0001 0010 1100 0111 0110 1000 1000 1100	315058316	B inv
6	0100 0010 0101 1001 1111 0001 1011 1010	1113190842	mseqa
7	0100 0011 0010 0111 1101 1100 0101 0110	1126685782	mseqb
8	0000 0000 0000 0000 0000 0000 0000 0000		31 zeros
9	0111 1111 1000 0000 0000 0000 0000 0000		8 ones
10	0111 1111 1111 1111 1000 0000 0000 0000		16 ones
11	0111 1111 1111 1111 1111 1111 1000 0000		24 ones
12	0111 1111 1111 1111 1111 1111 1111 1111		31 ones
13	0101 0101 0101 0101 0101 0101 0101 0101		0101 pattern
14	0110 0110 0110 0110 0110 0110 0110 0110		1100 pattern
15	0111 0001 1100 0111 0001 1100 0111 0001		111000 pattern

Table 14: Predefined 32-bit sequences for binary correlators "A" and "B"

Register 0x29 KORREL_THRESH_A

Bits 4..0: KORREL_THRESH_A defines the correlator 'A' threshold. Useful values are between 0x19 and 0x1E. Lower values of KORREL_TRESH allow only few bit errors during reception of 32-bit code sequences.

Register 0x2A KORREL_THRESH_B

Bits 4..0: KORREL_THRESH_B defines the correlator 'B' threshold. Useful values are between 0x19 and 0x1E. Lower values of KORREL_TRESH allow only few bit errors during reception of 32-bit code sequences.

Register 0x2B KORREL_STATE

Bits 2..0 indicate the receiver branch that had contributed for the Code A recognition. "Weak" (bit 0), "medium" (bit 1), "strong" (bit 0). These bits can indicate the reception level of the last wake-up sequence "A" that triggered above KORREL_THRESH_A.

Bits 5..3 indicate the receiver branch that had contributed for the Code B recognition. "Weak" (bit 0), "medium" (bit 1), "strong" (bit 0). These bits can indicate the reception level of the last wake-up sequence "B" that triggered above KORREL_THRESH_B.

Bits 7..6 indicate the frequency band for latest correlator A or B trigger. "00" means 433-MHz band, "01" means 868-MHz band, "10" means 2.4-GHz band.

KORREL_STATE can be cleared by the user writing "1" into register 0x7C "KORREL_SV_CLEAR". Thus, it becomes clear when the latest update of KORREL_STATE and KORREL_VAL took place.

Register 0x2C KORREL_VAL

Bits 3..0 indicates the correlator match level for the Code A recognition. These bits indicate the number of bit errors of the last wake-up sequence "A". A value of 15 corresponds with no bit errors, a value of 0 corresponds with 15 bit errors. This register is only refreshed after a positive Code A match. Typically, KORREL_VAL is greater than 5.

Bits 7..4 indicates the correlator match level for the Code B recognition. These bits indicate the number of bit errors of the last wake-up sequence "B". A value of 15 corresponds no bit errors, a value of 0 corresponds with 15 bit errors. This register is only refreshed after a positive Code B match. Typically, KORREL_VAL is greater than 5.

KORREL_VAL can be cleared by the user writing "1" into register 0x7C "KORREL_SV_CLEAR". Thus, it becomes clear when the latest update of KORREL_STATE and KORREL_VAL took place.

Register 0x2D FDD_ENABLE

Enabling the fast data decoder (FDD) using bits 0..2, the RFicient receiver interprets Code A sequences in slow operating mode as initiator for FDD packets. During fast operating mode, the receiver interprets incoming fast Code A sequences as logical "0" and fast Code B sequences as logical "1". This ensures a bit error tolerant reception of data packets for a robust unidirectional data communication. Slow Code A or B sequences are ignored in fast mode, as well as Code A or B sequences in data rates other than the FAST data rate. Assuming NFA_SLOW = 101_2 and NFA_FAST = 000_2, the slow Code A sequence takes 31.25 ms and each fast Code A or B sequence last only 0.97656 ms. Thus, a 16-bit FDD telegram takes 62.50 ms. The FDD decoder is tolerant towards very short pauses between each code. Example for a 16-Bit telegram 0x725C ($0111\ 0010\ 0101\ 1100_2$) :

(slow) Code A (fast) ABBB AABA ABAB BBAA

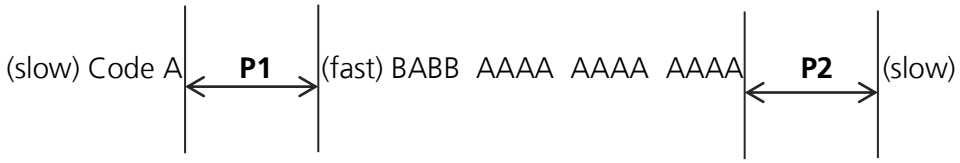
The FDD packet length is not limited. Incoming FDD bits are stored in the FIFO registers 0x2E..0x3F. The receiver switches back to slow mode after 1.526 ms (timeout; see timeout section below) if no valid fast Code A or B sequences are received. The user can force the receiver to switch back to slow mode by setting bit 7 in the registers 0x00, 0x01 or 0x02 respectively.

Timeout

The RFicient receiver automatically exits fast mode if no more fast bits are properly received or if no fast bit is decoded after slow Code A. Thus, the receiver switches back to slow mode and applies the low data rates (cf. registers 0x00<2:0>, 0x01<2:0>, 0x02<2:0>) again.

NFA_FAST	Fast Data Rate [Hz]	No Timeout		Safe Timeout	
		P1 max. [ms]	P2 max.	P1 min. [ms]	P2 min.
0	**32768 Hz**	5,371	1,465	5,432	1,526
1	**16384 Hz**	10,742	2,930	10,803	2,991
2	**8192 Hz**	21,484	5,859	21,545	5,920
3	**4096 Hz**	42,969	11,719	43,030	11,780
4	**2048 Hz**	85,938	23,438	85,999	23,499
5	**1024 Hz**	171,875	46,875	171,936	46,936
6	**512 Hz**	343,750	93,750	343,811	93,811
7	**256 Hz**	687,500	187,500	687,561	187,561

Table 15: Timeout values for various fast data rates



Register 0x2E FDD_ACTIVE

This register stores the FDD mode ("1": Fast Mode, "0": Slow Mode) for each frequency band: Bit 2: 433-MHz band, bit 1: 868-MHz band, bit 0: 2.4-GHz band. Only for read access.

Register 0x2F FO_QUIT

Writing ones into the FORCE_QUIT register, the user can force the receiver to quit fast mode immediately. This can be adjusted for each frequency band separately:
Bit 2: FO_QUIT_2G4: Setting this bit, the user can force the receiver to switch back to slow operation for the 2.4-GHz band.
Bit 1: FO_QUIT_868: Setting this bit, the user can force the receiver to switch back to slow operation for the 868-MHz band.
Bit 0: FO_QUIT_433: Setting this bit, the user can force the receiver to switch back to slow operation for the 433-MHz band.

Register 0x30 FDD_EXIT_COND

Bits 5..4: Indicates the 2-bit nibble for the 2.4-GHz band.
Bits 3..2: Indicates the 2-bit nibble for the 868-MHz band
Bits 1..0: Indicates the 2-bit nibble for the 433-MHz band.

2-bit FDD_EXIT_COND nibble	Reason	Comment
00	RX resetted or FDD is disabled	initial entry
01	timeout	no FAST Code A or B received
10	ID match failed	wrong 16-bit ID
11	FO_QUIT was set	user forced slow mode

Table 16: Exit conditions for Fast Data Decoding

Register 0x31 IRQ_SELECT

The user can select among IRQ event types 0..7 according to table 17. Bit 7 (MSB) corresponds to event type 7 (Cyclic Timer Alarm), bit 0 (LSB) to type 0 (ID match). If more event types are desired, more bits need to be selected. If any of the selected event types occur, an IRQ signal will be generated. This can be observed at the pins GPO1 or GPO2.

Event Type No.	Event Name	Remarks
0	ID match	16-Bit-ID (Reg. 0x35, 0x36: ID_HI & ID_LO) in FDD mode fits to received Fast Data Packet
1	FIFO overflow	indicates FIFO overflow if FDD data bits extend the specified FIFO length (Reg. 0x56): 16/24/32/40 bits.
2	FIFO buffer filled	indicates that FIFO is filled with FDD data bits according to the specified FIFO length (Reg. 0x56): 16/24/32/40 bits.
3	Code A or B detected	Correlation sequences A or B detected (no matter if FDD or Slow Mode is active)
4	ID match & FIFO filled	event type 0 occurred within the last <FIFO_length +2> bit cycles and event type 2 occurred after event type 0 ("telegram received")
5	ID match & SLOW	event type 0 occurred within the current FDD data reception and transition from FAST to SLOW mode occurred after event type 0 ("end of telegram")
6	RTC Timer Alarm	one of the 4 RTC timers (RTCSH0, RTCSH1, RTCLG0, RTCLG1) have reached the user-defined targets and caused alarm. Register RTC_EVENTS indicates which alarm rings.
7	Cyclic Timer Alarm	the cyclic timer has reached the user-defined target and caused alarm

Table 17: Available interrupt event types

Register 0x32 IRQ_STATUS

If any of the selected IRQ event types (reg. 0x31) occur, an IRQ signal will be generated. This can be observed at the pins GPO1 or GPO2. In register 0x32, the occurred IRQ event types 0..7 are marked with set bits according to table 17. Register 0x32 is a read-only register. It can be cleared by access to register 0x33.

Example:
If reg. 0x32 contains 0x49 (binary 0100 1001), the RTC Timer alarm has occurred, Code A or B have been detected and ID match was triggered.

Register 0x33 IRQ_CLR

When the user writes "1" bits into this register, the corresponding bit in reg. 0x32 will be cleared.

Example:
If the IRQ status register 0x32 contains the value 0x49, a write access to IRQ_CLR (reg. 0x33) writing 0x48 causes cleared bits in reg. 0x32:
The new content will be 0x01, as the LSB has not been cleared by register 0x33.

Note: Bits 4,2 and1 (FIFO_buffer_filled) cannot be cleared. To clear the IRQ_STATUS bits 4,2 and 1, the user must clear the FIFO buffer first by setting FIFO_COUNT to 0 (reg. 0x57, 0x58 or 0x59).

Register 0x34 IRQ_SET
By writing binary ones into this register, the user can set the corresponding status bit in the IRQ_STATUS register. Thus, IRQ events can be triggered for test purposes.

Example:
Writing 0x41 (binary 0100 0001) into reg. 0x34 causes set bits for event type 6 (RTC) and event type 0 (ID match) in register 0x32 IRQ_STATUS.

Register 0x35 ID_HI, Register 0x36 ID_LO
These two registers contain the 16-bit node ID. The HI byte comes first at adress 0x35. This is needed for individual wake-up (selective wake-up).

Register 0x37 IDM_ENABLE
Setting bits 2..0 in this register, the first 16 bits of the FDD data packet are interpreted as ID. Thus, the ID will not be handled as data and not be stored in the FIFO. FDD data bits that are sent after the ID will be stored in the FIFO (according to the frequency band).

IDM_ENABLE<2:0>			Frequency band with ID Match	Comment
<2>	<1>	<0>		
0	0	0	ID Match not activated	
1	X	X	active for 2.4-GHz band	
X	1	X	active for 868-MHz band	
X	X	1	active for 433-MHz band	
1	1	1	IDM for all bands active	default

Table 18: Selection of Frequency bands for ID Match

Bit 2: IDM_2G4 enables the ID match decoder for the 2.4-GHz band. Bit 1: IDM_868 enables the ID match decoder for the 868-MHz band. Bit 0: IDM_433 enables the ID match decoder for the 433-MHz band. If IDM_ENABLE is not set, no ID match IRQs will occur.

Register 0x38 IDM_CTRL
Bits 1..0 selects the ID match method for selective wake-up (see Table 19). The receiver can be addressed in three ways: individually (“00”), groupwise (“01” or “11”) or all nodes via broadcast call (“10” or “11”).

The **individual** 16-bit **ID** is set in registers 0x35 and 0x36. Chip default is 0x7DA8 (decimal 32168, binary 0111 1101 1010 1000). Thus, the according RF OOK telegram should look like:

(slow) A (fast) ABBB BBAB BABA BAAA

The most significant 4 bits of the group ID are derivated from the 4 MSBs of the individual 16-bit ID: GROUP_ID<15:12> := ID<15:12>.
The residual 12 bits of the group ID are zero: GROUP_ID<11:0> := 0.
Group calls are transmitted similar as a 16-bit ID telegram.

Example for group 0x0B (1011_2):
1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0

The RF OOK telegram should look like:
(slow) A (fast) BABB AAAA AAAA AAAA

The **16-bit broadcast ID** is 815_{10} (0x032F), 0000 0011 0010 1111_2 and fixed for all RFicient receivers. This can be used for network messages addressing all RX nodes. The broadcast ID cannot be changed. For this, the The RF OOK telegram should look like:

(slow) A (fast) AAAA AABB AABA BBBB.

IDM_CTRL<1:0>	ID Match Method	Comment
00	only individual 16-bit ID	
01	individual 16-bit ID or groupwise ID	
10	only broadcast ID	
11	individual 16-bit ID, groupwise ID, or broadcast ID	default

Table 19: Available Modes for selective wake-up

Register 0x39 IDM_CLR
Writing "1" into this register clears reg. 0x30 (FDD_EXIT_COND).

Register 0x3A IDM_BAND
This register indicates the frequency band for the latest ID Match event. Binary "00" means 433-MHz band, "01" means 868-MHz band, "10" means 2.4-GHz band.

Register 0x3B IDM_REASON
Bits 1..0 indicate the type of received selective wake-up message.

IDM_REASON<1:0>	Reason
00	-
01	individual ID
10	group ID
11	broadcast ID

Table 20: Types of incoming selective wake-up message

Register 0x3C RTC_SELECT

Each IC offers a 40-bit clock counter as a built-in system clock that comprises 1 year 23 days before overflow. Derived from that, four user-defined RTC timers are provided with a granularity of 30.517578 µs (i.e. 1 clock cycle):

Name	Function	Description
RTCSH0	16-Bit Short-term timer	Single Timer with a max. duration of 1.99996495 s (i.e. 65535 clock cycles)
RTCSH1	16-Bit Short-term timer	Single Timer with a max. duration of 1.99996495 s
RTCLG0	40-Bit Short-term timer	Single Timer with a max. duration of 33554432 s (i.e. 388 d 8h 40 min 32 s or 2^{40} -1 clock cycles)
RTCLG1	40-Bit Short-term timer	Single Timer with a max. duration of 33554432 s
CYCL	16-Bit cyclic timer	Provides periodic alarms out of the system clock divided by M, M can be 1 until 255. Thus, a maximum alarm period of 8 min 29.9922 s (i.e. 16711425 clock cycles) can be adjusted.

Table 21: Available RTC timers

The user can enable different built-in timers based on the 32768-Hz system clock in the register RTC_SELECT[4:0]. The RTC timers can generate interrupt events.

RTC_SELECT<4>	RTC_SELECT <3>	RTC_SELECT <2>	RTC_SELECT <1>	RTC_SELECT <0>
CYCLTOP_ENA	RTCLG1_ENA	RTCLG0_ENA	RTCSH1_ENA	RTCSH0_ENA
cyclic counter	long RTC counter #1	long RTC counter #0	short RTC counter #1	short RTC counter #0

Table 22: Selectable RTC timers

Register 0x3D RTC_STATUS

RTC_STATUS [3:0] indicates occured RTC events. Note that only enabled RTC timers can generate RTC events that may trigger IRQ6. As long as one event is active (corresponding bit in RTC_STATUS is set), IRQ6 cannot be cleared.

RTC_STATUS <3>	RTC_STATUS <2>	RTC_STATUS <1>	RTC_STATUS <0>
RTCLG1 event occured	RTCLG0 event occured	RTCSH1 event occured	RTCSH0 event occurred
long RTC counter #1	long RTC counter #0	short RTC counter #1	short RTC counter #0

Table 23: Observable occured RTC events

Register 0x3E RTC_CLR

Writing ones into register RTC_CLR[3:0], the user can clear event entries in RTC_STATUS[3:0].

RTC_CLR<3>	RTC_CLR <2>	RTC_CLR <1>	RTC_CLR <0>
clear RTCLG1 event	clear RTCLG0 event	clear RTCSH1 event	clear RTCSH0 event
long RTC counter #1	long RTC counter #0	short RTC counter #1	short RTC counter #0

Table 24: Clear RTC events

Register 0x3F RTCSH0_THRESH, Register 0x40 RTCSH0_THRESH

Upper byte RTCSH0_THRESH<15:8> is stored at address 0x3F, lower byte RTCSH0_THRESH<7:0> is stored at address 0x40. RTCSH0_THRESH defines the counter threshold when RTCSH0 will stop.

Register 0x41 RTCSH1_THRESH, Register 0x42 RTCSH1_THRESH
Upper byte RTCSH1_THRESH<15:8> is stored at address 0x41, lower byte RTCSH1_THRESH<7:0> is stored at address 0x42. RTCSH1_THRESH defines the counter threshold when RTCSH1 will stop.

Register 0x43 RTCLG0_THRESH, Register 0x44 RTCLG0_THRESH, Register 0x45 RTCLG0_THRESH, Register 0x46 RTCLG0_THRESH, Register 0x47 RTCLG0_THRESH:
RTCLG0_THRESH is a 40-bit register. Upper byte RTCLG0_THRESH<39:32> is stored at address 0x43, 2nd byte RTCLG0_THRESH<31:24> is stored at address 0x44. 3rd byte RTCLG0_THRESH<23:16> is stored at address 0x45. 4th byte RTCLG0_THRESH<15:8> is stored at address 0x46. 5th byte RTCLG0_THRESH<7:0> is stored at address 0x47. RTCLG0_THRESH defines the counter threshold when RTCLG0 will stop.

Register 0x48 RTCLG1_THRESH, Register 0x49 RTCLG1_THRESH, Register 0x4A RTCLG1_THRESH, Register 0x4B RTCLG1_THRESH, Register 0x4C RTCLG1_THRESH:
RTCLG1_THRESH is a 40-bit register. Upper byte RTCLG1_THRESH<39:32> is stored at address 0x48, 2nd byte RTCLG1_THRESH<31:24> is stored at address 0x49. 3rd byte RTCLG1_THRESH<23:16> is stored at address 0x4A. 4th byte RTCLG1_THRESH<15:8> is stored at address 0x4B. 5th byte RTCLG1_THRESH<7:0> is stored at address 0x4C. RTCLG1_THRESH defines the counter threshold when RTCLG1 will stop.

Register 0x4D CYCLPRESC
The chip provides periodic alarms out of the system clock (readable in CNTR40) divided by the CYCLPRESC prescaler M. M can be 1 until 255. Thus, a maximum alarm period of 8 min 29.9922 s (i.e. 16711425 clock cycles) can be adjusted.

Register 0x4E CYCLTOP, Register 0x4F CYCLTOP:
The value in register CYCLTOP<15:0> (adress 0x4E, 0x4F) defines the maximum cyclic counter value. Having reached this values, the cyclic counter CYCLCOUNT is automatically set to zero again. Thus, the 16-Bit counter CYCLCOUNT always runs from 0..CYCLTOP, then again from 0..CYCLTOP. Reaching CYCLTOP, IRQ event 7 can be triggered if properly enabled.

Register 0x50 CNTR40, Register 0x51 CNTR40, Register 0x52 CNTR40, Register 0x53 CNTR40, Register 0x54 CNTR40:
CNTR40 is a 40-bit register that represents the internal system clock counter (based on 32768 Hz) since last reset. This counter will overflow after 33554432 s (i.e. 1 a 23 d 8 h 40 min 32 s or 2^{40} -1 clock cycles). Thus, this built-in clock offers unique time stamps for one full year. Address 0x50 contains CNTR40<39:32>, address 0x51 contains CNTR40<31:24>, address 0x52 contains CNTR40<23:16>, address 0x53 contains CNTR40<15:8>, address 0x54 contains CNTR40<7:0>. This is read-only register.

Register 0x55 CNTR40_CLR
Writing „1“ into this register, the CNTR40 counter will be resetted. After this, a manual write access clearing this register is necessary.

Register 0x56 FIFO_LENGTH
Addressing register 0x56 FIFO_LENGTH<5:0>, the user can choose the length of the FIFO buffer for each frequency band.

Bits 1..0 define the length<1:0> of the FIFO buffers for incoming fast data bits in FDD mode for the 433-MHz frequency band. Default: 16 bits.

Bits 3..2 define the length<1:0> of the FIFO buffers for incoming fast data bits in FDD mode for the 868-MHz frequency band. Default: 24 bits.

Bits 5..4 define the length<1:0> of the FIFO buffers for incoming fast data bits in FDD mode for the 2.4-GHz frequency band. Default: 24 bits.

FIFO_LENGTH<1:0>	Length of FIFO buffer
00	16 bits
01	24 bits
10	32 bits
11	40 bits

Table 25: FIFO length settings

Register 0x57 FIFO_COUNT_433
Bits 5..0 indicate the number of received FDD bits in the FIFO buffer for the 433-MHz band. The user can write “0” in this register in order to clear the FIFO buffer.

Register 0x58 FIFO_COUNT_868
Bits 5..0 indicate the number of received FDD bits in the FIFO buffer for the 868-MHz band. The user can write “0” in this register in order to clear the FIFO buffer.

Register 0x59 FIFO_COUNT_2G4
Bits 5..0 indicate the number of received FDD bits in the FIFO buffer for the 2.4-GHz band. The user can write “0” in this register in order to clear the FIFO buffer.

Register 0x5A RX_FIFO_5_433, Register 0x5B RX_FIFO_4_433,
Register 0x5C RX_FIFO_3_433, Register 0x5D RX_FIFO_2_433,
Register 0x5E RX_FIFO_1_433, Register 0x5F RX_FIFO_0_433:
These 8-Bit registers form the FIFO buffer for the 433-MHz band. Register 0x5A is filled first.

Register 0x60 RX_FIFO_5_868, Register 0x61 RX_FIFO_4_868,
Register 0x62 RX_FIFO_3_868, Register 0x63 RX_FIFO_2_868,
Register 0x64 RX_FIFO_1_868, Register 0x65 RX_FIFO_0_868:
These 8-Bit registers form the FIFO buffer for the 868-MHz band. Register 0x60 is filled first.

Register 0x66 RX_FIFO_5_2G4, Register 0x67 RX_FIFO_4_2G4,
Register 0x68 RX_FIFO_3_2G4, Register 0x69 RX_FIFO_2_2G4,
Register 0x6A RX_FIFO_1_2G4, Register 0x6B RX_FIFO_0_2G4:
These 8-Bit registers form the FIFO buffer for the 2.4-GHz band. Register 0x66 is filled first.

Register 0x6C ACTUAL_NFA_433
This register contains the actual NFA value that is currently valid for 433-MHz data reception.

Register 0x6D ACTUAL_NFA_868
This register contains the actual NFA value that is currently valid for 868-MHz data reception.

Register 0x6E ACTUAL_NFA_2G4
This register contains the actual NFA value that is currently valid for 2.4-GHz data reception.

Register 0x6F ACTUAL_BANDSELECT
This 3-bit register contains the actual selected frequency bands that are currently used.
Bit 2: indicates 2.4-GHz operation, Bit 1: indicates 868-MHz operation, Bit 0: indicates 2.4-GHz operation.

Register 0x70 GENPURP_0
For internal use only. Write access is not recommended.

Register 0x71 GENPURP_1
8-Bit register free to use for read/write access.

Register 0x72 GENPURP_2
For internal use only. Write access is not recommended.

Register 0x73 XTAL_OSC_CTRL
1-Bit register that enables the built-in crystal oscillator driving a 2-pin crystal device.
Default: "1".

Register 0x74 XTAL_OSC_TRIM
Bit 3: Used for selecting an external system clock source that is applied to pin CLK32_XI. In this case, XTAL_OSC_TRIM<3> should bei "1". Thus, a 2-pin crystal will not be used.
Bits 2..0: Reserved, do not change!

Register 0x75 MUX_D_OUT_SEL
The RFicient® receiver offers numerous internal signals that can be selected and provided at two output pins GPO1 and GPO2. Writing register 0x75, the user can choose according to this table:

MUX_D_OUT_SEL	"GPO1" Pin 13	"GPO2" Pin 14	Description
0	FAST_CLK_2G4	FAST_DATA_2G4	RX data stream at 2.4 GHz
1	FAST_CLK_868	FAST_DATA_868	RX data stream at 868 MHz
2	FAST_CLK_433	FAST_DATA_433	RX data stream at 433 MHz
3	WUP_A_2G4	WUP_B_2G4	WakeUp signals at 2.4 GHz
4	KOMP_OUT_W	RX_ACTIVE	Test signals
5	KOMP_OUT_M	KOMP_OUT_S	Test signals
6	ID_MATCH	WUP_A_433	General ID match, WakeUp signal
7	WUP_A_868	WUP_B_868	WakeUp signals at 868 MHz
8	WUP_A_433	RX_ACTIVE	WakeUp signal, trigger signal
9	WUP_A_868	RX_ACTIVE	WakeUp signal, trigger signal
10	WUP_A_2G4	RX_ACTIVE	WakeUp signal, trigger signal
11	KOMP_OUT_M	RX_ACTIVE	Test signals
12	KOMP_OUT_S	RX_ACTIVE	Test signals
13	KOMP_OUT_W	WUP_A_868	Test signal, WakeUp signal
14	CLK32	IRQ_EVENT	System clock, Interrupt signal
15 (chip default)	IRQ_EVENT	CLK32	Interrupt signal, system clock

Table 26: Selection table for GPO1/GPO2 signals

Register 0x76 LC_TG_ENA *

For internal use only. Write access is not recommended.

Register 0x77 XTAL_GOOD

This is a 1-bit read-only register. It is "1" if the crystal oscillator has detect a stable system clock signal. Applying an external clock signal, this register content is not valid.

Register 0x78 COMP_THRESH_W

For internal use only. Write access is not recommended.

Register 0x79 COMP_THRESH_M

For internal use only. Write access is not recommended.

Register 0x7A COMP_THRESH_S

For internal use only. Write access is not recommended.

Register 0x7B OFFSETCAL_CTRL

For internal use only. Write access is not recommended.

Register 0x7C KORREL_SV_CLEAR

Writing „1" into this register, clears the KORREL_STATE and the KORREL_VAL registers.

Register 0x7D0, Register 0x7E0:

Not used.

Register 0x7F VERSION

This is a read-only 8-bit register, that contains a version byte. Currently: 0x41.

6 Package Outline Drawing

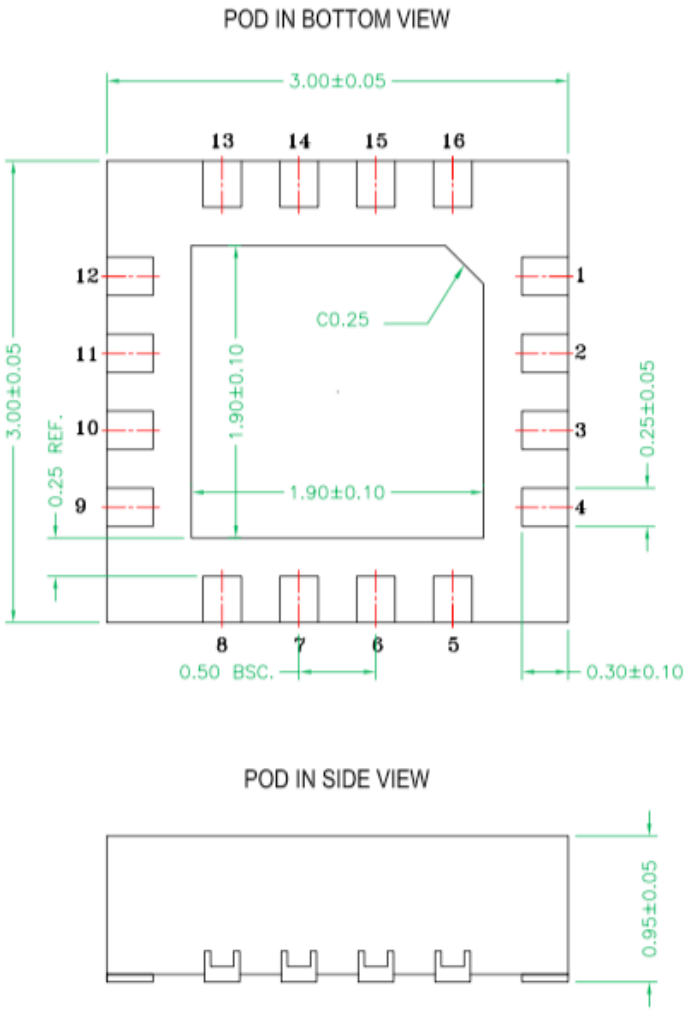


Figure 6: Package Outline Drawing (all dimensions in mm)

List of Tables