433/868 USB Dongle

RFPLAYER

API Specifications

V1.8

1/12/2016

Releases (Author : JP Gauthier):

V1.1 : 22/3/2016. Creation.

V1.2 : 22/6/2016. Release.

V1.3 : rename xml/json tag “value” to “v” .

V1.4 : add reserved fields in public API.

V1.5 : REPEATER ON/OFF management modification. Somfy RTS, Antennas, LED signalisation explanations. Screenshots examples. Based on Firmware V1.07

V1.6 : Add ALL_ON ALL_OFF CHACON & X10. PING instruction. Based on Firmware V1.08

V1.7 : Modification of HELLO command response. Based on Firmware V1.10

V1.8 : Modification of the end of HELLO command response (free area). Based on Firmware V1.12. Instruction INITLB added.

Summary

1 Introduction
2 USB Characteristics

The USB interface emulates a serial line with a classical chip FTDI FT232R.

Parameters : Baudrate : 115,2Kb/s, 8 bits data, no parity, 1 stop bit.

The host machine must be sensitive to flow control through the CTS line (Clear to Send), especially during firmware upgrade which handles a large amount of data.

NB: CTS line sensitivity is critical only during firmware download (update operation) which handles a large amount of data. At this time, the dongle is processing this incoming flow in real time so that CTS line handling isn't mandatory. CTS line handling could be seen as 'provision'.

The USB Dongle is not sensitive to RTS line (Request to Send). The host machine has to receive the incoming flow anyway.

Incoming data are memorized in a queue (FIFO) of 2048 bytes so that many independent commands can be enqueued before processing. CTS line becomes false when this queue become almost full.

Max interpreter command line buffer is 1500 bytes length so that many instructions can be specified in one command line.

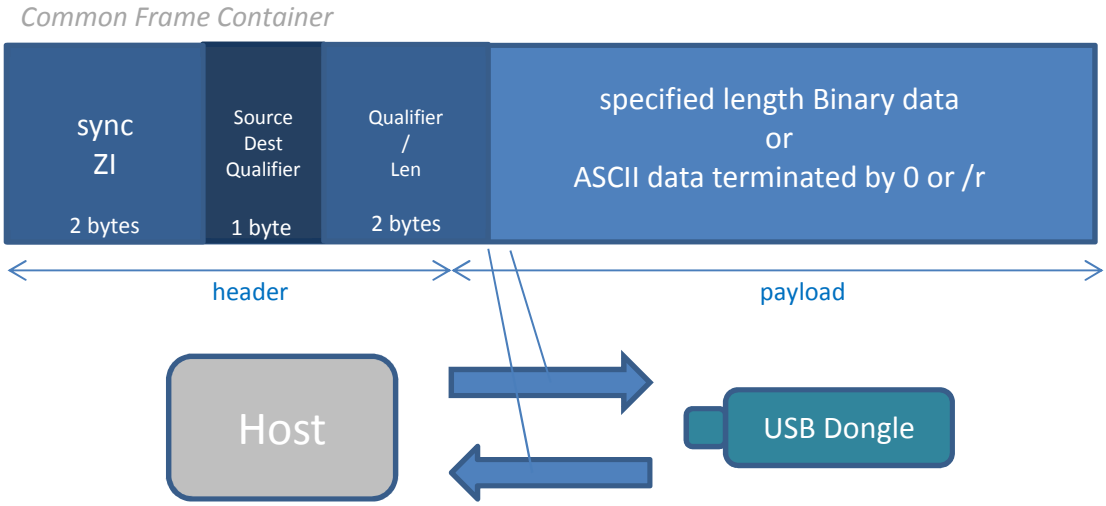
3 Host App

A Java App utility with friendly interface running an PC/Mac is available from our website to do dongle configuration. Il could be useful to understand how the present API works.

4 Common Frame Container

The common Frame container is bi-directional and applicable to:

- *Host* to *USB Dongle* transfers
- *USB Dongle* to *Host* transfers



The role of the Frame Container is to help de the destination to synchronize incoming flow, and multiplex or demultiplex the frame upon direction and the SourceDestQualifier field.

The frame payload is in binary or ASCII data form.

Binary form is used by the RF entity itself to handle the associated RF protocol. The Frame Container is agnostic to the payload field. The specifications of the payload field is RF entity dependent and sometimes completely specified by the provider of the chip handling this RF entity (e.g. Z-WAVE / ENOCEAN)

ASCII form is used to manage the RF entities for configuration, options setting and update processing. “AT commands style” is chosen and a commands interpreter permits to launch commands e.g. “ZIA++STATUS”.

4.1 ASCII data (used by commands Interface)

Nom	Size	Contains	Remark
Sync1	1	'Z'	0x5A
Sync2	1	'I'	0x49
SourceDestQualifier	1	'A' to 'O'	0x41 to 0x4F
Qualifier	2		Printable characters Defining the frame subtype
AsciiData[0…n]	0…infinite unsigned char[]		ASCII data
terminator	1	\0 or \r	0x0 or 0x0D

4.2 Binary data (used by HA protocol frames)

Name	Size	type	Contains	Remark
Sync1	1	unsigned char	'Z'	0x5A
Sync2	1	unsigned char	'I'	0x49
SourceDestQualifier	1	unsigned char	0x00 to 0x0A	Not printable characters
Len	2	unsigned char		Len of BinaryData [] LSB first
BinaryData []		unsigned char[]		Binary data

4.3 SourceDestQualifier

bit	7	6	5	4	3	2	1	0
	0 *reserved*	*Value = x04 for ASCII DATA* *Value = 0x0 for BINARY DATA*			SourceDest			

When the Host emits the frame, SourceDest contains the ID of the destination of the frame (e.g RF entity = 1 for 433/868).

When the USB Dongle emits the frame to Host, SourceDest contains the ID of the source transmitting the frame (e.g RF entity = 1 for 433/868), thus, Host as destination is implicit.

4.4 SourceDest

SourceDest/ RF entity	Value	Remark
MUX Management	0	
433/868	1	ZIA
Future RF entities	Other values	ZIB, ZIC, etc…

NOTA : It is expected that future USB dongles will handle more than one RF entity.

Several flows to/from several RF entities can be multiplexed on the same physical link with ZIx header.

5 API 433/868

The 433/868Mhz interface is able to emit/receive the protocols listed below.

All received protocols are decoded simultaneously without mutual exclusion on both 433 and 868Mhz bands. One hardware receiver is used for each band and the digital processing still works simultaneously on both bands.

Low band 'L' = 433 Mhz.

High band 'H' = 868 Mhz.

The receiver of each band is tuned for a specific frequency. Consequently, receivers must be adjusted (by the management command 'FREQ') on:

- 433.420 Mhz used by Somfy RTS or 433.920 Mhz used by a lot of devices.
- 868.350 Mhz used by Deltadore X2D or 868.950 Mhz used by Visonic.

However enlarge the spectrum and thus receiving several frequencies in the same band is allowed by decreasing the selectivity of the receiver by the management command 'SELECTIVITY H/L 1'.

Emissions are not affected by frequencies differences because the transmitting frequency is dynamically adjusted frame by frame upon protocols.

Brand	Protocol	Frequency Mhz	Transmitting Capability (USB RF gateway)	Receiving Capability (USB RF gateway)	Repeating Capability	Transcoding Capability: *Receiving Capability* ➔ *Transmitting Capability*
VISONIC	PowerCode SecureCode	433.920	Yes PowerCode only	Yes PowerCode &SecureCode	Yes PowerCode &SecureCode	Yes PowerCode &SecureCode(in)
VISONIC	PowerCode SecureCode	868.950				
CHACON/DIO	CHACON V2	433.920	Yes	Yes	Yes	Yes
DOMIA	DOMIA/CHACON V1	433.920	Yes	Yes	Yes	Yes
X10	X10	433.920	Yes	Yes	Yes	Yes
DELTADORE	X2D	433.920	Yes	Yes	Yes	Yes
DELTADORE	X2D	868.350				
SOMFY	RTS	433.420	Yes	Yes	Yes	Yes
BLYSS (433Mhz)	BLYSS/AVIDSEN433	433.920	Yes	Yes	Yes	Yes
KD101	KD101	433.920	Yes	Yes	Yes	Yes
PARROT (Learn & Play)	-	433/868Mhz	Yes	Yes	Yes	Yes
Scientific Oregon probes	OREGON V1	433.920	No	Yes	Yes	No
Scientific Oregon probes	OREGON V2.1	433.920	No	Yes	Yes	No
Scientific Oregon probes	OREGON V3.0	433.920	No	Yes	Yes	No
OWL probes	OWL	433.920	No	Yes	Yes	No

Trademarks

X10 is a trademark of X10 Company.

VISONIC, Powercode and SecureCode are trademarks of Visonic Company.

SOMFY and RTS are trademarks of Somfy Company.

DELTADORE and X2D are trademarks of Deltadore Company.

BLYSS is a trademark of KingFisher Company.

CHACON is a trademark of Chacon Company

OREGON is a trademark of Scientific Oregon Company.

5.1 Host → USB Dongle (configuration and transmitting data to RF)

5.1.1 ASCII data (ordinary used by commands Interface)

AsciiData[0…n]	Size	Remark
Local management	0…infinite	Upon 433/868 management commands interpreter

Preamble : Due to its ASCII nature and during debug, management commands (including the container header) can be entered by a PC utility eg. *Teraterm* when the Dongle is plugged on a PC. Terminator is \r or \0. Dual or over terminators e.g . "\r\n" are permitted because the first terminator is taken into account, the following are filtered because the receiver automata is looking for "ZI" header and strips others incoming characters. Replace "\r" by "\r\n" in the incoming flow for a best viewing. The incoming flow is sensitive to backspace character (\010) but excluding the header "ZIA++" (Useful during debug if commands are sent by a keyboard).

5.1.1.1 Commands

AsciiData[0…n] are analyzed by an interpreter of commands (see the list below). This interpreter has an input buffer of more 1000 characters. A command number can precede the entire line; this number is repeated in the answer.

The commands interpreter is insensitive to upper and lowercases. "ZIA++" header is sensitive.

One time "ZIA++" is received, the RF activity is stopped. The line remaining must be entered.

The format of the answer (TEXT, XML, JSON) can be specified after the command number.

STATUS and HELLO are the single commands which returns a response. Other commands do not return answer.

Several commands can be present in a single frame with a point (‘.’) acting as command separator.

The management commands are able to :

- Configure the device,
- Send RF orders as ON/OFF/DIM etc… (Exactly as binary requests but with a more friendly way).

Command name	parameters	Explanations
FREQ	H/L val H: val = 0 or 868950 or 868350 L: val = 0 or 433420 or 433920	Modify the receiver frequency on High band (around 868Mhz) or Low band (around 433Mhz). Low and high band receivers work simultaneously. H band : Set the high frequency receiver to Off, 868.950 or 868.350Mhz Default is 868.950Mhz L band : Set the Low frequency receiver to Off or 433.420 or 433.920 Mhz Default is 433.920 Mhz Frequency Value of 0 leads to shutdown the selected receiver including the transmitter of the specified band. Examples : FREQ H 868950 FREQ L 433920 FREQ H 0 Specified frequency is saved during shutdown.

SELECTIVITY	H/L val	Modify the high frequency receiver selectivity on the selected band (L:433/H:868Mhz). Selectivity is the ability to filter out of band signals. NOTE: Higher selectivity (low value in term of Khz) means higher RF receiver sensitivity, but out of frequency transmitting appliances could be discarded. Lower selectivity (high value in term of Khz) means lower RF receiver sensitivity (lower performance). Old cheap devices can have a large frequency offset or shift over time especially when outdoor used. Val : 0 : Default value - Medium selectivity (300Khz) 1 : Very low selectivity (800Khz), frequency centered between used frequencies (433420-433920 or 868350-868950) 2 : Very low selectivity (800Khz) 3 : Low selectivity (500Khz) 4 : Medium selectivity (300Khz) 5 High selectivity (200Khz) Default : 0 Examples : SELECTIVITY H 3 SELECTIVITY L 1 Specified selectivity is saved during shutdown.
SENSITIVITY	H/L val	Radio Frequency receiver sensitivity on the selected band (L:433/H:868Mhz) (Ultra-High Frequency analog antenna sensitivity) Val : 0 : Default value - High sensitivity (-0dB) 1: Very low sensitivity (-18dB) 2 : low sensitivity (-12dB) 3 : medium sensitivity (-6dB) 4 : high sensitivity (-0dB) – Default value Default : 0 Example : SENSITIVITY L 1 // set very low sensitivity on 433Mhz Decreasing RF sensitivity could be only useful in specific cases during limited time, as pairing procedure or RF sequence learning with isolation from far RF transmitters. Specified sensitivity is NOT saved during shutdown.
DSPTRIGGER	H/L val	Digital Signal Processing trigger on the selected band (L:433/H:868Mhz) Define the smallest signal amplitude leading to start frame detection and analysis. Low trigger value means high sensitivity. Too big trigger value leads to forget useful frames. Too low trigger value leads to detect ghostly frames, generated by floor noise, and sometimes forget useful frames during this time. Val : 4 to 20. Unit : dBm. Val = 0 or out of bounds value lead to come back to the default value.

		Default values : 433 Mhz = 8dBm 868Mhz = 6dBm Example : DSPTRIGGER L 15 Specified DSPTRIGGER value is saved during shutdown.
RFLINKTRIGGER	H/L val	RFLINK trigger on the selected band (L:433/H:868Mhz) Define the smallest signal amplitude leading to start frame detection and analysis. Low trigger value means high sensitivity. Too big trigger value leads to forget useful frames. Too low trigger value leads to detect ghostly frames, generated by floor noise, and sometimes forget useful frames during this time. Val : 4 to 20. Unit : dBm. Val = 0 or out of bounds value lead to come back to the default value. Default values : 433 Mhz = 12dBm 868Mhz = 10dBm Example : RFLINKTRIGGER L 15 Specified RFLINKTRIGGER value is saved during shutdown.
LBT	val	*Listen Before Talk* function Val : 6 to 30. Unit : dBm Out of bounds value of val leads to come back to the default value. Val = 0 inhibits LBT function. Default is LBT enabled (very highly recommended). When enabled, the transmitter will “listen” the current activity on the same frequency and wait a silence before to “talk”. Sent frames cannot be delayed more than 3 seconds. Default value : 16dBm Example : LBT 10 Specified LBT value is saved during shutdown.
SETMAC	mac	Set the interface MAC address to unsigned long decimal 32 bits value. Warning! Changing the MAC address is without effect on incoming RF Frames. But changing the MAC address will modify the ID contained in outcoming RF Frames on most protocols. Pairing between the dongle and actuators could be broken. Each Dongle owns a different MAC address configured at the factory. Examples : SETMAC 123456765 SETMAC 0x2AB265C3 Specified MAC value is saved during shutdown.

FACTORYRESET	[ALL]	Restore factory default parameters. Include “REMAPPING PARROT” command. PARROT records and TRANSCODER configuration are not affected. ALL optional parameter erases all, including PARROT records and TRANSCODER configuration. Examples : FACTORYRESET FACTORYRESET ALL Restored values are saved during shutdown.
STATUS	[SYSTEM, RADIO, TRANSCODER, PARROT] [TEXT, XML, JSON]	Gives the status of the device for a specific item : SYSTEM, RADIO, TRANSCODER, PARROT. The status can be given with several formats : TEXT, XML, JSON. (default : TEXT) Examples STATUS STATUS SYSTEM XML STATUS RADIO JSON STATUS TRANSCODER JSON // list all entries STATUS TRANSCODER ENTRY 5 JSON // list one entry STATUS PARROT XML // list all records STATUS PARROT B1 ON XML // list one record See specific paragraph below with examples of this command.
FORMAT	OFF BINARY HEXA HEXA FIXED TEXT XML JSON	Gives the format of the received RF Frames sent to USB port. FORMAT OFF shutdowns incoming RF frames to USB port. FORMAT command doesn’t give immediate responses, but asynchronous messages, one at each received RF frame. By default, format is OFF. After the Dongle detection, the home automation BOX has to set the chosen format with this command. Typically, an home automation BOX sends at startup: FORMAT BIN or FORMAT HEX or FORMAT XML or FORMAT JSON Etc… NOTE: Shutdown the repeater during GATEWAY usage is recommended because the time spent to repeate frames is lost for receive frames during this time. Thus the instruction FORMAT automatically disables the REPEATER function with an implicit “REPEATER OFF”. The REPEATER function can be re-enabled with “REPEATER ON”. Example : FORMAT XML See Specific paragraph below with examples of this command.

	RFLINK OFF RFLINK BINARY	FORMAT RFLINK OFF disables the RFLINK flow FORMAT RFLINK BINARY enables RFLINK flow. Due to the RFLINK records length, RFLINK flow can be delivered only in binary form. NOTE : Frames with RFLINK format are delivered to the host machine only when the internal engine ("One Step Decoder") is unable to decode the frame. Example : FORMAT RFLINK BINARY Specified FORMAT is NOT saved during shutdown.
HELLO		Returns the text string "Welcome to Ziblue Dongle xxxxxxxxxxxx". The beginning of the returned string is usable to recognize the Dongle at the startup of host application. xxxxxxxxxxxx is variable and MUST not be matched. Not sensitive to XML / JSON formats specification. Example : ZIA++HELLO ZIA-- Welcome to Ziblue Dongle RFPLAYER (RFP1000, Firmware V1.12 Mac 0xF6C09FA1)!
PING		Returns the text string "PONG". Could be useful to check periodically the alive state of the dongle. The command Hello can be used for the feature too. Not sensitive to XML / JSON formats specification. Example : PING
ON OFF DIM ASSOC DISSOC ASSOC_OFF DISSOC_OFF Only X10 & CHACON: ALL_ON ALL_OFF	ID *x* or pseudo-address X10 form Protocol chosen in the list : VISONIC433 VISONIC868 CHACON DOMIA X10 X2D433 X2D868 X2DSHUTTER X2DELEC X2DGAS RTS BLYSS PARROT KD101 % *x* BURST *x* QUALIFIER *x*	**Send an order over RF** (ASCII form). This request exists in binary form. (see next chapters). ID *x* or pseudo-address X10 form : address of the RF appliance. ID x : with x between 0 and 255 included. pseudo-address X10 form : equivalent to IDx but with more friendly form. A—P 1—16. ID 0 = A1 ID 15 = A16 … ID 255 = P16 % x : parameter of DIM command. Light set at x % BURST x : define a frame repetition factor. 0 is default. QUALIFIER : define an optional parameter eg shutter (0) or portal (1) for Somfy RTS. Mandatory : - ID *x* or pseudo-address X10 form - Protocol NOTE : Parameters can be given in any order. The Host can request ASSOC (pairing dongle<->appliance) through this command.

		ASSOC_OFF is useful to fill one entry 'OFF' of PARROT (ASSOC fills "ON' OF PARROT). In this way, a user reminder can not be specified. The command PARROTLEARN is another way to fill ON or OFF entry of PARROT but with a user reminder. Examples : ON A3 RTS QUALIFIER 1 DIM ID 12 CHACON %40 ON KD101 ID 90500 ASSOC X2D868 F7
RECEIVER	Operators : + - Protocol chosen in the list : * X10 RTS VISONIC BLYSS CHACON OREGONV1 OREGONV2 OREGONV3/OWL DOMIA X2D KD101 PARROT	Specify the list of enabled received protocols The operators specifies if the subsequent protocols are to add or remove. Parameters are read from left to right. Examples : RECEIVER + CHACON BLYSS // add CHACON and BLYSS to the current list RECEIVER - * + CHACON // receive only chacon RECEIVER + * - CHACON // remove CHACON from the current list Specified list is saved during shutdown.
REPEATER	Operators : + - Protocol chosen in the list : * X10 RTS VISONIC BLYSS CHACON OREGONV1 OREGONV2 OREGONV3/OWL DOMIA X2D KD101 _____ ON OFF	Specify the list of enabled repeated protocols The operators specifies if the subsequent protocols are to add (+) or remove (-). Parameters are read from left to right. Examples : REPEATER + CHACON BLYSS // add CHACON and BLYSS to the current list REPEATER - * + CHACON // remove ALL but repeate only CHACON REPEATER + * - CHACON // repeate ALL but remove CHACON from the current list Specified list is saved during shutdown. _____ ON : Temporary enables the repeater function. OFF : Temporary disables the repeater function. "REPEATER ON" / "REPEATER OFF" commands are NOT saved during shutdown. It is a fast way to enable/disable REPEATER without saving. After startup, "REPEATER ON" is the default state. NB: The instruction "FORMAT xxxx" executes automatically "REPEATER OFF" because when the dongle is connected to an HA BOX, the REPEATER is generally OFF to maximize receiving time. "REPEATER ON" permits to re-enable the REPEATER with an HA BOX.
LEDACTIVITY	0 / 1	Enable ou disable LED activity related to RF flow. LEDACTIVITY 0 disables LED activity LEDACTIVITY 1 enables LED activity

		Default is 1 after factory RESET. LED activity cannot be disabled during PARROTLEARN and TRANCODER processings or when the dongle signals an error.
INITLB		(re-) Initialization of the leaky buckets (LB) to full level. Each frequency owns a leaky bucket. At each frame sent, the leaky bucket level decreases, but idle time increases this level. Frames are not send and discarded when LB level reaches 0. The LB defines a maximum duty cycle per hour defined by European laws (e.g. 433Mhz: 10%, 868.350Mhz: 1%, 868.950Mhz: 0.1%) where the channel is usable to send frames. INTLB could be useful during tests to disable LB limitation of bandwidth. Example : INITLB. ON X10 A1
PARROTLEARN	ID *x* or pseudo-address X10 form ON or OFF [*reminder*]	Specify to PARROT learn a frame. Dongle RF sensitivity decreases during this processing. The dongle then enters into 'capture mode'. Place the transmitter to learn at 2-3m et force it to emit frames. Decrease gradually the distance if no effect. Too near device will give bad results. The frame must be captured 2 times (1: Low frequency blue blinking then 2: High frequency blue blinking ; Good comparison : PINK Lighting, bad comparison : RED lighting). Without success This processing is automatically stopped : - After around 2mn (upon your tries). - When the lateral button of the dongle is pushed ID *x* or pseudo-address X10 form : address of the RF appliance. ID x : with x between 0 and 239 included. pseudo-address X10 form : equivalent to IDx but with more friendly form. A—O 1—16. P1 to P16 are reserved. A1…B16 are sensitive entries and can be recognized in real-time when the associated frames are send later by the device. C1…O16 can be played on RF but not recognized in the incoming RF flow. Optional parameter ON or OFF specify if the frame means one ON or one OFF. Default is ON. A reminder can be specified between brackets [ ]. Max Length is 30 characters. Reminder is a text line chosen by the user to remember which device has been captured. Reminder will be displayed later with "STATUS PARROT" command Examples : PARROTLEARN A3 [sensor bedroom1] PARROTLEARN ID 17 ON [Gate open sensor] PARROTLEARN ID 17 OFF [Gate close sensor] PARROTLEARN B2 ON NOTE : Specify the chosen frequency with FREQ before to use this command. This frequency is memorized with the frame record and will be automatically used when the frame will be played.

		PARROTLEARN results are saved during shutdown.
REMAPPING	PARROT ONOFF < Send an order over RF command parameters> _____ CLEAR	REMAPPING is a transcoder shortcut command to remap all PARROT RF sensitive entries (A1…B16) to another unique protocol. It is useful when REPEATER function is enabled. ON or OFF PARROT attributes is reported to the output protocol. Examples : REMAPPING PARROT ONOFF RTS D1 (remap A1…B16 PARROT ENTRY to D1…E16 on RTS protocol (shutter) REMAPPING PARROT ONOFF RTS D1 QUALIFIER 1 (remap A1…B16 PARROT ENTRY to D1…E16 on RTS protocol (portal) NOTE : TRANSCODER command has a finest granularity, entry by entry. TRANSCODER has priority over REMAPPING command if transcoding conflicts appear. _____ CLEAR removes PARROT REMAPPING Exemple : REMAPPING PARROT CLEAR Specified REMAPPING PARROT ONOFF or CLEAR results are saved during shutdown.
TRANSCODER	<Source> To <Destination> [*reminder*]	TRANSCODER command permits to transcode an incoming RF frame to another protocol on one outcoming RF FRAME. TRANSCODER is a sub-function of the REPEATER. There are 32 entries. Incoming action (ON or OFF) are reported as outcoming action on X10, DOMIA, CHACON, RTS, PARROT protocols, so that 64 different frames can be generated. Other protocols are strictly transcoded by destination parameters. In operation, frame parameters are read in real time in the incoming RF flow and compared to TRANSCODER <source> parameters. If they match, the frame is transcoded with <destination> parameters. <source> Source can be : - Expressed by a line of parameters - Captured in real time from the incoming RF flow (by "CAPTURE") - Retrieved from the already memorized parameters (by "KEEP") *Expressed by a line of parameters :* - ENTRY *x* with *x* between 0 and 31 included. - Protocol chosen in the list : X10, VISONIC, BLYSS, CHACON, OREGON, DOMIA, OWL, X2D, RFY, KD101, PARROT - ID *x* or X10 form : Address of the RF transmitter (most time a 32 bits ID) - Optional SUBTYPE x with x protocol dependent (default : 0) - Optional QUALIFIER x with x protocol dependent (default : 0) *Captured in real time from the incoming RF flow :* 'CAPTURE' or 'CAPTURE Protocol' to filter undesirable protocols

		When CAPTURE is specified, the dongle enters into 'capture mode' with blue LED blinking during 1 mn. Push the lateral button of the dongle to break the processing. The frame must be captured twice times (1: Low frequency blinking then 2: High frequency blinking ; Good matching: PINK Lighting, bad matching: RED lighting). NOTE: Transcoding starts immediately after if REPEATER is enabled (including current frame), so a short RED lighting (transmitting signal) could appear after PINK lighting. Retrieved from the already memorized parameters : 'KEEP' <destination> ON or OFF or DIM actions followed by "Send an order over RF" parameters. Examples : TRANSCODER ENTRY 23 X10 D8 TO ON RTS B7 TRANSCODER ENTRY 23 CHACON ID 435432766 TO ON RTS B7 TRANSCODER ENTRY 23 CHACON CAPTURE TO ON RTS B7 TRANSCODER ENTRY 23 KEEP TO ON RTS ID 12 QUALIFIER 1 [*reminder*] A reminder can be specified between brackets [ ]. Max Length is 30 characters. Example: TRANSCODER ENTRY 23 X10 D8 TO ON RTS B7 [My Comment] Reminder is a text line chosen by the user to remember which device has been captured. Reminder will be displayed later with "STATUS TRANSCODER" command Examples : TRANSCODER ENTRY 23 X10 D8 TO ON RTS B7 [My reminder here]
	_____ CLEAR	_____ CLEAR argument permits to clear all TRANSCODER entries by "TRANSCODER CLEAR" or a specific entry "TRANSCODER ENTRY *n* CLEAR" Specified TRANSCODER command results are saved during shutdown.

5.1.1.1.1 STATUS answer examples

The line “ZIA++1234 STATUS SYSTEM” will return :

ZIA--systemStatus request number=1234

Version: 1.01, Mac: 0xF2FC7AF4, LBT: 16dBm, Factory Test: 1464773430, RT denials: 0,

transmitter available: VISONIC433 VISONIC868 CHACON DOMIA X10 X2D433 X2D868 X2DSHUTTER X2DELEC X2DGAS RTS BLYSS PARROT KD101

receiver available: X10 RTS VISONIC BLYSS CHACON OREGONV1 OREGONV2 OREGONV3/OWL DOMIA X2D KD101 PARROT

receiver enabled: X10 RTS VISONIC BLYSS CHACON OREGONV1 OREGONV2 OREGONV3/OWL DOMIA X2D KD101 PARROT

repeater available: X10 RTS VISONIC BLYSS CHACON OREGONV1 OREGONV2 OREGONV3/OWL DOMIA X2D KD101 PARROT

repeater enabled: X10 RTS VISONIC BLYSS CHACON OREGONV1 OREGONV2 OREGONV3/OWL DOMIA X2D KD101 PARROT

The line “ZIA++1234 STATUS SYSTEM XML” will return :

ZIA--<?xml version="1.0" encoding="ISO8859-1" ?><systemStatus><reqNum>1234</reqNum> <n>Version</n> <v>1.01</v> <unit></unit> <c></c> <n>Mac</n> <v>0xF2FC7AF4</v> <unit></unit> <c></c> <n>LBT</n> <v>16</v> <unit>dBm</unit> <c></c> <n>Factory Test</n> <v>1464773430</v> <unit></unit> <c></c> <n>RT denials</n> <v>0</v> <unit></unit> <c></c><transmitter><available> VISONIC433 VISONIC868 CHACON DOMIA X10 X2D433 X2D868 X2DSHUTTER X2DELEC X2DGAS RTS BLYSS PARROT KD101 </available></transmitter><receiver><available> X10 RTS VISONIC BLYSS CHACON OREGONV1 OREGONV2 OREGONV3/OWL DOMIA X2D KD101 PARROT </available></receiver><receiver><enabled> X10 RTS VISONIC BLYSS CHACON OREGONV1 OREGONV2 OREGONV3/OWL DOMIA X2D KD101 PARROT </enabled></receiver><repeater><available> X10 RTS VISONIC BLYSS CHACON OREGONV1 OREGONV2 OREGONV3/OWL DOMIA X2D KD101 PARROT </available></repeater><repeater><enabled> X10 RTS VISONIC BLYSS CHACON OREGONV1 OREGONV2 OREGONV3/OWL DOMIA X2D KD101 PARROT </enabled></repeater></systemStatus>

The line “ZIA++1234 STATUS SYSTEM JSON” will return :

ZIA--{"systemStatus": {"reqNum": "1234", "info": [{"n" : "Version", "v" : "1.01", "unit" : "", "c" : ""}, {"n" : "Mac", "v" : "0xF2FC7AF4", "unit" : "", "c" : ""}, {"n" : "LBT", "v" : "16", "unit" : "dBm", "c" : ""}, {"n" : "Factory Test", "v" : "1464773430", "unit" : "", "c" : ""}, {"n" : "RT denials", "v" : "0", "unit" : "", "c" : ""}, {"transmitter": {"available": { "p": ["VISONIC433", "VISONIC868", "CHACON", "DOMIA", "X10", "X2D433", "X2D868", "X2DSHUTTER", "X2DELEC", "X2DGAS", "RTS", "BLYSS", "PARROT", "KD101"]}}}, {"receiver": {"available": { "p": ["X10", "RTS", "VISONIC", "BLYSS", "CHACON", "OREGONV1", "OREGONV2", "OREGONV3/OWL", "DOMIA", "X2D", "KD101", "PARROT"]}}}, {"receiver": {"enabled": { "p": ["X10", "RTS", "VISONIC", "BLYSS", "CHACON", "OREGONV1", "OREGONV2", "OREGONV3/OWL", "DOMIA", "X2D", "KD101", "PARROT"]}}}, {"repeater": {"available": { "p": ["X10", "RTS", "VISONIC", "BLYSS", "CHACON", "OREGONV1", "OREGONV2", "OREGONV3/OWL", "DOMIA", "X2D", "KD101", "PARROT"]}}}, {"repeater": {"enabled": { "p": ["X10", "RTS", "VISONIC", "BLYSS", "CHACON", "OREGONV1", "OREGONV2", "OREGONV3/OWL", "DOMIA", "X2D", "KD101", "PARROT"]}}}]}}

The line “ZIA++1234 STATUS RADIO” will return :

ZIA--

radioStatus request number=1234

Frequency: 433920Khz Adapted to most 433Mhz devices,

Selectivity: 0 Default value, FloorNoise: -97dBm A bit noisy, DspTrigger: 8dBm, RFlink: 1 Enabled, RFlinkTrigger: 12dBm, sentFrames: 272, discardedFrames: 0, authorizedUsage: 3600000ms/h ETSI EN 300 220-1, remainingUsage: 3599906ms,

Frequency: 868950Khz Adapted to Visonic/Meian devices,

Selectivity: 0 Default value, FloorNoise: -108dBm Very small noise, DspTrigger: 6dBm, RFlink: 1 Enabled, RFlinkTrigger: 10dBm, sentFrames: 0, discardedFrames: 0, authorizedUsage: 3600ms/h ETSI EN 300 220-1, remainingUsage: 3600ms,

The line “ZIA++1234 STATUS RADIO XML” will return :

ZIA--<?xml version="1.0" encoding="ISO8859-1" ?><radioStatus><reqNum>1234</reqNum><band> <n>Frequency</n> <v>433920</v> <unit>Khz</unit> <c> Adapted to most 433Mhz devices</c> <n>Selectivity</n> <v>0</v> <unit></unit> <c> Default value</c> <n>FloorNoise</n> <v>-97</v> <unit>dBm</unit> <c> A bit noisy</c> <n>DspTrigger</n> <v>8</v> <unit>dBm</unit> <c></c> <n>RFlink</n> <v>1</v> <unit></unit> <c> Enabled</c> <n>RFlinkTrigger</n> <v>12</v> <unit>dBm</unit> <c></c> <n>sentFrames</n> <v>272</v> <unit></unit> <c></c> <n>discardedFrames</n> <v>0</v> <unit></unit> <c></c> <n>authorizedUsage</n> <v>3600000</v> <unit>ms/h</unit> <c> ETSI EN 300 220-1</c> <n>remainingUsage</n> <v>3599906</v> <unit>ms</unit> <c></c></band><band> <n>Frequency</n> <v>868950</v> <unit>Khz</unit> <c> Adapted to Visonic/Meian devices</c> <n>Selectivity</n> <v>0</v> <unit></unit> <c> Default value</c> <n>FloorNoise</n> <v>-108</v> <unit>dBm</unit> <c> Very small noise</c> <n>DspTrigger</n> <v>6</v> <unit>dBm</unit> <c></c> <n>RFlink</n> <v>1</v> <unit></unit> <c> Enabled</c> <n>RFlinkTrigger</n> <v>10</v> <unit>dBm</unit> <c></c> <n>sentFrames</n> <v>0</v> <unit></unit> <c></c> <n>discardedFrames</n> <v>0</v> <unit></unit> <c></c> <n>authorizedUsage</n> <v>3600</v> <unit>ms/h</unit> <c> ETSI EN 300 220-1</c> <n>remainingUsage</n> <v>3600</v> <unit>ms</unit> <c></c></band></radioStatus>

The line “ZIA++1234 STATUS RADIO JSON” will return :

ZIA--{"radioStatus": {"reqNum": "1234", "band": [{"i": [{"n" : "Frequency", "v" : "433920", "unit" : "Khz", "c" : " Adapted to most 433Mhz devices"}, {"n" : "Selectivity", "v" : "0", "unit" : "", "c" : " Default value"}, {"n" : "FloorNoise", "v" : "-97", "unit" : "dBm", "c" : " A bit noisy"}, {"n" : "DspTrigger", "v" : "8", "unit" : "dBm", "c" : ""}, {"n" : "RFlink", "v" : "1", "unit" : "", "c" : " Enabled"}, {"n" : "RFlinkTrigger", "v" : "12", "unit" : "dBm", "c" : ""}, {"n" : "sentFrames", "v" : "323", "unit" : "", "c" : ""}, {"n" : "discardedFrames", "v" : "0", "unit" : "", "c" : ""}, {"n" : "authorizedUsage", "v" : "3600000", "unit" : "ms/h", "c" : " ETSI EN 300 220-1"}, {"n" : "remainingUsage", "v" : "3599737", "unit" : "ms", "c" : ""}]}, {"i": [{"n" : "Frequency", "v" : "868950", "unit" : "Khz", "c" : " Adapted to Visonic/Meian devices"}, {"n" : "Selectivity", "v" : "0", "unit" : "", "c" : " Default value"}, {"n" : "FloorNoise", "v" : "-108", "unit" : "dBm", "c" : " Very small noise"}, {"n" : "DspTrigger", "v" : "6", "unit" : "dBm", "c" : ""}, {"n" : "RFlink", "v" : "1", "unit" : "", "c" : " Enabled"}, {"n" : "RFlinkTrigger", "v" : "10", "unit" : "dBm", "c" : ""}, {"n" : "sentFrames", "v" : "1", "unit" : "", "c" : ""}, {"n" : "discardedFrames", "v" : "0", "unit" : "", "c" : ""}, {"n" : "authorizedUsage", "v" : "3600", "unit" : "ms/h", "c" : " ETSI EN 300 220-1"}, {"n" : "remainingUsage", "v" : "3600", "unit" : "ms", "c" : ""}]}]}}

5.1.1.1.2 FORMAT answer examples

FORMAT TEXT

ZIA44 FRAME: frameType: 0, dataFlag: 1,

rfLevel: -49dBm, floorNoise: -108dBm, rfQuality: 10

protocol: 2 (VISONIC), infoType: 2, frequency: 868950Khz

subType: 0 (Detector/Sensor), id: 1166992416, qualifier: 1 (Tamper)

ZIA44 FRAME: frameType: 0, dataFlag: 0,

rfLevel: -34dBm, floorNoise: -97dBm, rfQuality: 10

protocol: 4 (CHACON), infoType: 1, frequency: 433920Khz

subType: 1, id: 146139014 (ON)

ZIA44 FRAME: frameType: 0, dataFlag: 0,

rfLevel: -58dBm, floorNoise: -97dBm, rfQuality: 7

protocol: 1 (X10), infoType: 0, frequency: 433920Khz

subType: 1, id: 33 (ON, C2)

ZIA44 FRAME: frameType: 0, dataFlag: 0,

rfLevel: -78dBm, floorNoise: -97dBm, rfQuality: 3

protocol: 5 (OREGON), infoType: 4, frequency: 433920Khz

subType: 0, id_PHY: 0x1A2D (THGR122/228/238/268,THGN122/123/132)

adr_channel: 54273, adr: 212, channel: 1

qualifier: 32, lowBatt: 0, measures:

temperature: +23.4 Celsius

hygrometry: 75 %

ZIA44 FRAME: frameType: 0, dataFlag: 0,

rfLevel: -45dBm, floorNoise: -98dBm, rfQuality: 10

protocol: 7 (OWL), infoType: 8, frequency: 433920Khz

subType: 0, id_PHY: 0x0000 (CM119/160)

adr_channel: 62851, adr: 3928, channel: 3

qualifier: 4, lowBatt: 0, measures:

energy: 0 Wh

power: 0 W

ZIA44 FRAME: frameType: 0, dataFlag: 0,

rfLevel: -64dBm, floorNoise: -98dBm, rfQuality: 6

protocol: 5 (OREGON), infoType: 6, frequency: 433920Khz

subType: 0, id_PHY: 0x1A89 (WGR800)

adr_channel: 40192, adr: 157, channel: 0

qualifier: 48, lowBatt: 0, measures:

wind speed: 0.5 m/s

direction: 225 degree

FORMAT XML

ZIA22<?xml version="1.0" encoding="ISO8859-1" ?><frame><header><frameType>0</frameType> <dataFlag>0</dataFlag> <rfLevel>-73</rfLevel>dBm <floorNoise>-98</floorNoise>dBm <rfQuality>5</rfQuality>/10 <protocol>5</protocol> <protocolMeaning>OREGON</protocolMeaning> <infoType>4</infoType> <frequency>433920</frequency>Khz</header><infos><subType>0</subType> <id_PHY>0xFA28</id_PHY> <id_PHYMeaning>THGR810</id_PHYMeaning> <adr_channel>59650</adr_channel> <adr>233</adr> <channel>2</channel> <qualifier>48</qualifier> <lowBatt>0</lowBatt></infos> <measures> <type>temperature</type> <value>+23.1</value> <unit>Celsius</unit> </measures> <measures> <type>hygrometry</type> <value>81</value> <unit>%</unit> </measures></frame>

ZIA22<?xml version="1.0" encoding="ISO8859-1" ?><frame><header><frameType>0</frameType> <dataFlag>0</dataFlag> <rfLevel>-41</rfLevel>dBm <floorNoise>-97</floorNoise>dBm <rfQuality>10</rfQuality>/10 <protocol>3</protocol> <protocolMeaning>BLYSS</protocolMeaning> <infoType>1</infoType> <frequency>433920</frequency>Khz</header><infos><subType>1</subType> <id>4261483730</id> <subTypeMeaning>ON</subTypeMeaning> </infos></frame>

ZIA22<?xml version="1.0" encoding="ISO8859-1" ?><frame><header><frameType>0</frameType> <dataFlag>0</dataFlag> <rfLevel>-64</rfLevel>dBm <floorNoise>-98</floorNoise>dBm <rfQuality>6</rfQuality>/10 <protocol>5</protocol> <protocolMeaning>OREGON</protocolMeaning> <infoType>6</infoType> <frequency>433920</frequency>Khz</header><infos><subType>0</subType> <id_PHY>0x1A89</id_PHY> <id_PHYMeaning>WGR800</id_PHYMeaning> <adr_channel>40192</adr_channel> <adr>157</adr> <channel>0</channel> <qualifier>48</qualifier> <lowBatt>0</lowBatt></infos> <measures> <type>wind speed</type> <value>0.0</value> <unit>m/s</unit> </measures> <measures> <type>direction</type> <value>225</value> <unit>degree</unit> </measures></frame>

ZIA22<?xml version="1.0" encoding="ISO8859-1" ?><frame><header><frameType>0</frameType> <dataFlag>0</dataFlag> <rfLevel>-56</rfLevel>dBm <floorNoise>-93</floorNoise>dBm <rfQuality>7</rfQuality>/10 <protocol>9</protocol> <protocolMeaning>RTS</protocolMeaning> <infoType>3</infoType> <frequency>433920</frequency>Khz</header><infos><subType>0</subType> <subTypeMeaning>Shutter</subTypeMeaning> <id>6793524</id> <qualifier>7</qualifier><qualifierMeaning>Up/On</qualifierMeaning></infos></frame>

ZIA22<?xml version="1.0" encoding="ISO8859-1" ?><frame><header><frameType>0</frameType> <dataFlag>0</dataFlag> <rfLevel>-43</rfLevel>dBm <floorNoise>-98</floorNoise>dBm <rfQuality>10</rfQuality>/10 <protocol>2</protocol> <protocolMeaning>VISONIC</protocolMeaning> <infoType>2</infoType> <frequency>433920</frequency>Khz</header><infos><subType>0</subType> <subTypeMeaning>Detector/Sensor</subTypeMeaning> <id>614725408</id> <qualifier>0</qualifier><qualifierMeaning></qualifierMeaning></infos></frame>

ZIA22<?xml version="1.0" encoding="ISO8859-1" ?><frame><header><frameType>0</frameType> <dataFlag>0</dataFlag> <rfLevel>-76</rfLevel>dBm <floorNoise>-96</floorNoise>dBm <rfQuality>4</rfQuality>/10 <protocol>5</protocol> <protocolMeaning>OREGON</protocolMeaning> <infoType>4</infoType> <frequency>433920</frequency>Khz</header><infos><subType>0</subType> <id_PHY>0x1A2D</id_PHY> <id_PHYMeaning>THGR122/228/238/268,THGN122/123/132</id_PHYMeaning> <adr_channel>54273</adr_channel> <adr>212</adr> <channel>1</channel> <qualifier>32</qualifier> <lowBatt>0</lowBatt></infos> <measures> <type>temperature</type> <value>+23.4</value> <unit>Celsius</unit> </measures> <measures> <type>hygrometry</type> <value>75</value> <unit>%</unit> </measures></frame>

FORMAT JSON

ZIA33{ "frame" :{"header": {"frameType": "0", "dataFlag": "0", "rfLevel": "-71", "floorNoise": "-98", "rfQuality": "5", "protocol": "5", "protocolMeaning": "OREGON", "infoType": "9", "frequency": "433920"},"infos": {"subType": "0", "id_PHY": "0x2A19", "id_PHYMeaning": "PCR800", "adr_channel": "39168", "adr": "153", "channel": "0", "qualifier": "48", "lowBatt": "0", "measures" : [{"type" : "total rain", "value" : "1040.1", "unit" : "mm"}, {"type" : "current rain", "value" : "0.00", "unit" : "mm/h"}]}}}

ZIA33{ "frame" :{"header": {"frameType": "0", "dataFlag": "0", "rfLevel": "-41", "floorNoise": "-97", "rfQuality": "10", "protocol": "3", "protocolMeaning": "BLYSS", "infoType": "1", "frequency": "433920"},"infos": {"subType": "0", "id": "4261483730", "subTypeMeaning": "OFF"}}}

ZIA33{ "frame" :{"header": {"frameType": "0", "dataFlag": "0", "rfLevel": "-64", "floorNoise": "-97", "rfQuality": "6", "protocol": "5", "protocolMeaning": "OREGON", "infoType": "6", "frequency": "433920"},"infos": {"subType": "0", "id_PHY": "0x1A89", "id_PHYMeaning": "WGR800", "adr_channel": "40192", "adr": "157", "channel": "0", "qualifier": "48", "lowBatt": "0", "measures" : [{"type" : "wind speed", "value" : "0.4", "unit" : "m/s"}, {"type" : "direction", "value" : "225", "unit" : "degree"}]}}}

ZIA33{ "frame" :{"header": {"frameType": "0", "dataFlag": "1", "rfLevel": "-50", "floorNoise": "-107", "rfQuality": "10", "protocol": "2", "protocolMeaning": "VISONIC", "infoType": "2", "frequency": "868950"},"infos": {"subType": "0", "subTypeMeaning": "Detector/Sensor", "id": "1166992416", "qualifier": "8", "qualifierMeaning": { "flags": ["Supervisor/Alive"]}}}}

ZIA33{ "frame" :{"header": {"frameType": "0", "dataFlag": "1", "rfLevel": "-52", "floorNoise": "-107", "rfQuality": "10", "protocol": "2", "protocolMeaning": "VISONIC", "infoType": "2", "frequency": "868950"},"infos": {"subType": "0", "subTypeMeaning": "Detector/Sensor", "id": "1166992416", "qualifier": "1", "qualifierMeaning": { "flags": ["Tamper"]}}}}

ZIA33{ "frame" :{"header": {"frameType": "0", "dataFlag": "0", "rfLevel": "-41", "floorNoise": "-97", "rfQuality": "10", "protocol": "4", "protocolMeaning": "CHACON", "infoType": "1", "frequency": "433920"},"infos": {"subType": "1", "id": "146139014", "subTypeMeaning": "ON"}}}

ZIA33{ "frame" :{"header": {"frameType": "0", "dataFlag": "0", "rfLevel": "-41", "floorNoise": "-97", "rfQuality": "10", "protocol": "7", "protocolMeaning": "OWL", "infoType": "8", "frequency": "433920"},"infos": {"subType": "0", "id_PHY": "0x0003", "id_PHYMeaning": "CM180i", "adr_channel": "784", "adr": "49", "channel": "0", "qualifier": "6", "lowBatt": "0", "measures" : [{"type" : "energy", "value" : "45150", "unit" : "Wh"}, {"type" : "power", "value" : "345", "unit" : "W"}, {"type" : "P1", "value" : "345", "unit" : "W"}, {"type" : "P2", "value" : "0", "unit" : "W"}, {"type" : "P3","value" : "0", "unit" : "W"}]}}}

5.1.2 Binary data (ordinary used to sent RF frames and by RFLINK interface)

5.1.2.1 Received from USB port and send to the RF transmitter

BinaryData []	*Size*	*type*	*Remark*
FrameType	1	unsigned char	Type of packet. Value = 0
Cluster	1	unsigned char	Set always 0. Future use.
Protocol	1	unsigned char	Define the protocol. See table
action	1	unsigned char	Define operation. See table
DeviceID	4	unsigned long	LSB first DeviceID : 0…255. House Code (A-P) must be on deviceID[7…4] for pseudo X10 address format 32 bits form is only used for KD101 protocol.
DimValue	1	unsigned char	0% ... 100%
Burst	1	unsigned char	Set 0 by default.
Qualifier	1	unsigned char	Set 0 by default.
Reserved2	1	unsigned char	Set 0 by default

Home Automation Protocols, excepted X10 and DOMIA LITE, use large MacDeviceIDs to distinguish separate installations (and avoid to command easily appliances of your neighbors!). Thus, an internal USB Dongle MAC 32 bits address is used to generate the MacDeviceIDs sent over the RF. The API requests only a small 8 bits DeviceID with the internal calculation MacDeviceID = f (DeviceID).

The internal USB Dongle MAC 32 bits address is preset at factory with a random value. It can be read or set to another value with a management command.

5.1.2.2 Protocol

Protocol	Value	Remark
VISONIC_433	1	PowerCode only
VISONIC_868	2	PowerCode only
CHACON_433	3	
DOMIA_433	4	
X10_433	5	
X2D_433	6	Alarm
X2D_868	7	Alarm
X2D_SHUTTER_868	8	Qualifier7:6 =1 to emit X2D variant, else Qualifier=0
X2D_HA_ELEC_868	9	Qualifier7:6 =1 to emit X2D variant, else 0
X2D_HA_GAS_868	10	Qualifier7:6 =1 to emit X2D variant, else 0
SOMFY_RTS_433	11	Qualifier D0=1 to emit portal frame, else Qualifier D0=0 (or nothing) DIM values between 0—15 emulate specific RTS functions.

		Specify DIM %4 to emulate RTS “My” function. Qualifier D1=1 can emulate the same specific RTS functions but specified on D7:4 of the qualifier. Specify D7:4 to 4 to emulate RTS “My” function.
BLYSS_433	12	
PARROT_433_OR_868	13	ID0 to ID254 (A1…P15). ID255 (P16) is reserved and must not be used.
reserved	14	
reserved	15	
KD101_433	16	Smoke detector. 32 bits ID

5.1.2.3 Action

action	Value	Remark
OFF	0	Used by most protocols
ON	1	Used by most protocols
DIM	2	Used by some protocols
BRIGHT	3	*not used*
ALL_OFF	4	*Used by BLYSS*
ALL_ON	5	*Used by BLYSS*
ASSOC	6	Used by most protocols
DISSOC	7	*provision*
ASSOC_OFF	8	Used by PARROT for its OFF entries
DISSOC_OFF	9	*provision*

5.1.2.4 RFLINK interface (USB to RF)

See Chapter *RFLINK interface (RF to USB)* in the opposite direction.

Packet structure is the same.

5.2 USB Dongle → Host (receiving data from RF)

5.2.1 ASCII data (ordinary used by commands answers and received RF Frames)

Header	Remark
ZIA--	Synchronous answers of the commands interpreter
ZIA00	Asynchronous received RF Frames. Enabled by "FORMAT HEXA"
ZIA11	Asynchronous received RF Frames. Enabled by "FORMAT HEXA FIXED"
ZIA22	Asynchronous received RF Frames. Enabled by "FORMAT XML"
ZIA33	Asynchronous received RF Frames. Enabled by "FORMAT JSON"
ZIA44	Asynchronous received RF Frames. Set by "FORMAT TEXT"

5.2.2 Binary data (ordinary used to received RF Frames and by RFLINK interface)

5.2.2.1 Received RF Frames from regular decoder

Binary Data[]	Size	Type	Remark
FrameType	1	unsigned char	0: received RF Frames from regular decoder
Cluster	1	unsigned char	Reserved.
DataFlag	1	unsigned char	0: 433Mhz, 1: 868Mhz
RFLevel	1	signed char	Unit : dB (high signal :-40dB to low : -110dB)
FloorNoise	1	signed char	Unit : dB (high signal :-40dB to low : -110dB)
RFQuality	1	unsigned char	RF signal quality : 1/10 (poor) to 10/10 (best)
Protocol	1	unsigned char	See below. Not significant with RFLINK frame
InfosType	1	unsigned char	See below. Not significant with RFLINK frame
Infos[0…9]	20	Signed or unsigned short upon context	LSB first. Define provided data by the device Not significant with RFLINK frame.

NOTE : Binary received RF Frames flow is enabled by "FORMAT BINARY" command.

NOTE : Binary format is an alternate way to receive RF Frames. Consider FORMAT HEXA, HEXA FIXED, XML and JSON with ASCII form to develop/debug more easily a Dongle driver.

5.2.2.2 RFLINK interface (RF to USB)

The Dongle is able to provide not decoded frames records to the Host system. The Dongle assumes entirely the real time frame recording and the host (e.g Linux system) can analyze these records without real time constraints .

Visit http://www.nemcon.nl/blog2/ for further information about RFLINK. Visit https://github.com/ to get sources.

Binary Data[]	Size	Type	Remark
FrameType	1	unsigned char	1: RFLINK Frame
frequency	4	unsigned long	Frequency expressed in Khz Available : 433420, 433920, 868350, 868950.
RFLevel	1	signed char	Unit : dB (high signal :-40dB to low : -110dB)
FloorNoise	1	signed char	Unit : dB (high signal :-40dB to low : -110dB)
PulseElementSize	1	unsigned char	Value : 1
number	2	unsigned short	Number of Pulses upon RFLINK definition
Repeats	1	unsigned char	Number of re-transmits upon RFLINK definition
Delay	1	unsigned char	Delay in ms. after trans. upon RFLINK definition
Multiply	1	unsigned char	Real pulse unit in microseconds upon RFLINK definition Value = 40
Time	4	unsigned long	Timestamp indicating when the signal was received upon RFLINK definition
Pulses[]	number+2	unsigned char[]	Pulses[0] and Pulses[number+1] are set to 0 upon historical RFLINK definition

NOTE : Binary RFLINK Frames flow is enabled by “FORMAT RFLINK BINARY” command.

5.2.2.3 DataFlag

bit	7	6	5	4	3	2	1	0
	reserved	*reserved*	*reserved*	*reserved*	*reserved*	*reserved*	reserved	868Mhz Flag 0: 433 1: 868

5.2.2.4 RFLevel / FloorNoise

They are expressed in dB. One RFLevel above -70dB (-70…-40) has to be considered high.

One RFLevel under -70dB (-110dB… -70) has to be considered low. Difference between RFLevel and FloorNoise is representative of the quality of the signal (1/10 to 10/10).

The FloorNoise parameter is classically under -90dB. 868Mhz is often less noisy than 433hz because it is less used. The Floor Noise must be keep as low as possible. Avoid to set the dongle near electrical devices.

5.2.2.5 Protocol

Protocol	Value	Remark
X10	1	Use InfosType 0, InfosType 1
VISONIC	2	Use InfosType 2
BLYSS	3	Use InfosType 1
CHACON	4	Use InfosType 1
OREGON	5	Use InfosType 4, 5, 6, 7, 9
DOMIA	6	Use InfosType 0
OWL	7	Use InfosType 8
X2D	8	Use InfosType 10, 11
RTS	9	Use InfosType 3
KD101	10	Use InfosType 1
PARROT	11	Use InfosType 0
reserved	12-255	

5.2.2.6 InfosType

InfosType value	Infos[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	subType	id	-	-	-	-	-	-	-	-
1	subType	id_lsb	id_msb							
2	subType	id_lsb	id_msb	qualifier	-	-	-	-	-	-
3	subType	id_lsb	id_msb	qualifier	-	-	-	-	-	-
4	subType	id_PHY	adr_channel	qualifier	temp	hygro	-	-	-	-
5	subType	id_PHY	adr_channel	qualifier	temp	hygro	pressur e	-	-	-
6	subType	id_PHY	adr_channel	qualifier	speed	direction	-	-	-	-
7	subType	id_PHY	adr_channel	qualifier	UV	-	-	-	-	-
8	subType	id_PHY	adr_channel	qualifier	Energy_lsb	Energy_msb	power	P1	P2	P3
9	subType	id_lsb	id_lsb	qualifier	TotalRain_l sb	TotalRain_msb	rain			
10	subType	id_lsb	id_lsb	qualifier	Mode	Data 0-1	Data 2-3	Data 4-5	Data 6-7	Data 8-9
11	subType	id_lsb	id_lsb	qualifier	0	Data 0-1	Data 2-3	Data 4-5	Data 6-7	Data 8-9

5.2.2.7 InfosType 0

InfosType 0	Used by X10 / DOMIA LITE protocol / PARROT
subType	0: OFF, 1: ON, 2: BRIGHT, 3: DIM, 4: ALL_ OFF, 5 : ALL_ ON

id	Id of the device in X10 format, housecode (A…P) =D[7:4], dev (1…16) =D[3:0]

5.2.2.8 InfosType 1

InfosType 1	Used by X10 (24/32 bits ID), CHACON , KD101, BLYSS
subType	0: OFF, 1: ON/Alert ; X10 & CHACON add 4: ALL_ OFF, 5 : ALL_ ON
id_lsb	Unsigned short. Lsb of the device ID
id_msb	Unsigned short. Msb of the device ID

5.2.2.9 InfosType 2

InfosType 2	Used by VISONIC
subType	0: detector/sensor(PowerCode device), 1: remote control (CodeSecure device)
id_lsb	Unsigned short. Lsb of the device ID
id_msb	Unsigned short. Msb of the device ID
qualifier	Visonic : detector/sensor/ PowerCode device : D0 : Tamper Flag, D1: Alarm Flag, D2: Low Batt Flag, D3: Supervisor Frame Flag (Nota : D0:3 of the id always set to 0 so that id and qualifier can be ORed) remote control device (MCT-234 style) : Key Id (4 buttons) values : 0x08, 0x10, 0x20, 0x40. (Nota : D0:7 of the id always set to 0) so that id and qualifier can be ORed.

5.2.2.10 InfosType 3

InfosType 3	Used by RTS protocol
subType	0: shutter device, 1: Portal device
id_lsb	Unsigned short. Lsb of the device ID
id_msb	Unsigned short. Msb of the device ID.
qualifier	Shutters Remote control) : D0:4 : code function 1 : Down /OFF 4 : My 7 : Up / ON 13 : ASSOC portals Remote control : D0:4 : code function 5 : Left button 6 : Right button

5.2.2.11 InfosType 4

InfosType 4	Used by Scientific Oregon protocol (thermo/hygro sensors)
subType	0: regular sensor frame
id_PHY	Unsigned short. Define the recognized physical device. See Table.
adr_channel	adr_channel[15:8] defines a random device address chosen after reset or battery change. Application can take or not into account this field. adr_channel[7:0] defines the channel programmed on device. : 1..N
qualifier	D7:4 = 1 : Oregon protocol V1, 2 : Oregon V2, 3 : Oregon V3 D0 = 0: Batt Ok, D0=1 : Low Batt (less than 20% battery remaining for some devices)
temp	Signed short. Temperature (Unit : 1/10 of degrees Celsius, e.g. 213 means 21.3°C)
hygro	Hygrometry (0…100%). 0 means hygrometry not available on device

5.2.2.12 InfosType 5

InfosType 5	Used by Scientific Oregon protocol (Atmospheric pressure sensors)
subType	0: regular sensor frame
id_PHY	Unsigned short. Define the recognized physical device. See Table.
adr_channel	iadr_channel[15:8] defines a random device address chosen after reset or battery change. Application can take or not into account this field. adr_channel[7:0] defines the channel programmed on device.
qualifier	D7:4 = 1 : Oregon protocol V1, 2 : Oregon V2, 3 : Oregon V3 D0 = 0: Batt Ok, D0=1 : Low Batt (less than 20% battery remaining for some devices)
temp	Signed short. Temperature (Unit : 1/10 of degrees Celsius, e.g. 213 means 21.3°C)
hygro	Hygrometry (0…100%).
pressure	Atmospheric/barometric pressure. (Unit : hPa)

5.2.2.13 InfosType 6

InfosType 6	Used by Scientific Oregon protocol (Wind sensors)
subType	0: regular sensor frame
id_PHY	Unsigned short. Define the recognized physical device. See Table.
adr_channel	adr_channel[15:8] defines a random device address chosen after reset or battery change. Application can take or not into account this field. adr_channel[7:0] defines the channel programmed on device.
qualifier	D7:4 = 1 : Oregon protocol V1, 2 : Oregon V2, 3 : Oregon V3 D0 = 0: Batt Ok, D0=1 : Low Batt (less than 20% battery remaining for some devices)
speed	Averaged Wind speed (Unit : 1/10 m/s, e.g. 213 means 21.3m/s)
direction	Wind direction 0…359° (Unit : degrees)

5.2.2.14 InfosType 7

InfosType 7	Used by Scientific Oregon protocol (UV sensors)
subType	0: regular sensor frame
id_PHY	Unsigned short. Define the recognized physical device. See Table.
adr_channel	adr_channel[15:8] defines a random device address chosen after reset or battery change. Application can take or not into account this field. adr_channel[7:0] defines the channel programmed on device.
qualifier	D7:4 = 1 : Oregon protocol V1, 2 : Oregon V2, 3 : Oregon V3 D0 = 0: Batt Ok, D0=1 : Low Batt (less than 20% battery remaining for some devices)
UV	UV index (Unit : 1/10 index)

5.2.2.15 InfosType 8

InfosType 8	Used by OWL (Energy/power sensors)
subType	0: regular sensor frame
id_PHY	Define the recognized physical device. CM119/CM160 = 0, CM130 = 1, CM180=2, CM180i=3
adr_channel	adr_channel[15:4] defines a random device address chosen after reset or battery change for discriminate devices on a same channel. Application can take into account or ignore this field. adr_channel[3:0] defines the channel programmed by mini-switchs on the device.
qualifier	D0 = 0: Batt Ok, D0=1 : Low Batt D1=0 : Only the total instantaneous Power is given. D1=1 : Power on each input 1, 2, 3 are added (CM180i only).
Energy_lsb Energy_msb	Define the measured energy since the RESET of the device (32 bits value). Unit : Wh
Power	Define the total instantaneous measured power. Unit : W. P=UI. U=230V
P1	Define the instantaneous measured power on I1 input. Unit : W P=UI. U=230V
P2	Define the instantaneous measured power on I2 input. Unit : W. P=UI. U=230V
P3	Define the instantaneous measured power on I3 input. Unit : W. P=UI. U=230V

5.2.2.16 InfosType 9

InfosType 9	Used by Scientific Oregon protocol (Rain sensors)
subType	0: regular sensor frame
id_PHY	Unsigned short. Define the recognized physical device. See Table.
id_channel	id_channel[15:8] defines a random device address chosen after reset or battery change. Application can take or not into account this field. id_channel[7:0] defines the channel programmed on device.
qualifier	D7:4 = 1 : Oregon protocol V1, 2 : Oregon V2, 3 : Oregon V3 D0 = 0: Batt Ok, D0=1 : Low Batt (less than 20% battery remaining for some devices)
TotalRain_lsb	Define the rain measured since the RESET of the device (lsb value). Unit : 0.1mm
TotalRain _msb	Define the rain measured since the RESET of the device (msb value). Unit : 0.1 mm
Rain	Define the instantaneous measured rain. Unit : 0.01 mm/h

5.2.2.17 InfosType 10

InfosType 10	Used by Thermostats X2D protocol
subType	0: GENERIC 1: RADIO TYBOX 2: TYBOX BUS 3: PACK LABEL 4: DELTA 200 5: DRIVER RF 6: STARBOX F03 7: OTHER 8: REC BIDIR
id_lsb	Unsigned short. Lsb of the device ID D3:0 : Area
id_msb	Unsigned short. Msb of the device ID.
qualifier	D0 : Tamper Flag, D1: anomaly device, D2: Low Batt Flag, D3: - , D4: Test/Assoc D5 : Domestic frame D7:6 : X2D variant
Function	D8:0 Values : 0 : SPECIAL 1 : HEATING_SPEED 2 : OPERATING_MODE 12 : REGULATION 26 : THERMIC_AREA_STATE
Mode/State	Case with with Function = HEATING_SPEED or Function = REGULATION or Function = THERMIC_AREA_STATE : D7:0 State : 0: OFF 1: ON Case with Function = OPERATING_MODE D7:0 State : 0: ECO 1: MODERAT 2: MEDIO 3: COMFORT 4: STOP 5: OUT OF FROST 6: SPECIAL 7: AUTO 8: CENTRALISED

5.2.2.18 InfosType 11

InfosType 11	Used by Alarm X2D protocol / Shutter
subType	0: detector/sensor device, 1: remote control device / shutter
id_lsb	Unsigned short. Lsb of the device ID D12:8: sensor type, D7:0 buttons expected for remote control devices
id_msb	Unsigned short. Msb of the device ID.
qualifier	SubType=0 D0 : Tamper Flag, D1: Alarm Flag, D2: Low Batt Flag, D3: - , D4: Test/Assoc D5 : Domestic frame D7:6 : X2D variant SubType=1 On : 1 Off : 2 Stop : 3

5.2.2.19 id_PHY Oregon

id_PHY	**Commercial Scientific Oregon device references**	**Fonction**	**Prot.**
0x0	Probes Oregon protocol V1	thermometer	V1
0x1A2D	THGR122/**228**/238/268, THGN122/123/132	Thermo+hygro	V2
0xCA2C	THGR328	Thermo+hygro	V2
0x0ACC	RTGR328	Thermo+hygro	V2
0xEA4C	THC238/268, **THWR288**,THRN122,THN122/132,AW129/131	thermometer	V2
0x1A3D	**THGR918**/928, THGRN228, THGN50	Thermo+hygro	V2
0x5A6D	THGR918N	Temp+Pressure	V2
0x1A89	WGR800	Wind sensor	V3
0xCA48	THWR800	S. pool thermo	V3
0xFA28	**THGR810**, THGN800	Thermo+hygro	V3
0x2A19	PCR800	Rain sensor	V3
0xDA78	UVN800	UV sensor	V3

5.3 Firmware Update

Firmware can be updated by a simple encrypted text file. Firmware files are provided by Ziblue.

A given version of firmware cannot be flashed twice times and is rejected the subsequent times. Return to older versions is allowed.

Current firmware version is visible by STATUS SYSTEM instruction.

The update time is a relatively long (2mn):

- One minute is spent to download the firmware through the serial line.
- One minute is spent to verify file integrity at each step and then program the embedded controller.

The LED is RED and blinking during this time. The LED remains inactive if the downloaded firmware version is already running. Wait the end of RED blinking.

The new firmware automatically starts after programming. Restart the dongle (unplug then plug again the dongle) if it doesn't restart.

A new firmware can be programmed by different ways :

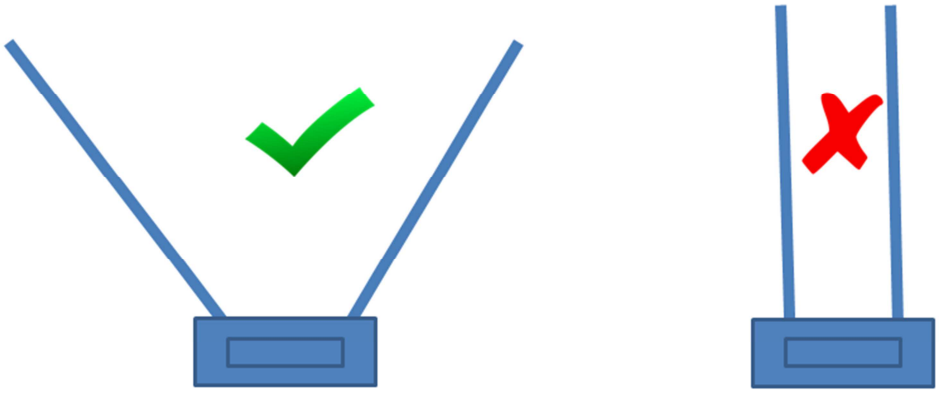
- By the Ziblue RFPLAYER configurator,
- By serial emulators available on the market (eg. TeraTerm, do "push a file"),
- By Home Automation BOX. Of course, do not send home automation commands during this time.

Previous user's configuration is preserved.

NB: The principle is "entirely store, verify, then flash, then verify again", so that power can be down at each step of the programming without damage and the programming/flashing phase isn't critical (no "brick" effect).

5.4 Antennas

Best sensitivity



5.5 LED signalisation

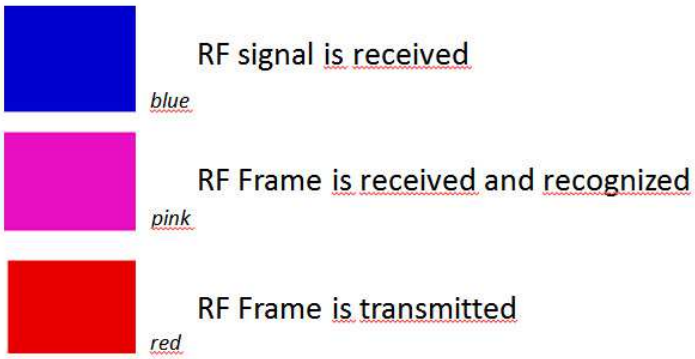
At startup

Pink flash during 1 second	REPEATER function is enabled

One very short blue flash then…	Hardware error :
1x RED flash	Low Frequency transceiver failure (433Mhz)
2x RED flashes	High Frequency transceiver failure (868Mhz)
3x RED flashes	EEPROM failure
4x RED flashes	SFLASH failure

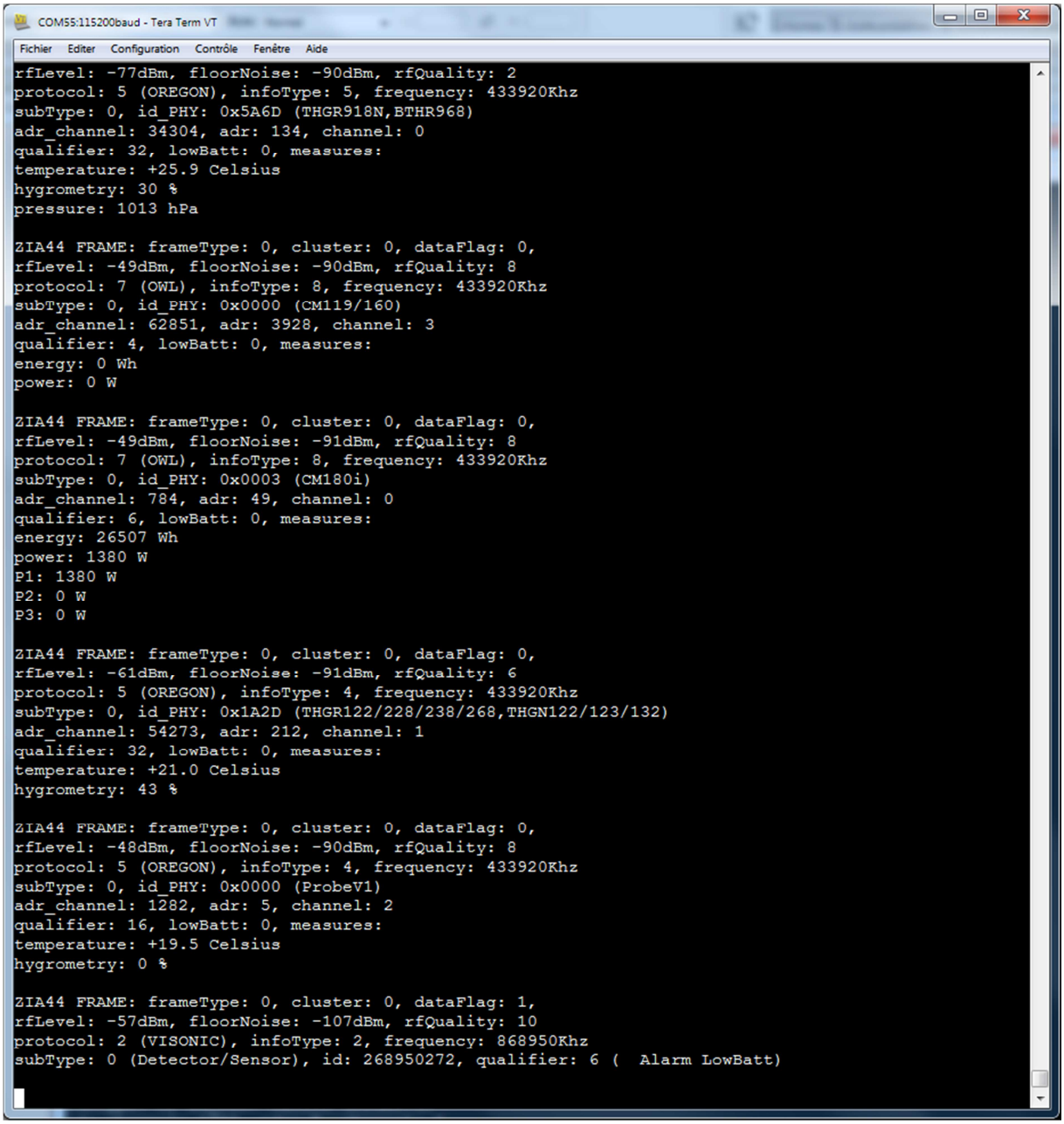
Blue/Red smooth variation during 2 seconds	Frequency calibration error

During operation

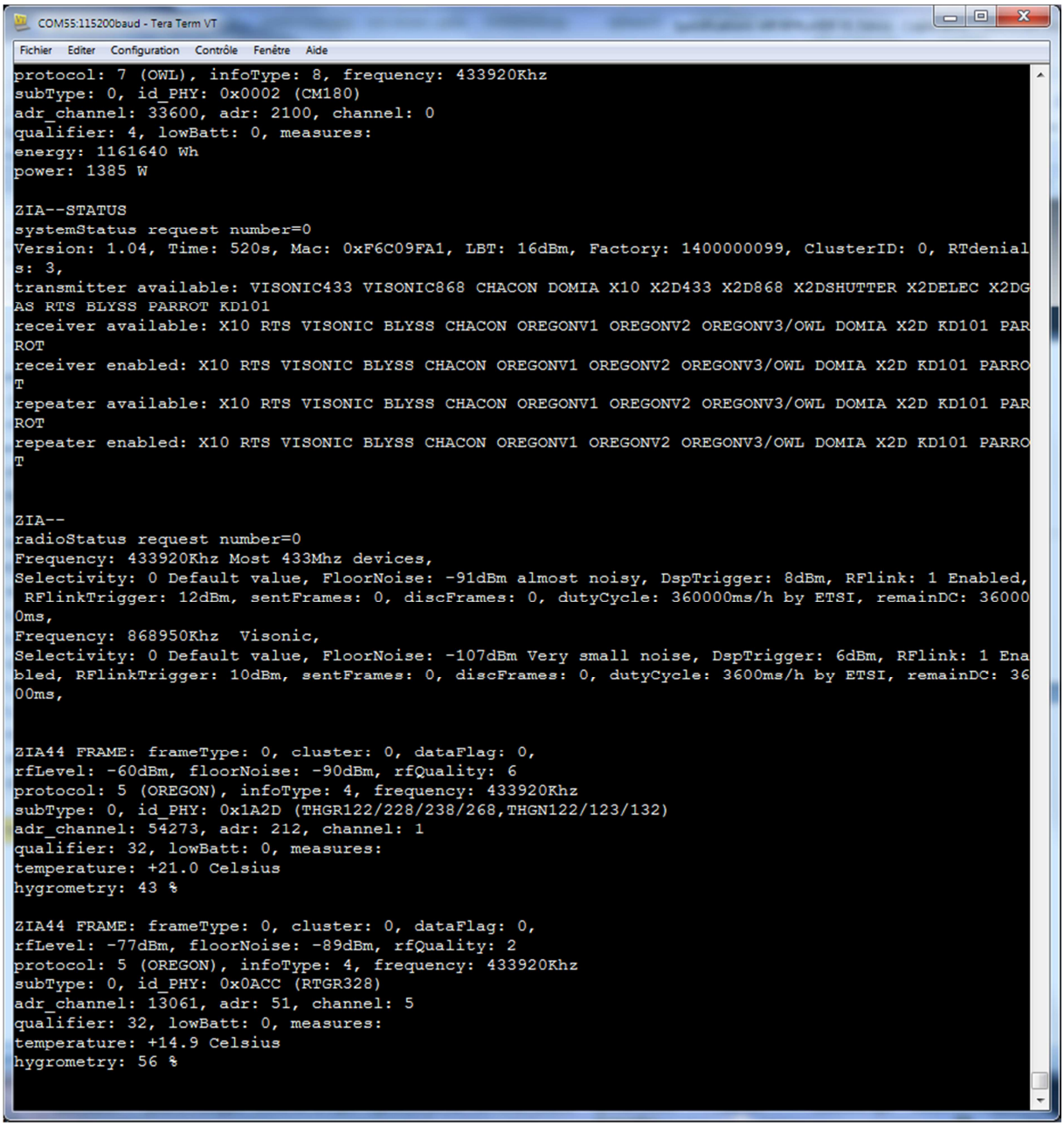


5.6 Screenshots

After FORMAT TEXT command :



After STATUS command :



Tera Term configuration :

END of document.