

RF Transceiver Module, 868 MHz, Mesh, TSCH, AES Encryption

Applications

- Low power wireless Transceiver
- Consumer Electronics
- Alarm and security systems
- Home and building automation
- Wireless sensor networks
- Industrial monitoring and control



1 Electrical Specifications

1.1 Absolute Maximum Ratings

Parameter	Min.	Max.	Units
Supply Voltage, VDD and VRF with Respect to GND	0	3.8	V
Voltage on any pin with Respect to GND	-0.3	VDD+0.3	V
Input RF level		10	dBm
Storage temperature range	-50	150	°C
Ambient Temperature under Bias	-40	125	
Maximum soldering temperature		260	°C

1.2 General Operating Conditions

Parameter	Min.	Typ.	Max.	Units	Condition
Power Supply	1.98	3.3	3.6	V	(1)
Operating Temperature	-30		+70	°C	
GPIO Input low voltage			0.3VDD	V	
GPIO Input high voltage	0.7VDD			V	
Output high voltage		0.8VDD		V	Sourcing 0.1 mA, VDD=1.98 V
		0.9VDD		V	Sourcing 0.1 mA, VDD=3.0 V
		0.85VDD		V	Sourcing 1 mA, VDD=1.98 V
		0.9VDD		V	Sourcing 1 mA, VDD=3.0 V
	0.6VDD			V	Sourcing 20 mA, VDD=1.98 V
	0.8VDD			V	Sourcing 20 mA, VDD=3.0 V

(1): Equipment is powered by a circuit (separately approved, not investigated) that is insulated from the mains supply by an insulation not less than that between the primary and secondary circuits of a safety isolating transformer according to IEC 61558-2-6 or equivalent.

The system is considered (only) for the conditions of network environment 0 (see CEI CLC/TR 62102:2006). No over-voltage from telecommunication network possible, no subject to transient over voltages due to atmospheric discharges and faults in power distribution systems. **Max VDD and VRF Ramp Time is 3 ms.**



1.3 RF Characteristic

Parameter	Min.	Typ.	Max.	Units	Condition
Frequency band	865.15		867.85	MHz	
Number of channels		28			
Channel Spacing		100		kHz	
Frequency Modulation		GFSK			
Frequency Deviation		+/- 20		kHz	
RF Rx Filter Bandwidth		83.2		kHz	
RF Data Rate		40		kbps	
Sensitivity		-108		dBm	
RF Output Power	-35		16	dBm	See 4.1.12

1.4 Current Consumption

Node Type	Min.	Typ.	Max.	Units	Condition
All		3		uA	@ Power Down
		16.5	25	mA	@ RX Continuous
		80		mA	@ TX Max Output Power
END NODE		13		uA	EM6 NO RX/TX (2)
		19		uA	EM6 1 RX/TX every 5 min (3)
		15		uA	EM5 NO RX/TX
		17		uA	EM4 NO RX/TX
		45		uA	EM0 NO RX/TX
		18		uA	EM6 NO RX/TX
ROUTER		193		uA	EM0 NO RX/TX
		300		uA	EM0 1 RX/TX every 5 sec (3)
COORDINATOR		1.5		mA	NO RX/TX

(2): For END NODE EM6 NO RX/TX we mean a device configured as END NODE (see 4.1.9) in Energy Mode EM6 (see 4.1.4 and 4.1.8) without receiving and transmitting anything

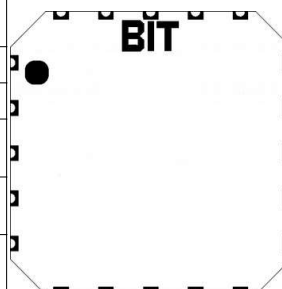
(3): TX @ Default RF Output Power (see 4.1.12)



2 PIN-OUT

PIN #	BIT Assigned Function	Type	Optional			
			UC Pin	Analog	Timers	Comm
1	Module UART READY	DIG OUT	PD_13			
2	HOST UART READY	DIG IN	PD_14			
3			PD_15	ADC0_CH7	TIM0_CC0#23 PCNT0_S0IN#23	US1_CLK#21
4			PA_0	ADC0_EXTN ADC0_CH8	TIM1_CC0#0 PCNT0_S0IN#0	US1_CS#29
5			PA_1	ADC0_EXTP ADC0_CH9	TIM1_CC0#1 PCNT0_S1IN#0	US1_TX#1
6			PB_11	ADC0_CH27	TIM1_CC0#6 PCNT0_S1IN#5	US1_RX#5
7			PB_12	ADC0_CH28	TIM0_CC0#7 PCNT0_S0IN#7	I2C0_SCL#6
8			PB_13	ADC0_CH29	TIM0_CC0#8 PCNT0_S0IN#8	I2C0_SDA#8
9	UART RX	DIG IN	PC_10	ADC0_CH10 DAC0_OUT1ALT#4	TIM0_CC0#15 PCNT0_S0IN#15	US0_RX#14
10	UART TX	DIG OUT	PC_11	ADC0_CH0 OPAMP_OUT2#1	TIM0_CC0#16 PCNT0_S0IN#16	US0_TX#16
11	RESETn	DIG IN	Reset input, active low. To apply an external reset source to this pin, it is required to only drive this pin low during reset, and let the internal pull-up ensure that reset is released.			
12	DGB-SWDIO # 0/1/2/3	DIG I/O	PF_1		TIM0_CC1#5 LETIM0_OUT1#2	US1_CS#2 LEU0_RX#3 I2C0_SCL#5
13	DGB-SWCLK # 0/1/2/3	DIG IN	PF_0		TIM0_CC0 #5 LETIM0_OUT0#2	US1_CLK #2 LEU0_TX#3 I2C0_SDA#5
14	GND	0V				
15	VDD	3.3 V IN				
16	GND	0V				
17	ANT	RF I/O				
18	GND	0V				
19	NC	NC				
20	GND	0V				

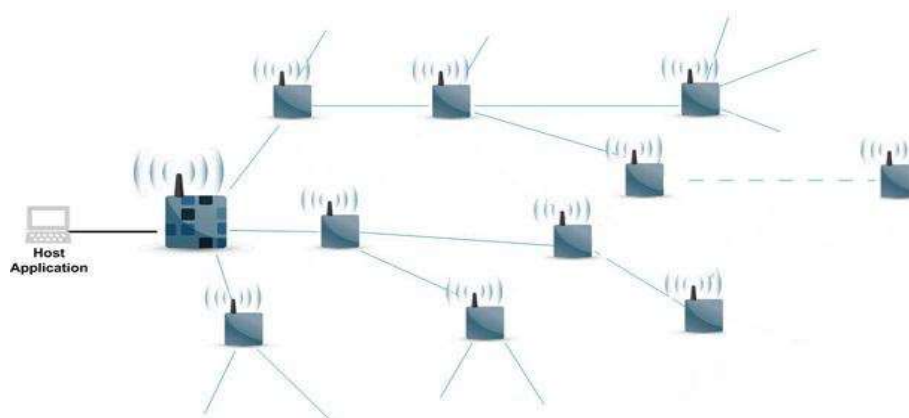
Top View



3 Description

BIT868MN is a transceiver module designed for low-power applications operates in the 865.15-867.85 MHz band.

The module implements a proprietary Low Power, Secure, Time Synchronized Mesh Network Protocol (hereinafter **BITMESH**)



A **BITMESH** Network consist in an set of nodes of different type:

C node: Network Coordinator, it monitors, configures and manage the network. Only the coordinator can join the unknown nodes into the network.

R node: Routers nodes, nodes that can expand the sync messages, they can get and relay the unknown nodes join request to the coordinator node.

E node: End nodes, nodes that can't expand the sync messages and so they cannot get and relay the unknown node join request.

S node: Satellite nodes, nodes not joined into the network but that they can interact with the joined nodes with the asynchronous slot mechanism

The primary benefits of **BITMESH** are

- Low power consumption
- Power management and optimization
- Auto-forming mesh technology for a self-healing and self-sustaining network
- Network management and configuration
- Zero collision low power packet exchange
- Scalability to large, dense, deep networks

BITMESH includes a Time Slotted Channel Hopping (TSCH) media access layer (MAC). TSCH works by dividing time into 'slots', and providing a mechanism to map time-slots to channels with a pre-assigned hopping sequence.

BITMESH also implements encrypts the frame according to the Advanced Encryption Standard (AES) algorithm. Automatic Frequency Compensation (AFC) and Listen Before Talk (LBT) are also implemented.

The built-in **BITMESH** intelligence utilizes TDMA and TSCH to ensure:

- Nodes know precisely when to talk, listen, or sleep
- Packet exchanges are synchronized
- No packets collide on the network
- Every packet is scheduled and synchronized for energy efficiency with no extra preamble (TX side) or guard interval time (RX side)

Network synchronization also enables pair-wise channel hopping – also known as frequency diversity - that provides important benefits throughout the network:

- Multiple transmissions can occur simultaneously, increasing network bandwidth
- Automatically changes channels to avoid inevitable
- RF interference Allows networks to be dense and scale without creating debilitating RF interference

RF Power Level used in transmission is dynamically adjusted in order to reduce power consumption. For example, if two nodes are close, the RF Power is reduced to the minimum level allowed to securely communicate between the same.

BITMESH also addresses the hidden node problem. Beacon Frames are sent with the Max RF Output Power (see 4.1.12) – 3 dB in order to reduce this problem.

In wireless networking, the hidden node problem or hidden terminal problem occurs when a node is visible from a wireless access point (AP), but not from other nodes communicating with that AP. This leads to difficulties in media access control sub-layer; for example it is difficult to implement LBT (see 3.4).

The module is driven by the host via a UART interface.

The following describes the features of the network.

3.1 TSCH

Each Node, when it joins the network, is assigned a time slot of 25 ms (nominal), a virtual channel chosen from among the 18 channels for data communications, a channel chosen from 8 for forwarding the sync message and eventually a channel from 2 for forwarding an asynchronous message.

A sequence of 50 slots forms a SLOT FRAME (1.25 seconds) while a sequence of 240 SLOT FRAME forms a SUPER SLOT FRAME (300 seconds).

Inside a SLOT FRAME, the router and end nodes are able to receive only one message, while the coordinator is able to receive 4 messages per frame.

The module hops on 28 channels following a table generated by a pseudo-random algorithm different for each COORDINATOR and will remain on each channel for 25 ms (nominal). 26 channels are used for synchronous communications and 2 channels are used for asynchronous communications.

Thus the following example table is obtained for secure transmission every 1.25 sec.

Slot #	00	01	02	03	04	...	48	49
Ch #								
00		CRD RX				...	RES	RES
01			NODE3 RX		NODE2 RX	...	RES	RES
02	RES	RES	RES	RES	RES	...	ASYNC1	ASYNC2
03	CRD BCN TX		Node2 BCN TX			...	RES	RES
...	RES	RES
24		NODE1 BCN TX		NODE3 BCN TX		...	RES	RES
25	RES	RES	RES	RES	RES	...	ASYNC3	ASYNC4
26			NODE1 RX				RES	RES
27					NODEn RX		RES	RES

Grey cells are not used.

Please note that every channel is one of the previous defined RF channels and varies for every SLOT FRAME.



3.2 AES128

All communications are encrypted with AES 128-bit. There are 2 encryption keys: a public key and a private key. The less significant bits of the private key will be used as coefficients of a random number generation algorithm which will serve to generate a TSCH hopping table.

The public key has to be set equal in all modules of the same installation. The private key has to be set different for different facility. Different Encryption Key generate different BEACON Frames so avoiding synchronization of Nodes with coordinators of other installations.

The default public encryption key is composed by the 16 char **BITPBLENCRYPTKEY**.

The default private encryption key is composed by the 16 char **BITPRVENCRYPTKEY**.

Private encryption key can be set by the apposite command (see 4.1.6); alternatively it could be set only in the coordinator and then will be passed to the other nodes during the join procedure.

To send the private AES key at nodes that have only the public AES key during the "Network Join", the coordinator must be set up to use the public key and build the network with that. When all nodes are joined into the network, its possible to change the key to use in the coordinator (see 4.1.7) and him with a broadcast silently message will change the used key for all the network.

3.3 AFC

The nodes implement the Automatic Frequency Compensation. It is an algorithm that causes the frequency (Transmit/Receive) of the PERIPHERAL is always centered on that of the COORDINATOR so as not to lose sensitivity and continue to operate even in the presence of temperature and/or supply voltage variations.

Important

If the temperature varies too quickly (for example from 25 to 70 ° C in less than 5 minutes when Network Variant is set in Low Power Mode – see 4.1.4), it is not guaranteed the continuity of operation; we mean that the NODE could go out-of-sync and could therefore not be able to communicate with the COORDINATOR for a round of synchronization (about 5 minutes if the Network Variant is set in low power mode). After receiving the next Beacon Frame, the NODE gets back in sync and thus is able to communicate with the COORDINATOR.

If the temperature varies slowly enough (for example 25 to 70 ° C for more than an hour) the AFC algorithm allows normal operation.

3.4 LBT

BITMESH implements the technique of Listen Before Talk to avoid interference in case of two nodes (in example a father and a child of a node) who want to transmit to the same node. This is similar to what Carrier Sense Multiple Access technique do:

- > Sense the channel
- > If the channel is idle, transmit
- > if the channel is busy,
 - waits a random amount of time
 - sense the channel again

3.5 Send Message

BITMESH is a complete address network, every single node can send messages only to another known node. The messages must be addressed by the destination node's long address and only if the this node was precedently joined in to the network.

A node can send messages to its father, all its nodes sons and to the network coordinator. Different branches of the network cannot be reached through the normal routing of messages, to make the different branches reachable, it is necessary to implement a mechanism that allows routing through the coordinator.



TSCH access requires that to transmit a message, the operation must be scheduled and it will be performed at the appropriate time (slot). To schedule a transmission use the "STX" (see 4.1.20) command in which it is necessary to enter the "long address" of the destination node. The source node will proceed independently according to the network to correctly route the message through the other nodes.

3.6 Auto-Acknowledge and Re-transmissions

All messages sent via BITMESH are sent in AutoAck mode, within an acknowledgment message it is possible to insert 9 bytes of data set using the "SAA" command.

if a node does not respond with a correct acknowledgment, the source node tries again to send the message a number of times depending on the network stress level, if the attempts are finished and the node does not respond, it will be eliminated from the network and the coordinator will be informed of the node loss.

3.7 Flooding broadcast message sending

BITMESH allows to send a message to all the nodes in the network through a mechanism called flooding. Once the flooding process has started, the node sends the message to all its direct children and to its father. All nodes that receive the message (including the coordinator) do the same thing by forwarding the message to all direct nodes except who sent the message to them.

Also the flooding broadcast message is sent in AutoAck mode, and in the same way of normal messages, every single acknowledge messages can contain 9 bytes of data.

To send a flooding broadcast message, use the STX command with 0xFFFFFFFF as destination long address.

3.8 Synchronization and Join

BITMESH is a synchronized network, in order to keep all nodes correctly synchronized, the coordinator and the router nodes send a periodically sync message called beacon.

A node not joined to the network can join the network only after receiving a beacon sent by the coordinator or a router node. After entering the network, regardless of the energy mode parameter, a node starts sending a series of beacon with the lowest energy mode in order to simplify the joining of unknown nodes into the network. after the first transient in which the network is build, each router starts sending the beacon message with a frequency set by the energy mode parameter. If a node loses the network for reintegration into the network, it must wait for the next beacon.

The energy mode parameter influences the speed with which the network can re-enter the lost nodes.

The node coordinator, send a beacon every 4 SLOT FRAME independently of the EM parameter. The nodes routers can send and receive a beacon every 4 – 8 – 16 – 32 – 64 – 128 – 256 SLOT FRAME dependently of EM parameter (0 – 1 – 2 – 3 – 4 – 5 – 6).

The routers beacon interval is the same for every routers nodes and is set from the coordinator during the join procedure. In order to change the network "Energy Mode" is necessary change the EM parameter from the coordinator node. The new set will be applied after the beginning of the new SUPER SLOT FRAME.

The Beacon time can be calculated by the formula:

$$Time_{BCN} = (4 \times 2^{EM}) \times T_{FRAME} \text{ sec}$$

Where EM is Energy Mode Value and T_{frame} is the duration of a SLOT FRAME 1.25 sec.

3.9 Life State Supervision

BITMESH implements an automatic supervision of the life state of each node in the network. Each Node supervises its direct own child nodes. The mechanism is organized in such a way as to generate a life state request only if there was not communications with the supervised node for a determinate number of SLOT FRAME.

In order to minimize the power consumption, every communications (Data or sync) with the supervised node reset the life state frame counter of the node.

In a **BITMESH** network the "life state supervision time" parameter is fixed from the coordinator and passed to the other nodes during the join procedures. each node joined into the network independently of its "life state supervision time" parameter use the parameter obtained from the coordinator.

3.10 Asynchronous Slot

The Asynchronous Slot (hereinafter ASC) is a system that allows a node enrolled in the network to communicate with a node outside from the network. This feature allows the implementation in the network of asynchronous devices such as radio commands.

Any node can request the CRD of an ASC slot through the UART command SAS.

Once the ASC parameters have been received, the node starts sending a periodically signal containing the data entered in the "Asynchronous Field" field that can be set with the SAF command. At this signal, the Satellite nodes respond with a data contained in the "Satellite Answer" field that can be set using the SSA command.

The ASC is assigned by the CRD Node. Enable the ASC means sending a request to the CRD Node to obtain the ASC parameters. Disable ASC means sending a message to the CRD node to inform it that it must free the ASC parameters, in order to make them available for another node.

3.11 Static and Volatile Settings

There are 2 types of settings.

The statics settings are memorized in the non volatile memory and when the node is rebooted these settings are used for config the node. The static settings are set with the "Write Commands (W)" only if the node is in "Config Mode" and can be read with the "Read Commands (R)".

The volatile settings are memorized in RAM and when the node is rebooted this settings are lost. The volatiles settings are set with the "Sets Commands (S)" and can be read with "Volatile Commands (V)".

The volatile settings are used for the extraordinary operations for example its possible change the "energy mode param" without reset the network, or its possible set the asynchronous slot to a node for little time without reset the network.



4 UART Interface

To interact with the host, a node uses a UART interface. the node's UART is have 8 bit char, No parity bit, 1 stop bit and the baud rate is set at 115200 of default, but it can be changed with the "WBD" command.

The "**BITMESH** UART Interface", provides that in addition to the pins of "UART TX" and "UART RX", there are two pins called "Host Ready" and "Module Ready" which are used for the communication handshake.

If host want to send an UART message, it must set the "Host Ready" pin to logic HI (for wake-up module) and wait for the module to set its "Module Ready" pin to logic HI.

If module has to send an UART message, it set the "Module Ready" pin to logic HI (for wake-up host) and wait for the host to set the "Host Ready" pin to logic HI.

The handshake procedure can be disable by pull-up both handshake pins at Vdd.

Any UART message has to be terminated by a carriage return and a new line character (\r\n - 0x0D 0x0A).

At start-up and after any command, BIT868MN print the prompt: "\r\n>:\".

4.1 UART Command table

The **BITMESH** UART Command Interface API is a set of commands that can be sent to the module.

All commands are composed of 3 char and the eventually parameters. The first char of command indicate the type of command.

ASCII code R indicate a statics settings read command.

ASCII code V indicate a volatile settings read command.

ASCII code W indicate a statics settings write command.

ASCII code S indicate a set command.

In the following C means the ASCII coded C char.

In the following table the \r\n characters and the prompt: "\r\n>:\" are not reported.

Command Description	Syntax	Cap.	Def Fab Value
Read Configuration Mode Status	<u>RCM</u>	See Cap. 4.1.1	<u>RST</u>
Sets Configuration Mode Status	<u>SCM=val</u>		
Write all settings to fab values	<u>WDF=val</u>	See Cap. 4.1.2	
Read UART Baud Rate	<u>RBR</u>	See Cap. 4.1.3	<u>3</u>
Write UART Baud Rate	<u>WBR=val</u>		
Read Network Variant	<u>RNV</u>	See Cap. 4.1.4	0x00FA6300
Write Network Variant	<u>WNV=val</u>		
Read Public AES Encryption Key	<u>REK</u>	See Cap. 4.1.5	<u>BITPBLENCRYPTKEY</u>
Write Public AES Encryption Key	<u>WEK=val</u>		
Read Private AES Encryption Key	<u>RPK</u>	See Cap. 4.1.6	<u>BITPRVENCRIPTKEY</u>
Write Private AES Encryption Key	<u>WPK=val</u>		
Read Static AES KEY in use	<u>RKT</u>	See Cap. 4.1.7	<u>PRV</u>
Set AES KEY to use	<u>SKT=val</u>		



Command Description	Syntax	Cap.	Def Fab Value
Read Static Energy Mode	<u>REM</u>	See Cap. 4.1.8	<u>0</u>
Set Energy Mode	<u>SEM=val</u>		
Read Volatile Energy Mode	<u>VEM</u>		
Read Node Type	<u>RNT</u>	See Cap. 4.1.9	<u>U</u>
Write Node Type	<u>WNT=val</u>		
Read Long Address	<u>RLA</u>	See Cap. 4.1.10	0xFFFFFFFF
Write Long Address	<u>WLA=val</u>		
Read Life State Supervision Time	<u>RLT</u>	See Cap. 4.1.11	0x0003
Write Life State Supervision Time	<u>WLT=val</u>		
Read Max RF Output Power	<u>RPA</u>	See Cap. 4.1.12	<u>80</u>
Write Max RF Output Power	<u>WPA=val</u>		
Read Power Down Status	<u>RPD</u>	See Cap. 4.1.13	<u>0</u>
Write Power Down Status	<u>WPD=val</u>		
Read Static Asynchronous Slot	<u>RAS</u>	See Cap. 4.1.14	<u>RES, NUL</u>
Write Static Asynchronous Slot	<u>WAS=v1, v2</u>		
Set Asynchronous Slot	<u>SAS=v1, v2</u>		
Read Volatile Asynchronous Slot	<u>VAS</u>		
Set Asynchronous Field	<u>SAF=val</u>	See Cap. 4.1.15	–
Read Asynchronous Field	<u>RAF</u>		
Write Asynchronous Field	<u>WAF=val</u>		
Read Satellite Answer	<u>RSA</u>	See Cap. 4.1.16	–
Set Satellite Answer	<u>SSA=val</u>		
Write Satellite Answer	<u>WSA=val</u>		
Read Acknowledge Answer	<u>RAA</u>	See Cap. 4.1.17	–
Write Acknowledge Answer	<u>WAA=val</u>		
Set Acknowledge Answer	<u>SAA=val</u>		
Read Firmware Version	<u>RFW</u>	See Cap. 4.1.18	
Print Network Tree	<u>RPN</u>	See Cap. 4.1.19	
Enqueue an RF TX	<u>STX=val</u>	See Cap. 4.1.20	

For the responses to UART commands see Cap. 4.2



4.1.1 CF: Configuration Mode

In order to accept the write commands, BIT868MN must be in configuration mode.

To enter configuration mode host has to send the command SCM=SET.

When the host sends the SCM=RST command, the module saves all current settings in NVM and reboots.

Sending the command SCM=RES will cause the device to restart even if the device is not in configuration mode

RCM command return current status.

4.1.2 DF: Default Fab Values

WDF=DEF sets all settings parameters to default values.

4.1.3 BR: UART Baud rate

WBR=val sets the UART baud rate. The UART baud rate can be set to 4 different values. The possible settings are shown in next table.

val	UART Baud Rate (bps)
<u>0</u>	9600
<u>1</u>	38400
<u>2</u>	57600
<u>3</u>	115200 (default)

RBR command return current baud rate setting.

4.1.4 NV: Network Variant

WNV=val sets some parameters of the network. Usually only the coordinator needs to be set with these parameters because, during the join procedure, the coordinator sends they to all nodes.

val is composed of 4 bytes: BYTE0, BYTE1, BYTE2, BYTE3

BYTE 0 is a bit field value as follows

BIT	7	6	5	4	3	2	1	0
Description	RES	VRB1	VRB0	AES Key to Use	RES	Energy Mode		

Energy Mode can be a value between 0 and 6. This parameter influences the reactivity of the network and in this way also affects the energy consumption of the nodes configured as ROUTER or END NODE.

For more information see section 3.8.

Default fab value is 0.

"AES Key to Use" parameter force module to use Private or Public AES Key in network join operations.

"AES Key to Use" = 0 force module to use Private AES Key

"AES Key to Use" = 1 force module to use Public AES Key

Default fab value is 0.



VRB0, VRB1 set the Verbose mode of Unsolicited Messages see section 4.3.

VRB1	VRB0	Note	Cap.
0	0	Normal type of unsolicited message only are enabled.	See Cap. 4.3.1
0	1	Network type of unsolicited messages and lower levels are enabled	See Cap. 4.3.2
1	0	System type of unsolicited messages and lower levels are enabled	See Cap. 4.3.3
1	1	Debug type of unsolicited messages and lower levels are enabled	See Cap. 4.3.4

(*) Please note that level 3 make a very intensive use of UART; so it is intended only for debug purpose: it must be not used in production design.

VERB0=0 and VERB1=0 are the default fab value.

RES are reserved fields: set always to 0

BYTE 1 is max level of network hops; it can takes values from 1 to 99; default fab value is 99

Please note that the nodes that have been associated in the max level do not send Beacon Frames in order to reduce power consumption.

BYTE 2 is Number of Nodes of the network; it can takes values from 1 to 250; default fab value is 250.

BYTE 3 is the min level value of the father to which a node can join; it can takes values from 0 to 99; default fab value is 0. This field has no effect for the coordinator.

RNV returns the current network parameters.

4.1.5 EK: Public AES Encryption Key

WEK=val sets module PUBLIC AES ENCRYPTION KEY. val is a 16 bytes value. Please note that all nodes in the network have to be set with same AES Key in order to work properly.

REK returns current PUBLIC AES ENCRYPTION KEY

4.1.6 PK: Private AES Encryption Key

WPK=val sets module PRIVATE AES ENCRYPTION KEY. val is a 16 bytes value. Please note that all nodes in the network have to be set with same AES Key in order to work properly.

RPK returns current PRIVATE AES ENCRYPTION KEY

Private encryption key can be set by the apposite command; alternatively it could be set only in the coordinator and then will be passed to the other nodes during the join procedure.

In this case, Coordinator has to be set to use the public key by the SKT command (see 4.1.7) during the join procedure. Once the join procedure is finished, Coordinator has to be set to use the Private Key.



4.1.7 KT: Key type

SKT=val change the "AES Key to Use"

val=PBL force the module to use the public key for AES encryption.

val=PRV force the module to use the private key for AES encryption.

Please note that: SKT is the only "set command" which the result will be saved in the "non volatile memory".

RKT return the static value of "AES Key to Use" field written by the WNV command.

PBL indicate that the module is using the public key for AES encryption.

PRV indicate that the module is using the private key for AES encryption.

4.1.8 EM: Energy Mode

REM return current static value of "Energy Mode" parameter written by the WNV command.

SEM=val set the volatile Energy Mode parameter. val could be a character from 0 to 6

Please note that in order to correctly inform each node of the change the new "Energy Mode" will be applied at the beginning of the new SUPER SLOT FRAME

VEM return the volatile value of "Energy Mode" written by the SEM command.

4.1.9 NT: Node Type

RNT returns current node type.

Default fab value is u undefined.

WNT=val val could be:

C sets node type COORDINATOR;

R sets node type ROUTER;

E sets node type END DEVICE.

S sets node type SATELLITE;

4.1.10 LA: Long Address

WLA=val writes the module Long Address. val must be a unique identifier on the network. It is a 4 bytes value in Little Endian Order.

RLA returns current Long Address.

Default fab value is 0xFFFFFFFF.

Please note that a node with default long address will not be associated in the network by the coordinator node.

4.1.11 LT: Life State Supervision Time

WLT=val writes the module Life State Supervision Time. val can takes values from 0 to 0xFFFF. It is a 2 bytes value in Little Endian Order. Val is expressed in multiple of 10 sec.

RLT returns current Life State Supervision Time.

Default fab value is 0x0003.

4.1.12 PA: Max RF Output Power

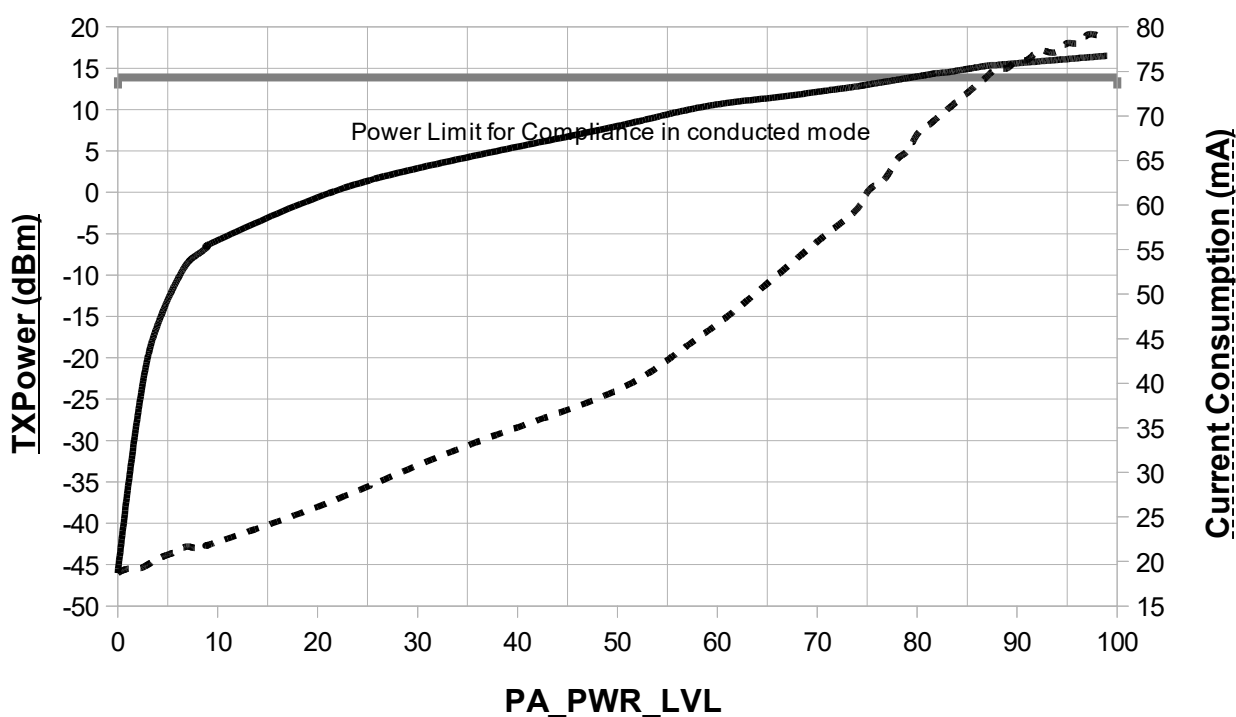
WPA=val writes the maximum output power of the module. val could be 2 ASCII coded chars from 00 to 99. Please note that the RF Power Level used in transmission is dynamically adjusted in order to reduce power consumption. For example, if two nodes are close, the RF Power is reduced to the minimum level allowed to securely communicate between the same.

Beacon Frame are always sent with the Max RF Output Power – 3 dB in order to reduce the hidden node problem.

In wireless networking, the hidden node problem or hidden terminal problem occurs when a node is visible from a wireless access point (AP), but not from other nodes communicating with that AP. This leads to difficulties in media access control sub-layer (for example it is difficult to implement LBT).

The Max RF Output Power is changed in 100 steps through the PA_PWR_LVL val according to the following figure where the relative current consumption is also reported.

Please note that for regulatory compliance max Output Power is 13.9 dBm in conducted mode.



RPA returns the current maximum output power of the module. Default fab value is 80.

4.1.13 PD: Power Down Mode

WPD=val sets the Power Down mode of the module.

If argument val is SET, module enter power down mode. If val is RST, BIT868MN resume normal operation.

RPD return current Power Down Status.

4.1.14 AS: Asynchronous Slot

WAS=v1, v2 enable and disable the static parameters for ASC Mechanism.

SAS=v1, v2 enable and disable the volatile parameters for ASC Mechanism.

V1 and v2 parameters could be:

Function	Key Type	Syntax
Enable ASC	Public Key	v1 → <u>SET</u> , v2 → <u>PBL</u>
	Private Key	v1 → <u>SET</u> , v2 → <u>PRV</u>
Disable ASC	Not Used	v1 → <u>RST</u> , v2 → <u>NUL</u>

RAS returns the current static values of ASC parameters.

VAS returns the current volatile values of ASC parameters.

4.1.15 AF: Asynchronous Field

WAF=val write the static Asynchronous Field data buffer. It is a 24 bytes of data.

SAF=val write the volatile Asynchronous Field data buffer. It is a 24 bytes of data.

The AF buffer is written in NVM with the WAF command, but it can be run time replaced with SAF command without reboot the module.

After the power of or a reboot, the module reload the buffer saved in NVM.

Please note that the "Long Address" of the source of Asynchronous Message is already managed in the mechanism, so is not necessary insert the "Long Address" in the Asynchronous Field.

RAF return the current AF data buffer used as data for Asynchronous messages.

4.1.16 SA: Satellite Answer

WSA=val write the static Satellite Answer Field data buffer. It is a 8 bytes of data.

SSA=val write the volatile Satellite Answer Field data buffer. It is a 8 bytes of data.

The SA buffer is written in NVM with the SAF command, but it can be run time replaced with SSA command without reboot the module.

After the power of or a reboot, the module reload the buffer saved in NVM.

Please note that the "Long Address" of the satellite node is already managed in the mechanism, so is not necessary insert the "Long Address" in the Satellite Answer Field.

RAF return the current AF data buffer used as data for Asynchronous messages.

4.1.17 AA: Acknowledge Answer Field

WAA=val write the static Acknowledge Answer data buffer. It is a 9 bytes of data.

SAA=val write the volatile Acknowledge answer data buffer. It is a 9 bytes of data.

The AA buffer is written in NVM with the WAA command, but it can be run time replaced with SAA command without reboot the module.

After the power of or a reboot, the module reload the buffer saved in NVM.

RAA return the current AA data buffer used as response for data incoming transmission.



4.1.18 FW: Firmware Version

RFW returns the firmware version in the form of 2 char of major version, a '.' char and 2 char of minor version. For example FW version 01.23.

4.1.19 PN: Print Network Tree

RPN prints current network tree.

Following an example of a RPN response of a coordinator in case of network of 8 nodes in 3 level + 1 coordinator:

L	A		/		L	V	L				0	0		0	1		0	2		0	3	
1	2	3	4	5	6	7	8			<	0	0	>									
1	2	3	4	5	6	7	9							0	1							
1	2	3	4	5	6	7	D										0	5				
1	2	3	4	5	6	7	F													0	7	
1	2	3	4	5	6	8	0													0	8	
1	2	3	4	5	6	7	A							0	2							
1	2	3	4	5	6	7	B							0	3							
1	2	3	4	5	6	7	C							0	4							
1	2	3	4	5	6	7	E										0	6				

In this example in level 0 there is only one node: coordinator with long address 12345678 and short address 00; in level 1 there are 4 nodes: long address 12345679 to 1234567C which are child of coordinator and have short address from 01 to 04; in level 2 there are 2 nodes: 1234567D (short address 05) that is child of 12345679 and 1234567E (short address 06) that is child of 1234567C; in level 3 there are 2 nodes: 1234567F (short addr 07) and 12345680 (short addr 08) that are children of 1234567D. Every cell is a char, so empty cells are white spaces. Long and Short Address are coded as ASCII code of every nibble hex value.

Please note that coordinator knows the entire network, while any other nodes know only the network below itself, the father and the coordinator.

Following an example of a RPN response of the node with long address 0x1234567F in case of same network of 8 nodes in 3 level + 1 coordinator:

L	A		/		L	V	L				0	0		0	1		0	2		0	3	
1	2	3	4	5	6	7	8				0	0										
1	2	3	4	5	6	7	D										0	5				
1	2	3	4	5	6	7	F												<	0	7	>



4.1.20 TX: Enqueue a RF Transmission

STX=val En-queues a RF Transmission towards another node.

val has several fields:

BYTE [00 – 03]: Destination Long Address Byte (Byte 0: LSByte)

BYTE 04: Packet Len (max allowed value: 27)

BYTE 05: Data Byte 0

...

BYTE 05 + Packet Len: Data Byte N

if Destination Long Address is equal at 0xFFFFFFFF the Tx is performed in BroadCast mode.

Please note that, due to the AES Encryption limitation (16 bytes data chunk), if Packet Len is less or equal to 11, 11 bytes will be transmitted in RF (5 bytes are reserved for network header); if Packet Len is greater of 11 (but less or equal to 27), 27 bytes will be transmitted.

4.2 Responses to UART commands

If the command is well formatted and is not generated any exception the module return the header of the command precedently sent.

In case of read (static or volatile) commands, the header of the command is followed by a and the requests data.

If an exeption is generated the module return the string ERR= followed by the error code.

In the following table the \r\n characters and the prompt: "\r\n>:\" are not reported.

Code	String	Description
0	<u>ERR=0</u>	Appears when the UART command sent is not well formatted or is unknown
1	<u>ERR=1</u>	Appears when a write command is sent with module not in "configuration mode".
2	<u>ERR=2</u>	Appears when a STX command is sent with a unknown long address as parameter
3	<u>ERR=3</u>	Appears when is impossible execute the requested operation
4	<u>ERR=4</u>	Appears when the maximum number of queuing host tx operations has been reached

4.3 Unsolicited Messages

In addition to the previous commands originating from the host, there are a series of messages that the module can independently send to the host. These messages are called "unsolicited messages" and they always start with the ASCII code U. There are 4 types of "unsolicited messages" Normal - Network - System - Debug. Each type indicates a usage level of the serial where "Normal" represents the minimum usage while "Debug" represents the maximum usage level. Each level includes all lower levels.

"**Normal**" is the verbosity level 0 at which the module calls the host the bare minimum and only to inform it of the activity related to the specific host.

"**Network**" is verbosity level 1. These messages start with UNK char set. Where this level is set the module informs the host about what is happening in the network including activities that do not concern the specific host.

"**System**" is verbosity level 2. These messages start with UST char set. Where this level is set the module informs the host about repetitive system activities also if these not generate a result.

"**Debug**" is verbosity level 3. These messages start with "UDG" char set. Where this level is set, the module in addition to detailing the operations of the lower levels, generates a series of debug messages.

4.3.1 *Normal Type Unsolicited Messages*

The normal type unsolicited messages are sent always because correspond at the verbose level 0

Following there is a table of **Normal type unsolicited messages**.

In the following table the \r\n characters and the prompt: "\r\n>:\" are not reported.

Syntax	Message Description	Cap.
<u>UJR</u> =val	Node Join Report	See Cap. 4.3.1.1
<u>UOR</u> =val	Node Orphan Report	See Cap. 4.3.1.2
<u>URM</u> =val	Host Message received	See Cap. 4.3.1.3
<u>URB</u> =val	Broadcast Message received	See Cap. 4.3.1.4
<u>URA</u> =val	Acknowledge Message received	See Cap. 4.3.1.5
<u>URS</u> =val	Satellite Message received	See Cap. 4.3.1.6
<u>URH</u> =val	Hook Message received	See Cap. 4.3.1.7

4.3.1.1 *UJR – Node Join Report*

When a node join a network, the node itself, the coordinator and all the nodes in the path between node and coordinator send a message to their host as follow.

In case in which the node have joined itself at the network the "long address node" field is filled with its "long address".

UJR=val. val has the following fields:

BYTE [00 – 03]: Long Address Coordinator (Byte 0: LSByte)

BYTE [04 – 07]: Long Address Father (Byte 4: LSByte)

BYTE [08 – 11]: Long Address Node (Byte 8: LSByte)

Long Address Coordinator is the long address of the coordinator of the network;

Long Address Father is the long address of the father of the node;

Long Address Node is the long address of the node who joined the network

4.3.1.2 *UOR – Node Orphan Report*

When a node loose a network, the node itself, the coordinator and all the nodes in the path between node and coordinator send a message to their host as follow.

In case in which the node have loose itself from the network the "long address node" field is filled with its "long address".

UOR=val. val has the following fields:

BYTE [00 – 03]: Long Address Coordinator (Byte 00: LSByte)

BYTE [04 – 07]: Long Address Father (Byte 04: LSByte)

BYTE [08 – 11]: Long Address Node (Byte 08: LSByte)

Long Address Coordinator is the long address of the coordinator of the network;

Long Address Father is the long address of the father of the node;

Long Address Node is the long address of the node who joined the network

4.3.1.3 URM – Host Message Received

When the module receive a data message for the host, it sends a UART message to host as follow.

URM=val. val has the following fields:

BYTE [00 – 03]	Source Long Address	(Byte 00: LSByte)
BYTE [04]	Packet Len	(max allowed value: 26)
BYTE [05]	Data Byte 0	
...		
BYTE [05+N]	Data Byte N	

4.3.1.4 URB – Broadcast Message Received

When the module receive a data message for all the nodes of the network, it sends a UART message to host as follow.

URB=val. val has the following fields:

BYTE [00 – 03]	Source Long Address	(Byte 00: LSByte)
BYTE [04]	Packet Len	(max allowed value: 26)
BYTE [05]	Data Byte 0	
...		
BYTE [05+N]	Data Byte N	

4.3.1.5 URA – Acknowledge Answer Received

When the module send a data message, and the node destination respond with an acknowledge message that contain data, it sends a UART message to host as follow.

URA=val. val has the following fields:

BYTE [00 – 03]	Replied node's Long Address	(Byte 00: LSByte)
BYTE [04 – 13]	Acknowledge data	

4.3.1.6 URS – Satellite Answer Received

When the module send an asynchronous hook data message, and a satellite node respond with an message that contain data, it sends a UART message to host as follow.

URS=val. val has the following fields:

BYTE [00 – 03]	Satellite node's Long Address	(Byte 00: LSByte)
BYTE [04 – 12]	Satellite answer data	

4.3.1.7 URH – Asynchronous Hook message Received

When a module configured as satellite node found an asynchronous message, after send a satellite answer, it sends a UART message to host as follow.

URH=val. val has the following fields:

BYTE [00 – 03]	Satellite node's Long Address	(Byte 00: LSByte)
BYTE [04 – 28]	Asynchronous hook message data	

4.3.2 Network type Unsolicited Messages

The Network type unsolicited messages are sent if the verbose level is set to at least 1

Following there is a table of **Network type unsolicited messages**.

In the following table the \r\n characters and the prompt: "\r\n>:\" are not reported.

Syntax	Message Description	Cap.
<u>UNK=APP RIN</u>	Application Re-initialization	See Cap. 4.3.2.1
<u>UNK=BCN FND</u>	Father beacon message	See Cap. 4.3.2.2
<u>UNK=CHK SNC</u>	Check synchronization performed	See Cap. 4.3.2.3
<u>UNK=KEY CHN</u>	Network key type Changed	See Cap. 4.3.2.4
<u>UNK=PRM REC</u> -val	Network Parameters Received	See Cap. 4.3.2.5
<u>UNK=CNF ACK</u> -val	Join procedure confirmation	See Cap. 4.3.2.6
<u>UNK=LVE ACK</u> -val	Received a live acknowledge	
<u>UNK=MEX ACK</u> -val	Received an acknowledge	
<u>UNK=LVE SND</u> -val	Live request answered	See Cap. 4.3.2.7
<u>UNK=LVE STA</u> -val	Enqueue a live request to a node	
<u>UNK=ALT INS</u> -val	Alternative join request received	See Cap. 4.3.2.8
<u>UNK=ALT RET</u> -val	Alternative join retry in progress	
<u>UNK=ALT RDY</u> -val	Alternative join ready for join	
<u>UNK=PRM NEP</u> -val	Received parameters for a node not directly son	
<u>UNK=STR</u> -v1_v2	Start BITMESH stack Network	See Cap. 4.3.2.9
<u>UNK=TXM ERR</u> -v1_v2	No answer from a node	See Cap. 4.3.2.10
<u>UNK=TXM CCA</u> -v1_v2	Channel busy during a message transmission	
<u>UNK=TXM RQF</u> -v1_v2	Radio queue full	
<u>UNK=BCN CHK</u> -v1_v2	Problem during receiving a beacon from the father node	See Cap. 4.3.2.11
<u>UNK=BCN CCA</u> -v1	Channel busy during sending the beacon message	
<u>UNK=BCN ERR</u> -v1	Impossible send beacon message for channel busy	

4.3.2.1 Application Re-initialization

When a node perform an application re-initialization procedure, the module send an UART message to host as follow:

UNK=APP RIN

4.3.2.2 Father Beacon Found

When a node ROUTER or END is looking for a network and find a good quality node father to start the join procedure, it send an UART message to host as follow:

UNK=BCN FND

4.3.2.3 Check synchronization performed

When a node ROUTER or END is already joined in a network and it lost a beacon message, perform a procedure in order to define if it is synchronized with the net or not. if this procedure produce a positive result, the module send an UART message to the host as follow:

UNK=CHK SNC

4.3.2.4 Network key type Changed

When the COORDINATOR network start a "key change procedure" is necessary wait the conclusion of the "SUPER SLOT FRAME" in progress. after that the module send an UART message to the host as follow:

UNK=KEY CHN

4.3.2.5 Network Parameters Received

When a node ROUTER or END during the join procedure receive the network parameters, the module send an UART message to the host as follow:

UNK=PRM REC-val. val has the following fields:

BYTE [00 – 03] *Father 's long address*

4.3.2.6 Network Acknowledge Received

When a father node confirm the join in the network of a new node son the module send an UART message to the host as follow:

UNK=CNF ACK-val

When a node send a "Life State Supervision" request and the supervised node answer, the module send an UART message to the host as follow:

UNK=LVE ACK-val

When a node send a network message to other node and this answer, the module send an UART message to the host as follow:

UNK=MEX ACK-val

In all three cases, the val field indicate the Long address of the node that answered at the module. And it is composed as follow;

BYTE [00 – 03] *long address of the node in question*

4.3.2.7 Life State Request

When a node enqueue a "Life State Supervision" transmission operation, the module send an UART message to the host as follow:

UNK=LVE STA-val

When a node answer at an a "Life State Supervision" the module send an UART message to the host as follow:

UNK=LVE SND-val

In both cases, the val field indicate the Long address of the node that answered at the module. And it is composed as follow;

BYTE [00 – 03] *long address of the node in question*

4.3.2.8 Alternative join information

When a node ROUTER receive a join request from a not joined node, and it doesn't have the parameters the module perform a "network parameters request" and send an UART message to the host as follow:

UNK=ALT INS-val val: not joined router long address

When a node router after have performed a network parameters request haven't received them, the module perform again the request and send an UART message as follow:

UNK=ALT RET-val val: not joined router long address

When a node ROUTER receive a join request from a not joined node, and it have the parameters the module start the join procedures and send an UART message to the host as follow:

UNK=ALT RDY-val val: not joined router long address

When a node ROUTER receive the network parameters that it have request, the module send to the host an UART message as follow:

UNK=PRM NEP-val val: not joined router long address

4.3.2.9 BITMESH start Network Stack

When a node correctly configured start its network procedures, the module send an UART message to the host as follow:

UNK=STR-v1-v2 v1: this node long address v2: node type of this node

4.3.2.10 Data Transmission Error Unsolicited

When a node tried to send a message to an other node and the destination doesn't answer with the acknowledge, the module if it can enqueue a retry transmission and send an UART message to the host as follow:

UNK=TXM ERR-v1-v2 v1: not joined router long address v2: number of retry remains

When a node tried to send a message to an other node and it can't transmit for channel busy, the module enqueue a retry transmission and send an UART message to the host as follow:

UNK=TXM CCA-v1-v2 v1: not joined router long address v2: number of retry remains

When a destination node can't transport a message from this node to an other because it have the radio queue full, it send an UART message to the host as follow:

UNK=TXM RQF-v1-v2 v1: not joined router long address v2: number of retry remains

4.3.2.11 Beacon Transmission Error Unsolicited

When a node loose a beacon message, check if it continue to be synchronized with the network and send an UART message to the host as follow:

UNK=BCN CHK-v1-v2 v1: father long address v2: number of check sync remains

When a node ROUTER or COORDINATOR tried to send a beacon message and it can't transmit for channel busy, the module enqueue a retry transmission and send an UART message to the host as follow:

UNK=BCN CCA-v1 v1: number of retry available

When a node ROUTER or COORDINATOR can't to send a beacon message for channel busy and reach the max number of retry, the beacon won't be transmitted and the module send an UART message to the host as follow:

UNK=BCN ERR-v1 v1: is always at 0 value

4.3.3 System type Unsolicited Messages

The System type unsolicited messages are sent if the verbose level is set to at least 2

Following there is a table of **System type unsolicited messages**.

In the following table the \r\n characters and the prompt: "\r\n>:" are not reported.

Syntax	Message Description	Cap.
<u>UST=BTX-NN-FFF-SSS</u>	Sent a beacon message	See Cap. 4.3.3.1
<u>UST=BRX-val-NN-FFF-SSS</u>	Received beacon message	See Cap. 4.3.3.2
<u>UST=HTX-NN-FFF-SSS</u>	Sent an hook message	See Cap. 4.3.3.3

4.3.3.1 Sent Beacon Message

When a node ROUTER or COORDINATOR send a beacon message, the module send an UART message to the host as follow:

UST=BTX-NN-FFF-SSS

Where:

NN-> (2 char 00-49) slot number in which beacon was sent
 FFF-> (3 char 000-255) frame number in which beacon was sent
 SSS-> (3 char 000-255) superframe number in which beacon was sent

4.3.3.2 Received Beacon Message

When a node receive a beacon message, the module send an UART message to the host as follow:

UST=BRX-val-NN-FFF-SSS

Where:

val-> BYTE[0-3] long address of node who sent beacon
 NN-> (2 char 00-49) slot number in which beacon was received
 FFF-> (3 char 000-255) frame number in which beacon was received
 SSS-> (3 char 000-255) superframe number in which beacon was received

4.3.3.3 Sent an Hook message

When a node with asynchronous slot associated send an asynchronous Hook message, the module send an UART message to the host as follow:

UST=HTX-NN-FFF-SSS

Where:

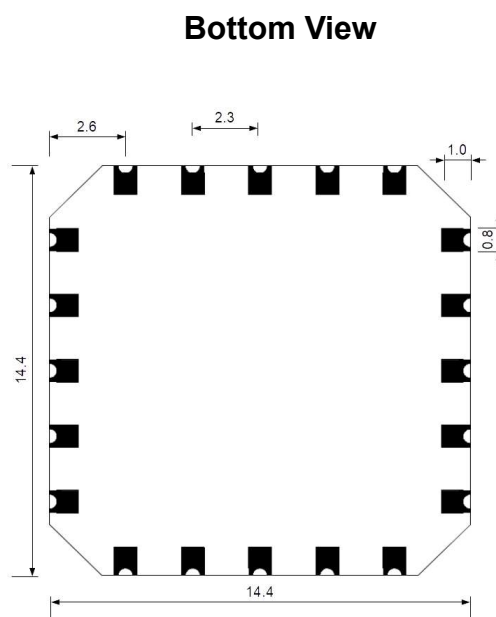
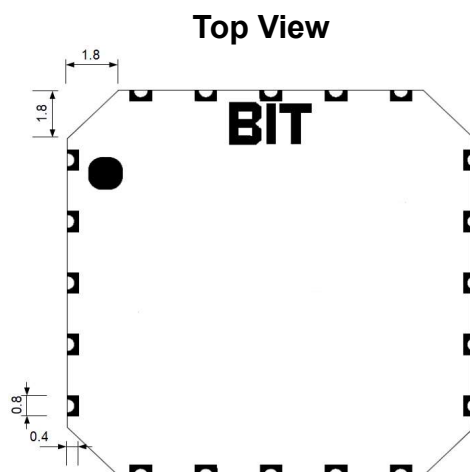
NN-> (2 char 00-49) slot number in which hook is send
 FFF-> (3 char 000-255) frame number in which hook is send
 SSS-> (3 char 000-255) superframe number in which hook is send

4.3.4 Debug type Unsolicited Messages

The Debug type unsolicited messages are sent if the verbose level is set to 3

In addition to level 2 more unsolicited messages are generated for debug purpose.

5 Mechanical Dimension (mm)



6 Soldering profile Recommendation

JEDEC standard IPC/JEDEC J-STD-020D.1 (page 7 and 8), Pb-Free Assembly is recommended.

The standard requires that the heat dissipated in the "surroundings" on the PCB is taken into account. The peak temperature should be adjusted so that it is within the window specified in the standard for the actual motherboard.

Aperture for paste stencil is normally areal-reduced by 20-35%, please consult your production facility for best experience aperture reduction. Nominal stencil thickness of 0.1-0.12 mm recommended.



7 *Packaging Information*

Orderable device	Status	Package Type	Pins	Package Qty	Eco Plan	MSL Peak Temp
BIT868MN	ACTIVE	MLF	20	96	Green (RoHS & no Sb/Br)	Level-2-260C-1 YEAR

8 *General Information*

8.1 *Disclaimer*

B.I.T. srl believes the information contained herein is correct and accurate at the time of this printing. However, B.I.T. srl reserves the right to make changes to this product without notice. B.I.T. srl does not assume any responsibility for the use of the described product; neither does it convey any license under its patent rights, or the rights of others. The latest updates are available at the BIT website or by contacting BIT directly.

As far as possible, major changes of product specifications and functionality, will be stated in product specific Errata Notes published at the BIT website. Customers are encouraged to sign up to the Developers Newsletter for the most recent updates on products and support tools.

Compliance with regulations is dependent on complete system performance. It is the customer's responsibility to ensure that the system complies with regulations.

8.2 *Life Support Policy*

This BIT product is not designed for use in life support appliances, devices, or other systems where malfunction can reasonably be expected to result in significant personal injury to the user, or as a critical component in any life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. B.I.T. srl customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify B.I.T. srl for any damages resulting from any improper use or sale.



Bit is Italian Technology S.r.l.
Via Luigi Settembrini, 29 – 20020 Lainate (MI)
Tel.: (+39) 02 30465311, fax: (+39) 0230465396
<http://www.bit.it/> info@bit.it



9 ***Revision list***

Revision	Data	Description
01.00	2020-06-13	<ul style="list-style-type: none">• First Release
01.01	2020-09-20	<ul style="list-style-type: none">• Bug Fixing• Added the
01.01a	2020-09-23	<ul style="list-style-type: none">• Added some specification in the Description Chapter• Add a broadcast message after a SKT command.
01.02	2020-10-07	<ul style="list-style-type: none">• Fixed Bug in the “UNK=TXM CCA” Long Address message