AX-ENOCEAN-MOD

AX-EnOcean Gateway Ministamp / Antstamp

Ultra−low Power, EnOcean® Compliant Modules

ON Semiconductor®

www.onsemi.com

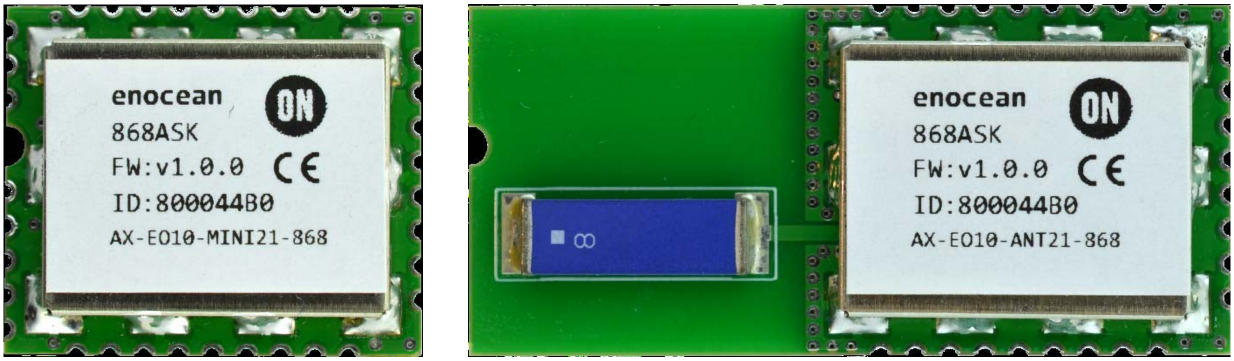


Figure 1. AX−ENOCEAN Ministamp/Antstamp Modules

OVERVIEW

The AX−EnOcean modules are ultra−low power module solutions for communication with certified EnOcean products. The AX−ENOCEAN modules connect to the customer application using a logic level RS232 UART. The EnOcean Serial Protocol 3 is used to send and receive telegrams as well as configure parameters.

The AX−EnOcean module comes in two flavors

- AX−EnOcean Ministamp with 50 Ω Antenna Port
- AX−EnOcean Antstamp with On−board −5 dBi Chip Antenna

To facilitate development, we also offer a USB−stick with integrated USB−to−serial converter for operation with a host PC that is compatible to EnOcean’s USB300.

The modules act as a gateway between a host (such as another microcontroller or a PC) and other certified EnOcean products (for example another AX−EnOcean product):

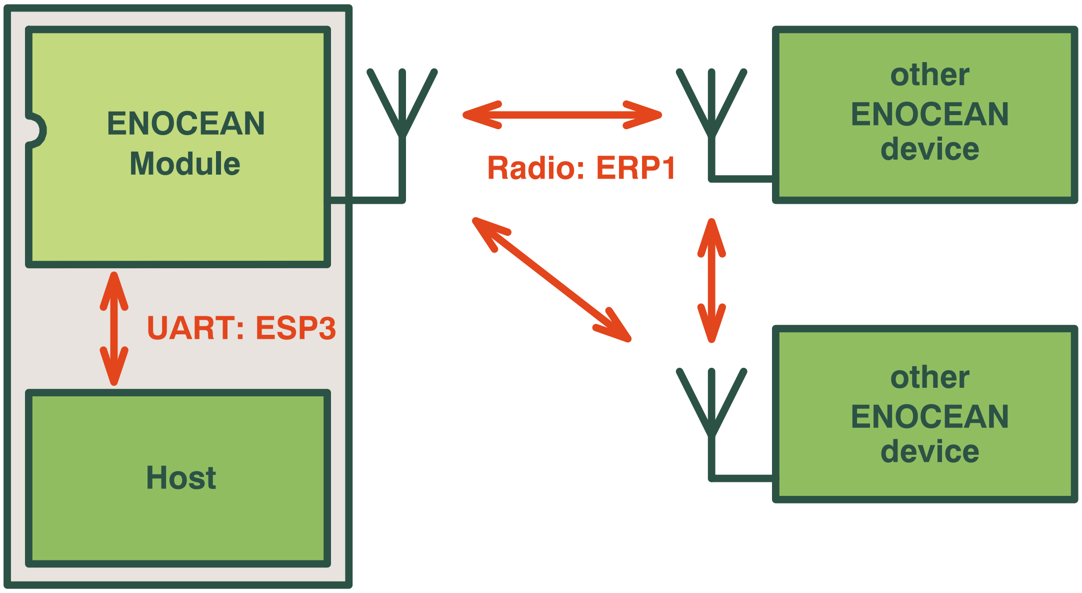


Figure 2.

EnOcean devices talk with each other over the EnOcean Radio Protocol 1 (ERP1, specified in ISO/IEC 14543−3−10). The communication between the host and an EnOcean module is based on the Command interface: ESP3, a binary serial protocol that is used to send and receive radio messages (called telegrams) as well as to read and write various parameters.

Functionality and Ecosystem

- EnOcean transmission and reception functionality controlled by ESP3 commands
- The AX−EnOcean modules are part of a whole development and product ecosystem available from ON Semiconductor for any EnOcean requirement. Other parts of the ecosystem include
  - Ready to go AX−EnOcean development kit with fully functional AX−EnOcean module
  - ON Semiconductor EnOcean ICs
  - AX−EnOcean−API IC for customers wishing to write their own application software based on the ON Semiconductor EnOcean Library

General Features

- 18.2 x 22 x 3 mm^3 without Chip Antenna
  18.2 x 39.7 x 3 mm^3 with Chip Antenna
- Supply Range from 1.8 V to 3.3 V
- −20°C to 65°C
- 15 GPIO Pins, controllable through Serial Interface
  - 2 GPIO pins with selectable voltage measure functionality, differential (1 V or 10 V range) or single ended (1 V range) with 10 bit resolution
  - 2 GPIO pins with selectable ΣΔ DAC or clock output functionality
  - 3 GPIO pins selectable as SPI master interface
- Temperature Sensor
- Supply Voltage Measurements
- Compatible with EnOcean’s TCM310

Power Consumption

Ultra Low Power Consumption:

- Sleep current: 1.5 μA
- Continuous Radio Reception at 868.3 MHz: 13.3 mA
- Continuous Radio Transmission at 868.3 MHz and 7 dBm: 24 mA

High Performance Narrow−band EnOcean Transceiver

- 125 kbps Data Rate
- 7 dBm Peak Output Power = 4 dBm RMS
- −96 dBm Sensitivity

Fully Certified

- EnOcean Certified (ISO/IEC 14543−3−10)
- EN 300 220

About the EnOcean Technology

EnOcean is an energy harvesting wireless standard for building automation and industrial automation. By using a high data rate and small packets, very short transmission times and ultra−low power consumption can be achieved.

The use of short transmission times and redundant packets ensures a high reliability in globally available ISM bands (license−free frequency bands).

SPECIFICATIONS

SUPPLY

Symbol	Description	Condition	Min	Typ	Max	Units
T_{AMB}	Operational ambient temperature		−20	27	65	°C
VDD	Supply voltage		1.8	3.0	3.3	V
I_{SLP}	Sleep current	Code 01: CO_WR_SLEEP 55 00 05 00 05 DB 01 00 00 03 E8 CB		1.5		μA
I_{STDBY}	Standby current	Code 128: CO_WR_RX_MODE Receiver disabled 55 00 02 00 05 CD 80 00 B6		710		μA
I_{RX}	Continuous receive current	Code 128: CO_WR_RX_MODE Receiver enabled 55 00 02 00 05 CD 80 01 B1		13.3		mA
I_{TX}	Modulated Transmitter Current	Code 129: CO_WR_TX_TEST P_{OUT} = 7 dBm 55 00 02 00 05 CD 81 03 AA		23.8		mA
Q_{4BS}	Charge to send a 4BS packet	ESP3: Packet Type 1: RADIO_ERP1 55 00 0A 07 01 EB A5 12 34 56 78 00 00 00 00 00 03 FF FF FF FF FF 00 67		42		mC

1. If VDD_IO ramps cannot be guaranteed, an external reset circuit is recommended, see the AX8052 Application Note: Power On Reset

Typical Current Waveform

Shown here is a 4BS−packet (4 bytes of data payload) with three subtelegrams

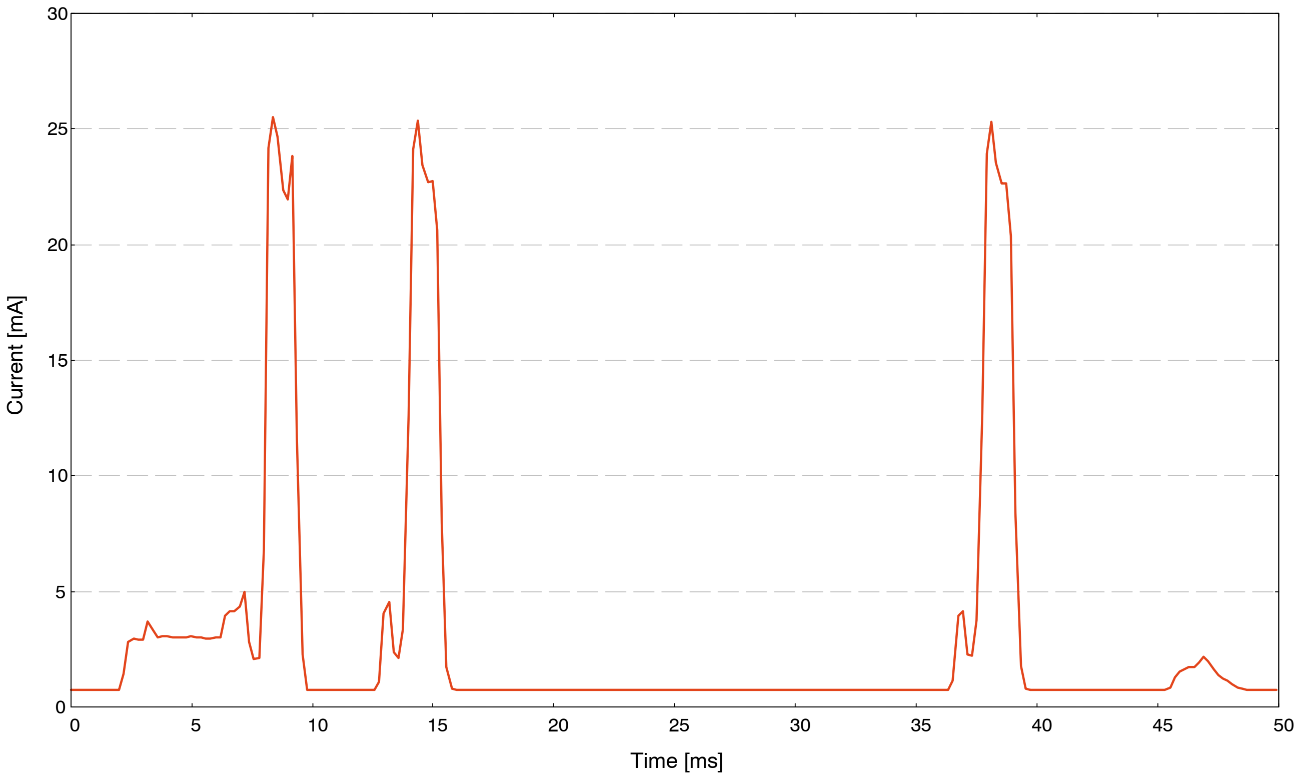


Figure 3.

LOGIC

Symbol	Description	Condition	Min	Typ	Max	Units
Digital Inputs						
V_{T+}	Schmitt trigger low to high threshold point	VDD = 3.3 V		1.55		V
V_{T-}	Schmitt trigger high to low threshold point			1.25		V
V_{IL}	Input voltage, low				0.8	V
V_{IH}	Input voltage, high		2.0			V
V_{IPA}	Input voltage range, GPIO13−14, UARTRX		−0.5		VDD	V
V_{IPBC}	Input voltage range, RESET_N, GPIO0−12		−0.5		5.5	V
I_L	Input leakage current		−10		10	μA
R_{PU}	Programmable Pull−Up Resistance			65		kΩ
Digital Outputs						
I_{OH}	Output Current, high	V_{OH} = 2.4 V	8			mA
I_{OL}	Output Current, low	V_{OL} = 0.4 V	8			mA
I_{OZ}	Tri−state output leakage current		−10		10	μA

TRANSMITTER

Symbol	Description	Condition	Min	Typ	Max	Units
SBR	Signal bit rate			125		kbps
$f_{carrier}$	Carrier frequency			868.3		MHz
PTX_{peak}	Transmitter output power, ASK high level (peak)			7		dBm
PTX_{RMS}	Transmitter output power, RMS			4		dBm
dTX_{temp}	Transmitter power variation vs. temperature	−40°C to +85°C		±0.5		dB
dTX_{Vdd}	Transmitter power variation vs. VDD	1.8 to 3.3 V		±0.5		dB

Bit−Shaping

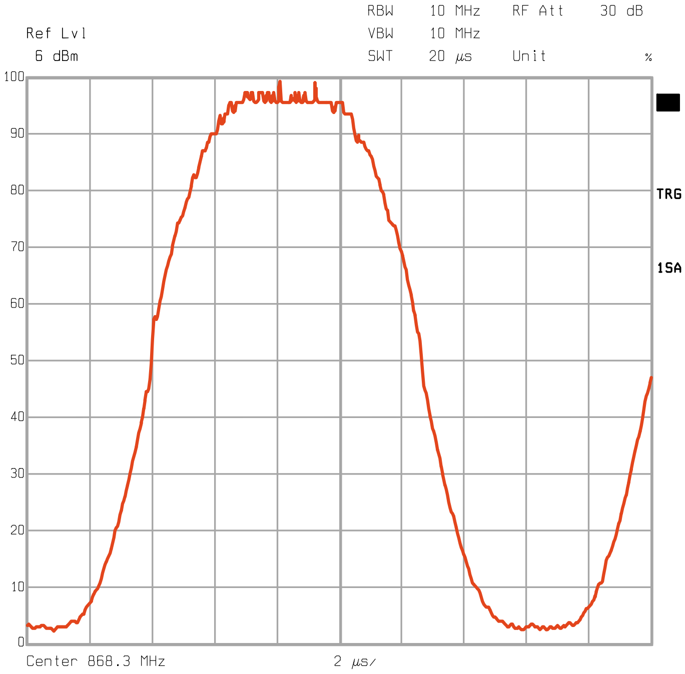


Figure 4. Bit−Shaping

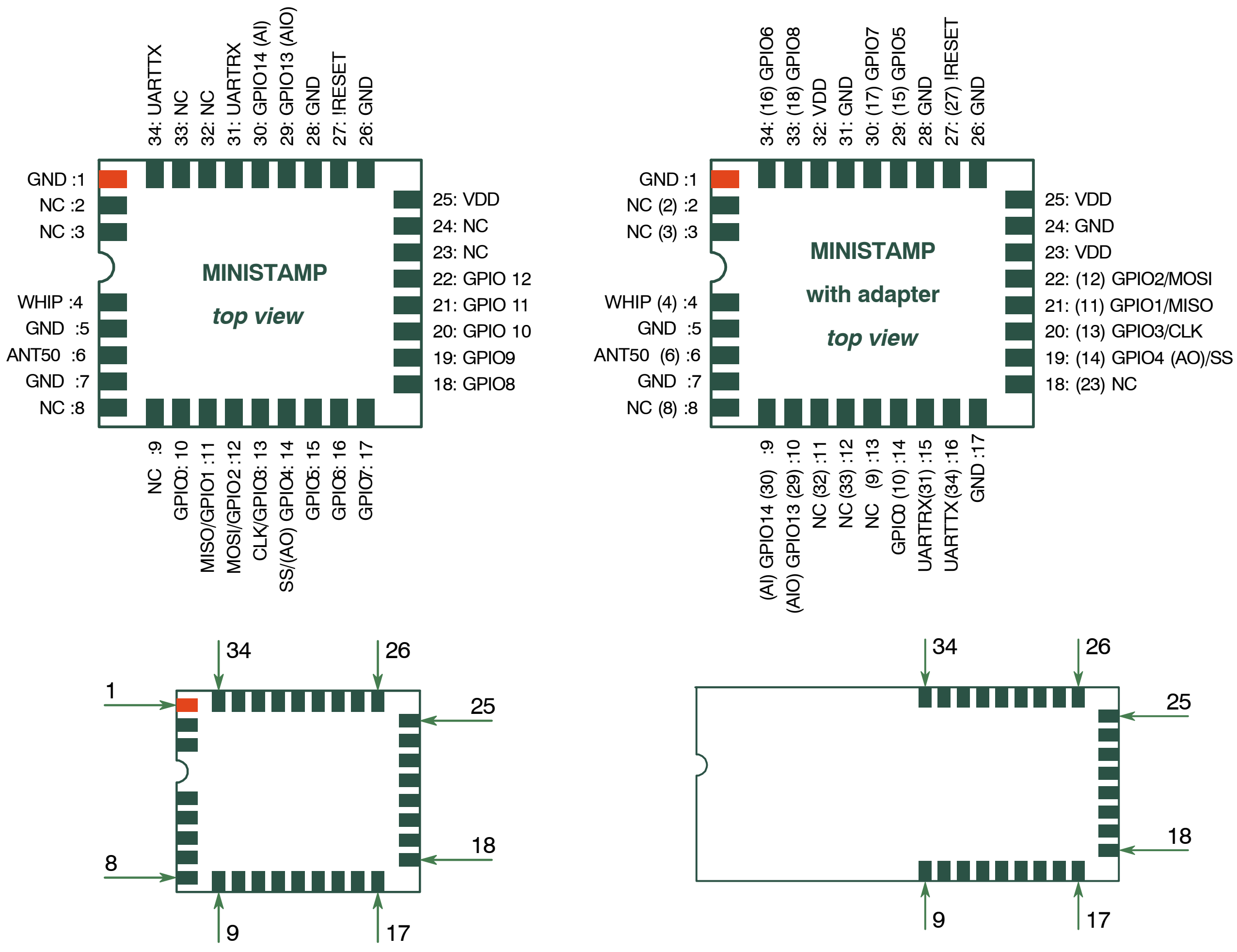
RECEIVER

Symbol	Description	Condition	Min	Typ	Max	Units
SBR	Signal bit rate			125		kbps
$f_{carrier}$	Carrier frequency			868.3		MHz
IS	Sensitivity	4BS, PER < 10% 3 subtelegrams		−96		dBm
BLK_2	Blocking at ± 2 MHz offset	Received signal is 3 dB above typical sensitivity limit (PER < 10%) and the blocker is a continuous wave		52		dB
BLK_{10}	Blocking at ± 10 MHz offset			65		dB
f_{tol}	RX frequency tolerance	4BS, PER < 10% 3 subtelegrams		± 35		kHz
AR_{tol}	RX high state amplitude to low state amplitude ratio tolerance		16			dB
BR_{tol}	RX bitrate tolerance			± 0.25		%
PRX_{max}	Maximal receiver input power, RMS			−25		dBm

Table 1. ADC / TEMPERATURE SENSOR

Symbol	Description	Condition	Min	Typ	Max	Units
ADCRES	ADC resolution			10		bit
V_{ADCREF}	ADC reference voltage		0.95	1	1.05	V
Z_{ADC00}	Input capacitance				2.5	pF
DNL	Differential nonlinearity			± 1		LSB
INL	Integral nonlinearity			± 1		LSB
OFF	Offset			3		LSB
GAIN_ERR	Gain error			0.8		%
ADC in Differential Mode						
V_{ABS_DIFF}	Absolute voltages & common mode voltage in differential mode at each input		0		VDD	V
V_{FS_DIFF01}	Full swing input for differential signals	Gain x0.1	−5		5	V
V_{FS_DIFF1}		Gain x1	−500		500	mV
V_{FS_DIFF10}		Gain x10	−50		50	mV
ADC in Single Ended Mode						
V_{MID_SE}	Mid code input voltage in single ended mode			0.5		V
V_{IN_SE00}	Input voltage in single ended mode		0		VDD	V
V_{FS_SE01}	Full swing input for single ended signals	Gain x1	0		1	V
Temperature Sensor						
T_{RNG}	Temperature range		−40		85	°C
T_{ERR_CAL}	Temperature error			± 2		°C

PINOUT



MINISTAMP

ANTSTAMP

Figure 5. Pinout Drawings (Top View)

PIN FUNCTION DESCRIPTION

Pin	Pin with Adapter	Name	Function
1*	1*	gnd	Ground
2*	2*	nc	Do not connect
3*	3*	nc	
4*	4*	WHIP	Whip antenna port
5*	5*	gnd	Ground
6*	6*	ant 50 Ω	50 Ω antenna port
7*	7*	gnd	Ground
8*	8*	nc	Do not connect
9	13	NC	
10	14	GPIO0	GPIO 0
11	21	GPIO1 / SPI MISO	GPIO 1, SPI data input (MISO)
12	22	GPIO2 / SPI MOSI	GPIO 2, SPI data output (MOSI)
13	20	GPIO3 / SPI CLOCK	GPIO 3, SPI clock output

PIN FUNCTION DESCRIPTION

Pin	Pin with Adapter	Name	Function
14	19	GPIO4 / SPI SS	GPIO 4, analog output, SPI chip select output
15	29	GPIO5	GPIO 5
16	34	GPIO6	GPIO 6
17	30	GPIO7	GPIO 7
18	33	GPIO8	GPIO 8
19	−	GPIO9	GPIO 9
20	−	GPIO10	GPIO 10
21	−	GPIO11	GPIO 11
22	−	GPIO12	GPIO 12
23	18	NC	Do not connect
24	−	NC	
25	25	VDD	Power supply (1.8 – 3.3 V)
26	26	GND	Ground
27	27	RESET_N	Optional reset (active low). Do not connect the pin if not used, has internal pull−up.
28	28	GND	Ground
29	10	GPIO13	GPIO 13, analog input/output
30	9	GPIO14	GPIO 14, analog input
31	15	UARTRX	UART used to communicate with the module (input to module).
32	11	NC	Do not connect
33	12	NC	
34	16	UARTTX	UART used to communicate with the module (output of module).
−	17	GND	Ground
−	23	VCC	Power supply (1.8 – 3.3 V)
−	24	GND	Ground
−	32	GND	
−	33	VDD	Power supply (1.8 – 3.3 V)

*MINISTAMP only, the ANTSTAMP does not have these pins.

1. All digital pins are Schmitt trigger inputs, digital input and output levels are VCMOS/LVTTL compatible.
2. Pins 29 (GPIO13), 30 (GPIO14), 31 (UARTRX) and 34 (UARTTX) must not be driven above VDD; all other digital inputs are 5 V tolerant.
3. All VDD and ground pins must be connected. Pins marked ‘NC’ should not be connected and must be left floating.
4. GPIO pins are digital inputs with pull−up upon startup.

TCM310 Adapter

Even though the MINISTAMP uses the same footprint as EnOcean’s TCM310 module, they are not pin compatible. For this purpose, we provide an adapter PCB that maps the MINISTAMPS pins to the same location as on the TCM310. This mainly concerns the location of the UART interface: On the normal MINISTAMP and ANTSTAMP the serial receive and transmit lines lie on pin 31 and 34 respectively, with the adapter these pins are mapped to pin 15 and 16. The full mapping is specified in the table above and the graphic shows the pins (with original pin numbers in brackets) at their new positions.

Differences to EnOcean’s TCM310

While the AX−ENOCEAN modules strive to be as compatible as possible to EnOcean’s TCM310 module, there are a few differences:

- The MINISTAMP does not have a RF_WHIP output to connect a whip antenna, only a 50 Ω connection is provided to save the losses of a whip antenna output. If an onboard antenna is desired, we recommend using the ANTSTAMP, which incorporates a chip antenna.
- There is no READ_EN pin to enable readout of the firmware’s configuration area.
- There is no digital supply voltage regulator output (DVDD). However, the module can be driven with voltages from 1.8 to 3.3 V.

DIMENSIONS

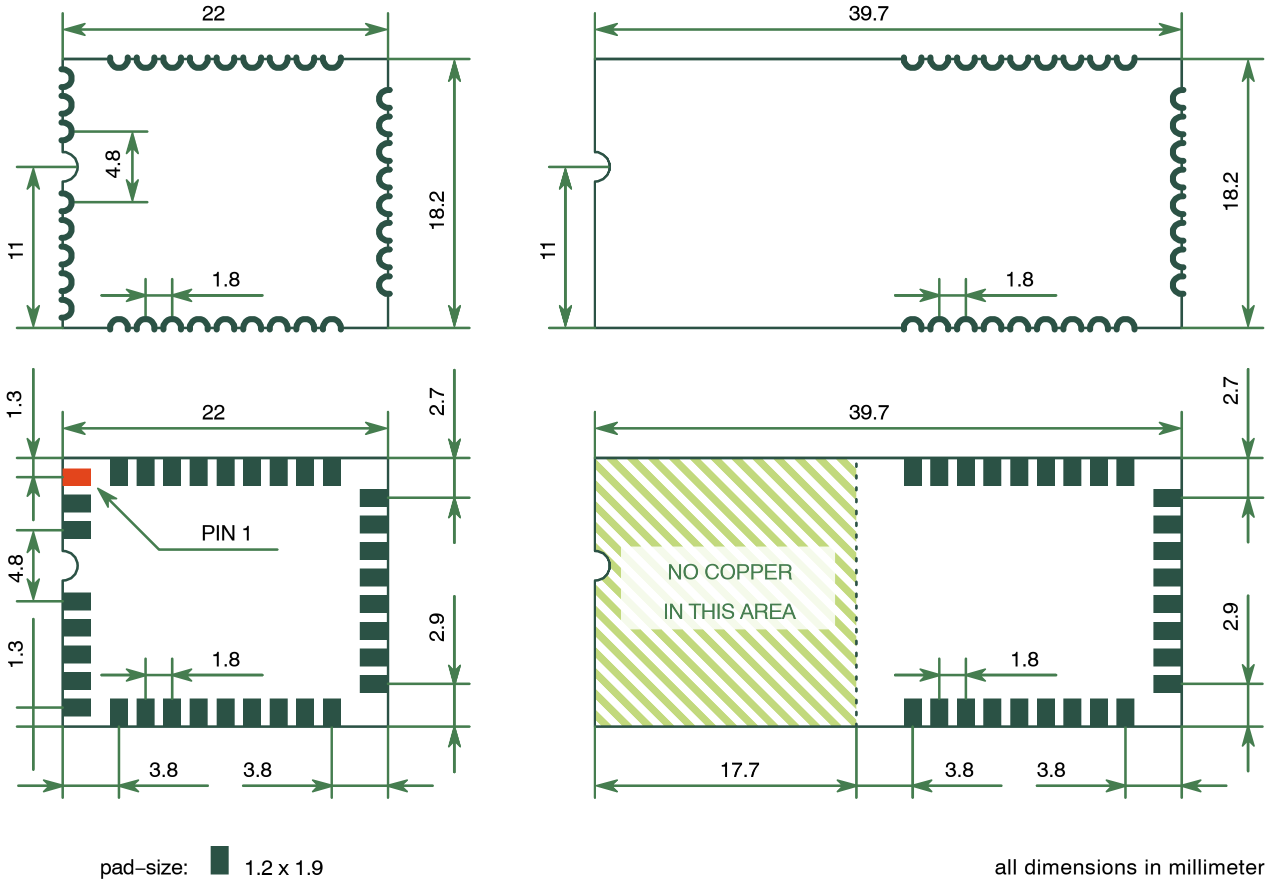


Figure 6. Dimensions (mm)

The area under the modules must be covered in solder stop. To guarantee good antenna performance with the AX−ENOCEAN ANTSTAMP, the left part of the module must remain free from any copper (i.e. no ground plane and no traces).

FUNCTIONALITY

The AX−ENOCEAN modules are designed as gateways between the EnOcean Radio Protocol 1 (ERP1) and the EnOcean Serial Protocol 3 (ESP3). Received radio packets are forwarded over the serial link as ESP3: Packet Type 1: RADIO_ERP1 serial packet, while sending such a serial packet to the device results in an ERP1 radio packet to be transmitted over the air.

While the functionality of a gateway is straightforward, AX−ENOCEAN modules have additional features, such as repeating or filtering packets, which complicate the data flow.

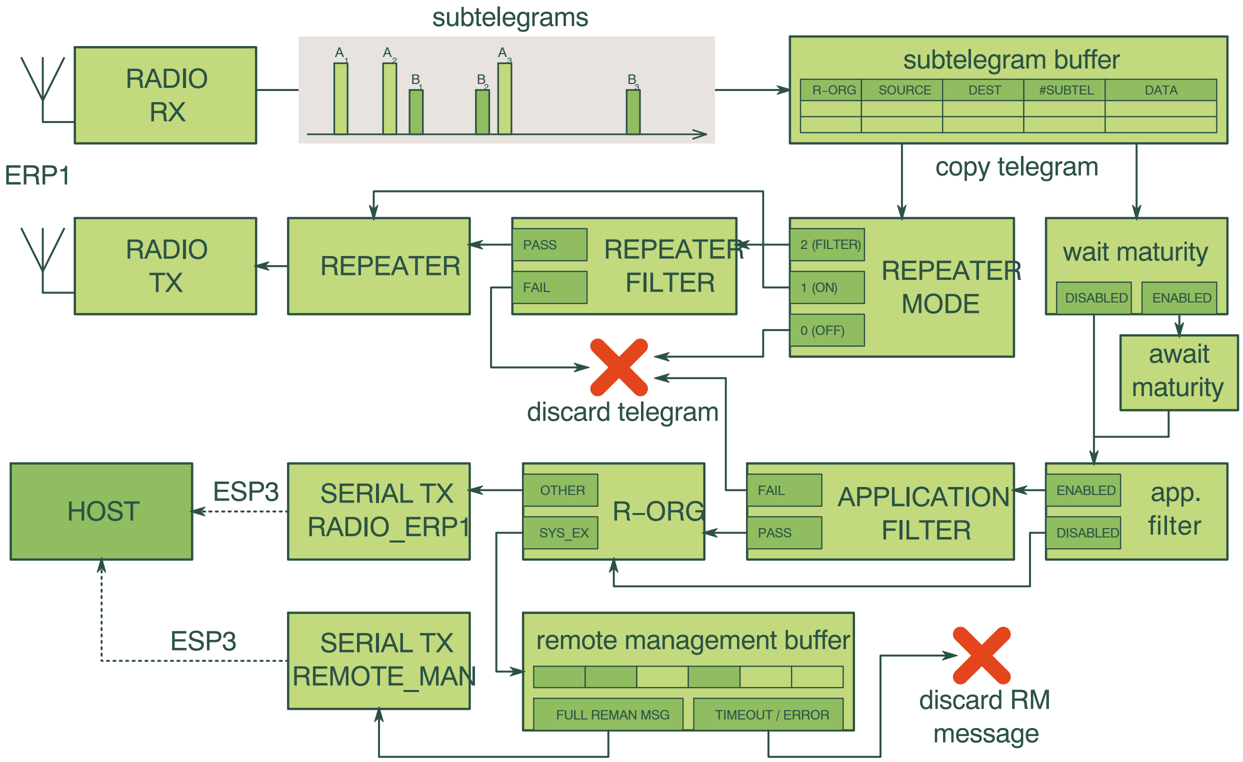


Figure 7.

EnOcean radio packets are always repeated three times for increased reliability. These are cached for further processing. If enabled and it matches possible filters, each received radio packet is repeated. Radio packets are also forwarded on the serial link to a connected host (again, if they pass the filters first). Remote management messages are treated a bit differently, as they can consist of multiple radio telegrams.

R−ORG

Each EnOcean telegram has a type, called R−ORG (**R**adio−Telegram **Org**anization). This type specifies the general format and purpose of a telegram. The AX−ENOCEAN gateway devices do not support all types and will just forward any telegram with unknown R−ORG. If this is not desired, add an R−ORG−filter.

Note that the R−ORG does not specify the format of the actual payload data, merely how much of it there is and which purpose it serves. The payload data encoding is done by the host and specified in the EnOcean Equipment Profiles.

Table 2.

R−ORG	Name	Description	Supported?
0xF6	RPS	Repeated Switch Communication	Yes
0xD5	1BS	1 Byte Communication	Yes
0xA5	4BS	4 Byte Communication	Yes
0xD2	VLD	Variable Length Data	Yes
0xD1	MSC	Manufacturer Specific Communication	No
0xA6	ADT	Addressing Destination Telegram	Yes

Table 2.

R−ORG	Name	Description	Supported?
0xC6	SM_LRN_REQ	Smart Ack Learn Request	No
0xC7	SM_LRN_ANS	Smart Ack Learn Answer	No
0xA7	SM_REC	Smart Ack Reclaim	No
0xC5	SYS_EX	Remote Management	Yes
0x30	SEC	Secure Telegram	No
0x31	SEC_ENCAPS	Secure Telegram with R−ORG encapsulation	No

EnOcean Equipment Profiles

The EnOcean Radio Protocol and EnOcean Serial Protocol define physical, link and network layer to be used by EnOcean protocols. However, the application layer – i.e. the encoding of the actual data, such as temperature measurements or button presses – is defined by the EnOcean Equipment Profiles, with each profile targeting a specific use case (such as a temperature sensor or a light−switch).

As the AX−ENOCEAN gateway modules just forwards telegrams in a transparent fashion, the host has to handle the actual payload data while the gateway module does not touch the data at all.

For a more detailed description and a full list of all profiles, consult the official EnOcean Equipment Profiles specification available at:
http://www.enocean−alliance.org/eep/

Addressing

Each EnOcean device has its own, factory−defined and globally unique 32−bit device ID (also called chip ID). Every radio packet contains this address to identify the source of a packet. When transmitting a packet, the source ID `0x0000'0000` can be used which the device will automatically replace with its device ID.

Usually, radio packets do not contain a destination address and are treated as broadcast messages. On the serial interface, such packets have a destination address of `0xFFFF'FFFF` (the explicit broadcast address). When transmitting a packet, a destination address can be specified. However, keep in mind that an addressed radio packet can still be received by anyone and does not provide any security.

Important: AX−ENOCEAN gateway modules will receive and forward all telegrams by default, even if they are addressed to a different device. To receive only telegrams with specific addresses, set up a corresponding Filter.

An EnOcean module usually transmits packets using its device ID as sender address. However, for some purposes it can be useful to change the ID of a device (such as not having to teach−in a replacement device to every receiver). This is where the base ID comes in: It defines a range of 128 additional IDs that can be used when transmitting a radio packet.

The Code 07: CO_WR_IDBASE command changes the base ID to anything in the range of `0xFF80'0000` to `0xFFFF'FFFE` (aligned to 7bit, i.e. the last seven bits are 0). When using the ESP3: Packet Type 1: RADIO_ERP1 command (or any other command which will transmit a radio packet, such as an ESP3: Packet Type 7: REMOTE MANAGEMENT command), the allowed sender IDs are:

- CHIP_ID (use CO_RD_VERSION to read the device ID or just use `0x0000'0000`),
- BASE_ID, BASE_ID+1, BASE_ID+2, ..., BASE_ID+3

For example, possible base IDs include:

...
`0xFFFF'4680`
`0xFFFF'4700`
`0xFFFF'4780`
`0xFFFF'4800`
`0xFFFF'4880`
`0xFFFF'4900`
...

Therefore, when using the base ID 0xFFFF_4680, the possible sender IDs are:

`0xFFFF'4680`
`0xFFFF'4681`
`0xFFFF'4682`
...
`0xFFFF'46FE`
`0xFFFF'46FF`

Use the Code 08: CO_RD_IDBASE and Code 07: CO_WR_IDBASE commands to read and write the base ID of an EnOcean device.

Wait Maturity

EnOcean radio packets are usually transmitted three times (with semi−random delays) to increase reliability. In some cases devices send only a single packet (mainly if energy consumption is critical), while the installation of repeaters can lead to more than three received subtelegrams. The EnOcean Radio Protocol 1 standard specifies that all (identical) subtelegrams received in a window of 100 ms (beginning with the first received subtelegram) shall be considered part of the same packet.

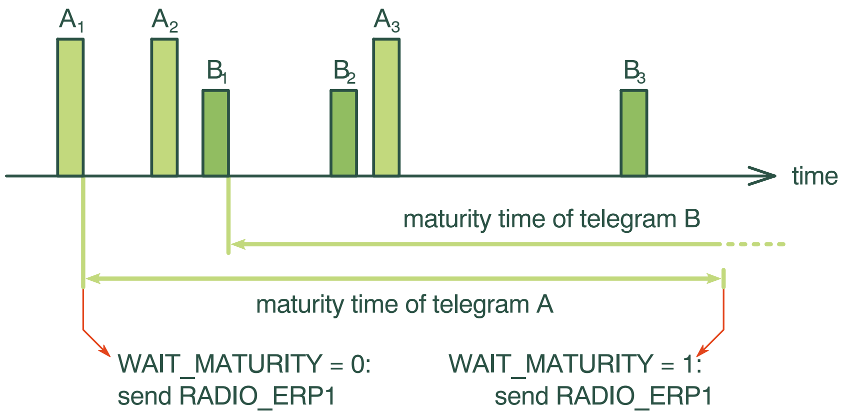


Figure 8.

Normally, packets are forwarded as soon as possible, i.e. the first received subtelegram will immediately be forwarded to the serial port as an ESP3: Packet Type 1: RADIO_ERP1 packet and the remaining subtelegrams are silently discarded. This behavior can be changed with the Code 16: CO_WR_WAIT_MATURITY command: If WAIT_MATURITY is enabled the device always waits the maturity time of 100ms to collect all subtelegrams of a transmission before the packet is forwarded to the host. This allows for more statistical measurements at the cost of an increased latency.

Filter

The AX–ENOCEAN gateway module forwards all received radio telegrams by default. Using filters, the host can choose which packets shall be passed through and which shall be discarded based on the packet's R–ORG, destination or source ID and RSSI. Filters are added (Code 11: CO_WR_FILTER_ADD) and removed (Code 12: CO_WR_FILTER_DEL, Code 13: CO_WR_FILTER_DEL_ALL) trough serial commands and filtering is disabled by default (use Code 14: CO_WR_FILTER_ENABLE).

Each filter can be either **band–pass** or **band–reject**: Passing filters will only allow packets trough that match the filter, while the behavior for rejecting filters is inverted and all packets are allowed through which do **not** match the filter.

Secondly, filters are either **application** or **repeater** filters: Application filters determine which radio telegrams are forwarded to the host as an ESP3: Packet Type 1: RADIO_ERP1 packet; while repeater filters determine which radio telegrams are repeated. Repeater filters are enabled independently of application filters by setting the repeater level to 2 (with Code 09: CO_WR_REPEATER) and will work even if application filters are disabled.

Multiple filters are combined either in a conjunctive fashion ('AND'), where a packet has to pass all filters, or in a disjunctive way ('OR'), where a single passing filter is enough for the packet to go through. This chaining mode can be set independently for application and repeater filters.

For example, assume the following filters are configured:

Filter	Pass	Reject	App.	Rep.
Source ID = 0xDEAD'BEEF	X		X	
R–ORG: 4BS		X	X	
R–ORG: RPS	X		X	
RSSI > –70 dBm	X			X
R–ORG: 4BS	X			X

Then, they will be applied as follows, depending on the mode:

Source ID	R–RORG	RSSI	Serial Forward?		Repeated?	
			AND	OR	AND	OR
0xDEAD'BEEF	4BS	–30 dBm	N	Y	Y	Y
0xDEAD'BEEF	RPS	–30 dBm	Y	Y	N	Y
0xDEAD'BEEF	VLD	–30 dBm	N	Y	N	Y
0x1234'5678	4BS	–30 dBm	N	N	Y	Y
0x1234'5678	RPS	–30 dBm	N	Y	N	Y
0x1234'5678	VLD	–30 dBm	N	Y	N	Y
0xDEAD'BEEF	4BS	–80 dBm	N	Y	N	Y
0x1234'5678	VLD	–80 dBm	N	N	N	N

Repeater

The AX−ENOCEAN gateway module has the ability to repeat radio telegrams to increase the range of existing EnOcean devices. To prevent overeager repetition, each radio telegram contains a counter, indicating how often this packet has already been received. The device will only repeat a telegram that has never been repeated before (if the repeater is set to level 1) or only if it has been repeated at most once (if set to level 2).

Important: When using repeaters, take care to ensure that regulatory transmitter duty cycle limits (if applicable) are not exceeded.

Important: Level 2 repeating functionality should only be activated after careful study of the radio conditions. Otherwise, the system performance can suffer from more collisions.

The receiver functionality can be enabled, disabled and configured through the serial interface with the Code 09: CO_WR_REPEATER command. It is also possible to only selectively repeat telegrams based on source or destination address, R−ORG and signal strength. See chapter ‘Filter’ for details of filters applied during repetition.

Remote Management

Remote Management allows EnOcean devices to be configured and maintained over the EnOcean protocol, either over the air or over the serial interface. Remote Management messages are special telegrams, containing a manufacturer ID, a function number and up to 512 bytes of payload data.

The AX−ENOCEAN modules transparently forward remote management messages through the EnOcean Radio Protocol 1 as SYS_EX telegrams (R−ORG 0xC5). Use the ESP3: Packet Type 7: REMOTE MANAGEMENT command to send a remote management message to another EnOcean device. The AX−ENOCEAN module will take care of the proper encoding and splitting of the payload data into multiple radio packets. The ESP3: Packet Type 1: RADIO_ERP1 can also be used to transmit a remote management command, but the host must handle the encoding and splitting manually. Received SYS_EX radio packets are automatically merged and (if successful) forwarded as ESP3: Packet Type 7: REMOTE MANAGEMENT serial packet to the host.

Important: Remote management messages are not handled by the AX−ENOCEAN gateway modules themselves, but must be acted upon by the host connected to the device. For example, the device will not respond to a ‘ping’ request and will just forward the message to the host.

For the full remote management specifications, consult the Remote Management Specification from EnOcean.

GPIO

There are 15 GPIO signals that can be queried and set using serial commands (ESP3: Packet Type 0x81: GPIO). All pins support digital input and output and have an optional pull−up. Pins marked as ‘analog out’ can be driven by a clock (square wave) or a DAC. Apart from GPIO13 and GPIO14, all pins are 5 V tolerant. All GPIO pins and UARTRX start up as input with pull−up.

Number	Module Pin	Module with Adapter Pin	Additional Functionality
GPIO0	10	14	
GPIO1	11	21	SPI MISO
GPIO2	12	22	SPI MOSI
GPIO3	13	20	SPI clock
GPIO4	14	19	SPI chip select, analog out
GPIO5	15	29	
GPIO6	16	34	
GPIO7	17	30	
GPIO8	18	33	
GPIO9	19		
GPIO10	20		
GPIO11	21		
GPIO12	22		
GPIO13	29	10	analog in, analog out
GPIO14	30	9	analog in

RADIO INTERFACE: ERP1

This chapter provides a quick overview over the EnOcean Radio Protocol 1. Knowledge of the technicalities of ERP1 is not required to operate an AX−ENOCEAN module though. Please consult the EnOcean Radio Protocol 1 specification ISO/IEC 14543−3−10 for all technical details.

Normal ERP1 Telegram

This is the basic EnOcean Telegram and used for most communications. The size of the data field depends on the R−ORG and varies between 1 and 14 bytes.

Offset	Size	Field	Description
0	1	R−ORG	The type of radio packet (see R−ORG)
1	N	Data	Actual payload of telegram, in big−endian order (highest byte is sent first, with highest bit first).
N+1	4	Source ID	Unique ID of sender device
N+5	1	Status	Various flags, such as repetition count or hash type
N+6	1	Hash	Either CRC8 or checksum, depending on status field. All fields from R−ORG to Status are hashed.

STATUS

Bit	Name	Description
0 (LSB)	REPETITION COUNT	Number of times this telegram has been repeated. Zero means this is the original telegram and has never been repeated, 0xF means don’t repeat at all.
1		
2		
3		
4	ORIGINAL TYPE	For converted switch telegrams (RPS), this indicates the original R−ORG of the switch packet: 0b10: R−ORG was 5, 0b11: R−ORG was 6
5		
6	−	Not used
7 (MSB)	HASH TYPE	0: checksum, 1: CRC8

Switch Telegram

Some energy harvesting devices such as battery−less switches have a very limited energy budget and short transmission times are required to conserve power. The switch telegram is the shortest possible EnOcean telegram, consisting of only 6 bytes and requiring just half a millisecond to send.

Switch telegrams are converted to normal telegrams upon reception and do not have to be handled differently. They are converted to an RPS (‘repeated switch telegram’) message and the original 4−bit R−ORG (5 or 6) is stored in the status field.

Offset	Size	Field	Description
0	0.5	R−ORG	4−bit R−ORG, either 5 or 6
0.5	1	Data	Actual payload of telegram
1.5	4	Source ID	Unique ID of sender device
5.5	0.5	Hash	4−bit checksum of R−ORG, data and ID

Addressed Telegram

To provide not only a source ID but also a destination ID, a telegram can be encapsulated in an ADT telegram (‘Addressing Destination Telegram’) as follows (yellow fields are added, while the other fields are encapsulated without modification):

Offset	Size	Field	Description
0	1	R−ORG = 0xA6	Addressed telegram encapsulating a normal one
1	1	Original R−ORG	The R−ORG of the original telegram
2	N	Original Data	The data of the original telegram
N+2	4	Destination ID	The destination address
N+6	4	Original Source ID	Original sender ID
N+10	1	Original Status	Original status byte
N+11	1	Hash	Either CRC8 or checksum, depending on status field

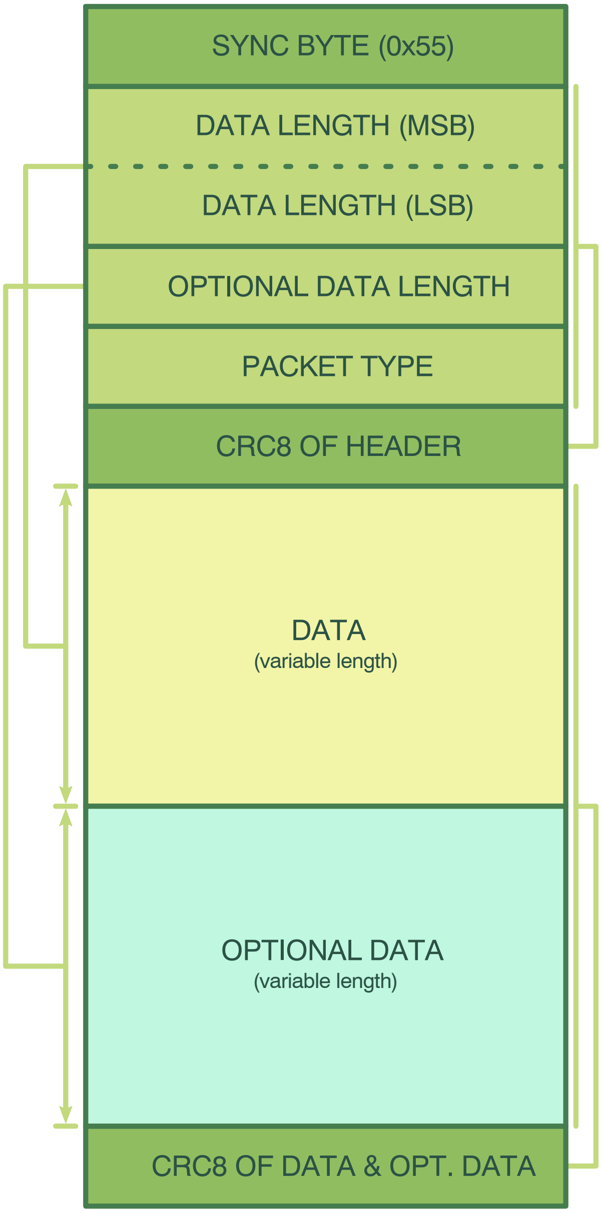
While manual encapsulation is possible to transmit an addressed telegram, the ESP3: Packet Type 1: RADIO_ERP1 has an optional destination ID field and the AX−ENOCEAN module will take care of the encapsulation. Likewise, addressed telegrams are automatically unpacked upon reception and the destination address is in the optional data part of the serial packet.

Important: AX−ENOCEAN gateway modules will receive and forward all telegrams by default, even if they are addressed to a different device. To receive only telegrams with specific addresses, set up a corresponding Filter.

COMMAND INTERFACE: ESP3

Serial Parameters: 57600, 8, N, 1

The AX−ENOCEAN module uses UART to communicate with a host (such as a microcontroller or a PC) with a bit rate of 57600 baud, no parity bit, 8 data bits and one stop bit. The ENOCEAN modules use the EnOcean Serial Protocol 3 (ESP3), a binary protocol designed for reliability and flexibility. The full specifications are publicly available from EnOcean, parts of which were reproduced here for reference.



Framing

The EnOcean Serial Protocol 3 uses a 6−byte header for packet synchronization. A synchronization byte (0x55) is used to detect the start of a packet. When such a byte is received, the subsequent 5 bytes are checked and if the CRC is correct, packet reception begins. If not, the sync. byte is discarded and the module looks for the next sync. byte.

Note that there is no escaping at all, a synchronization byte can also appear in the body of a packet, valid synchronization bytes are recognized only through the CRC8 following 5 bytes later.

The actual payload of a serial packet has two parts: DATA, containing the actual information, and OPTIONAL DATA, which contains additional data that can be left out. The format of the payload is determined by the packet type field of the header. Some packets are further classified through a subtype, called “code”, at the beginning of the DATA section (see ESP3: Packet Type 5: COMMON_COMMAND for details).

Timeout

An ESP3 packet is dropped when the time between two bytes exceeds 100 ms. Likewise, if a ESP3: Packet Type 5: COMMON_COMMAND is not followed by a ESP3: Packet Type 2: RESPONSE within 500 ms a timeout is identified as well. Note though that other types of packets might have a longer response time due to their nature (such as a remote management message, which can have a delay up to 2 seconds).

Upward Compatibility

The ESP3 protocol is defined as a specific structure of Sync.−Byte, Header & CRC8, which should not be changed in future versions. For each type of packet the content and the length of DATA is different. Today’s applications have to be compliant with later versions of the ESP3 protocol ensuring an upwards compatibility. New software applications or devices might require the definition of new types of packet.

Existing packet types may be modified only via the field OPTIONAL_DATA. The field DATA is not to be changed.

Existing devices will react as follows:

- Unknown packet types are confirmed with the ESP3: Packet Type 2: RESPONSE message ‘not supported’ and will not be processed further.
- New fields in the OPTIONAL DATA section of an existing packet type will be ignored; an ESP3: Packet Type 2: RESPONSE message will not be sent.
- It is allowed to skip bytes (not transfer them) from optional fields when they are located at the end of the optional field.

Thus, backwards compatibility is secured.

Commands

ESP3 packets are categorized into different types:

Type	Name	Description
1	ESP3: Packet Type 1: RADIO_ERP1	Radio Telegram
2	ESP3: Packet Type 2: RESPONSE	Response to any command or query
4	ESP3: Packet Type 4:	Event generated by device, might require answer from host
5	ESP3: Packet Type 5: COMMON_COMMAND	Various commands
7	ESP3: Packet Type 7: REMOTE MANAGEMENT	Remote management commands
11 – 127		Reserved for EnOcean
128 – 255		Manufacturer specific commands and messages

ESP3: Packet Type 5: COMMON_COMMANDs are further divided by code:

Code	Name	Description
1	Code 01: CO_WR_SLEEP	Enter in energy saving mode for a pre−specified duration.
2	Code 02: CO_WR_RESET	Reset the device
3	Code 03: CO_RD_VERSION	Read the software and hardware version, chip ID etc.
7	Code 07: CO_WR_IDBASE	Write ID range base number
8	Code 08: CO_RD_IDBASE	Read ID range base number
9	Code 09: CO_WR_REPEATER	Set Repeater Level off,1,2
10	Code 10: CO_RD_REPEATER	Read Repeater Level off,1,2
11	Code 11: CO_WR_FILTER_ADD	Add filter to filter list
12	Code 12: CO_WR_FILTER_DEL	Delete filter from filter list
13	Code 13: CO_WR_FILTER_DEL_ALL	Delete all filters
14	Code 14: CO_WR_FILTER_ENABLE	Enable/Disable supplied filters
15	Code 15: CO_RD_FILTER	Read supplied filters
16	Code 16: CO_WR_WAIT_MATURITY	Wait until the end of the RX maturity time before forwarding received radio telegrams.

All these commands are sent to the chip and will be answered with an ESP3: Packet Type 2: RESPONSE packet. Unknown or unsupported types are answered with a NOT_SUPPORTED ESP3: Packet Type 2: RESPONSE.

Important: All common commands which change settings (CO_WR_*) are writing to volatile memory and these settings will be lost upon reset or power−loss! The only exception to this is Code 07: CO_WR_IDBASE, which writes the base ID to flash and can only be executed a limited number of times.

ESP3: PACKET TYPE 1: RADIO_ERP1

RADIO_ERP1 packets are used to send and receive radio telegrams. When an AX−ENOCEAN device receives an EnOcean radio telegram, it is encapsulated in a RADIO_ERP1 serial packet and transmitted through UART (no acknowledge from the host is require in this case). To transmit a radio packet, encapsulate it in the data part of the serial packet. The fields in the optional data part can be omitted.

The format of a radio packet is described in 'Radio Interface: ERP1', see next chapter for a few examples.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0xnnnn	Length of radio telegram
	3	1	Optional Data Length	0x01	7 bytes of optional data
	4	1	Packet Type	0x01	RADIO_ERP1 = 1
	5	1	CRC8		
Data	6	N	Radio Telegram	...	Radio telegram without hash.
Optional Data	6+N	1	Subtelegram count	0x0n	Receive: Number of received subtelegrams. Transmit: Ignored
	7+N	4	Destination ID	0xnnnn'nnnn	Address of destination device. Default: 0xFFFF'FFFF = broadcast
	11+N	1	Best dBm	0xnn	Receive: Highest RSSI value of all received subtelegrams (absolute value, without minus)
	12+N	1	Security Level	0x00	Type of encryption (deprecated, not supported anymore)
	13+N	1	CRC8		

Possible return codes:

00	RET_OK
01	RET_ERROR
02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

Examples

For more details and radio telegram types, see 'Radio Interface: ERP1' or consult the documentation from EnOcean (especially the EnOcean Equipment Profiles). Here are a few examples of how the radio telegrams are encoded as an ESP3 packet:

VLD: VARIABLE LENGTH DATA

Group	Offset	Size	Field	Value	Description
Data	6	1	R−ORG	0xD2	Radio type VLD = 0xD2
	7	N	User Data	...	1−14 bytes of payload data
	7+N	4	Sender ID	0xnnnn'nnnn	Unique device sender ID or base ID.
	11+N	1	Status	0xnn	Telegram control bits; used for repeating, switch telegram encapsulation and hash type.

ADT: ADDRESSING DESTINATION TELEGRAM

Group	Offset	Size	Field	Value	Description
Data	6	1	R–ORG	0xnn	Radio type (R–ORG, e.g. 0xA5 for 4BS)
	7	N	User Data	...	1–9 bytes of payload data
	7+N	4	Sender ID	0xnnnn’nnnn	Unique device sender ID or base ID.
	11+N	1	Status	0xnn	Telegram control bits; used for repeating, switch telegram encapsulation and hash type.
Optional Data	12+N	1	Subtelegram count	0x0n	Receive: Number of received subtelegrams. Transmit: Ignored
	13+N	4	Destination ID	0xnnnn’nnnn	Address of destination device.
	17+N	1	Best dBm	0xnn	Receive: Highest RSSI value of all received subtelegrams (absolute value, without minus)
	18+N	1	Security Level	0x00	Type of encryption (deprecated, not supported anymore)

4BS: 4 bytes Communication (with EEP Profile 07–02–14)

This is an example of how actual sensor data is encoded in the EnOcean protocol. Encoding and decoding is performed by the host, the AX–ENOCEAN modules just transparently forward the payload data. EnOcean Equipment Profiles are used to specify the data format.

In this example, we are using EEP 07–02–14 (Temperature sensor with a range of –20°C to +60°C).

Group	Offset	Size	Field	Value	Description
Data	6	1	R–ORG	0xA5	Radio type 4BS = 0xA5
	7	1	Data Byte 3	0x00	Unused in EEP 07–02–14
	8	1	Data Byte 2	0x00	
	9	1	Data Byte 1	0xnn	Temperature value 0–255 0 = –20°C, 255 = +60°C
	10	1	Data Byte 0	0b0000’n000	DB_0.BIT3 = Learn Bit 1: Normal, data telegram 0: Teach–in telegram
	11	4	Sender ID	0xnnnn’nnnn	Unique device sender ID or base ID.
	15	1	Status	0xnn	Telegram control bits; used for repeating, switch telegram encapsulation and hash type.

ESP3: PACKET TYPE 2: RESPONSE

Packets sent by the host are always acknowledged through a RESPONSE message. RESPONSE messages are identified by their type 0x02 and the first DATA byte, which specifies the return code:

Sync. Byte	
Header	
CRC8	
DATA	Return Code
	Optional Response Data
CRC8	

LIST OF RETURN CODES

Code	Name	Description
00	RET_OK	Everything is fine, command was successfully executed.
01	RET_ERROR	Something went wrong; the command could not be successfully executed.
02	RET_NOT_SUPPORTED	This functionality is not supported by the current firmware, the command was ignored.
03	RET_WRONG_PARAM	There was a wrong (e.g. out of bounds) or missing parameter.
04	RET_OPERATION_DENIED	The command would have violated some sort of protection (such as accessing code−protected memory).
> 128		Return codes greater than 0x80 are used for commands with special return information and are specific to these commands.

EXAMPLE: RET_OK (WITHOUT RESPONSE DATA)

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0001	1 byte of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x02	RESPONSE = 2
	5	1	CRC8		
Data	6	1	Return Code	0x00	RET_OK = 0
	7	1	CRC8		

Each command will be answered with RET_OK upon success. Some commands only respond with RET_OK without any additional data. Others respond to a query by delivering additional data. The format of such responses is documented by the respective commands.

ESP3: PACKET TYPE 4: EVENT

Some operations or processes generate asynchronous events to notify the host of something or to ask for a response. Events are always sent by the AX−ENOCEAN chip and might require an ESP3: Packet Type 2: RESPONSE from the host.

Currently, only the sleep command generates an event when the chip wakes up.

Sync. Byte	
Header	
CRC8	
DATA	Event Code
	Optional Event Data
CRC8	

LIST OF RETURN CODES

Code	Name	Description
01	SA_RECLAIM_NOT_SUCCESSFUL	Informs the backbone of a Smart Ack Client about an unsuccessful reclaim.
02	SA_CONFIRM_LEARN	Used for SMACK to confirm/discard learn in/out.
03	SA_LEARN_ACK	Inform backbone about result of a learn request.
04	Code 04: CO_READY	Inform host about the readiness for operation.
05	CO_EVENT_SECUREDEVICES	Informs about a secure device.
06	CO_DUTYCYCLE_LIMIT	Informs about duty cycle limit.

CODE 04: CO_READY

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0002	2 bytes of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x04	EVENT = 4
	5	1	CRC8		
Data	6	1	EVENT Code	0x04	CO_READY = 4
	7	1	Reset Cause	0xnn	02: Wake up from sleep
	8	1	CRC8		

This event requires no response from the host.

ESP3: PACKET TYPE 5: COMMON_COMMAND

Code 01: CO_WR_SLEEP

Puts the chip into sleep mode to minimize energy consumption by powering down all components (including the radio – it will not receive radio or serial packets when sleeping). The chip will automatically wake up after the specified period. After wake up, it generates a Code 04: CO_READY. The chip will retain its memory contents during sleep. To clear the memory and reset the device to its default state, use the Code 02: CO_WR_RESET.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0005	5 bytes of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x01	CO_WR_SLEEP = 1
	7	4	Sleep time	0x00nnnnnn	Time to sleep in 10ms units. 0x00000000 = use max. value = max. data range (0x00FFFFFF, ~46 h)
	11	1	CRC8		

Possible return codes:

00	RET_OK
02	RET_NOT_SUPPORTED

Code 02: CO_WR_RESET

Resets the chip. This clears all settings that have not been written to flash.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0001	1 byte of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x02	CO_WR_RESET = 2
	7	1	CRC8		

Possible return codes:

00	RET_OK
01	RET_ERROR
02	RET_NOT_SUPPORTED

Code 03: CO_RD_VERSION

Reads information from the device, such as firmware and hardware version, chip−ID etc.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0001	1 byte of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x03	CO_RD_VERSION = 3
	7	1	CRC8		

Possible return codes:

00	RET_OK

Group	Offset	Size	Field	Value	Description	
	0	1	Sync. Byte	0x55		
Header	1	2	Data Length	0x0021	33 bytes of data	
	3	1	Optional Data Length	0x00	No optional data	
	4	1	Packet Type	0x02	RESPONSE = 2	
	5	1	CRC8			
Data	6	1	RETURN Code	0x00	RET_OK = 0	
	7	4	Application Version	0xnnnnnnnn	Byte 1	Major Version
					Byte 2	Minor Version
					Byte 3	Revision
					Byte 4	Build
	11	4	API Version	0xnnnnnnnn	Not currently used by ON Semiconductor modules.	
	15	4	Chip ID	0xnnnnnnnn	Unique Enocean ID of this device.	
	19	4	Chip Version	0xnnnnnnnn	Byte 1	Silicon Revision
					Byte 2	
					Byte 3	Unused
					Byte 4	Unused
	23	16	Application Description	ASCII	8bit ASCII, 16 character string (Null−terminated)	
	39	1	CRC8			

02	RET_NOT_SUPPORTED

Code 07: CO_WR_IDBASE

Write the Base ID range. See Addressing for details of this feature.

Important: This function can only be called 10 times to change the base ID. There is no possibility to reset this constraint. Also power off/on will not allow more than 10 changes!

Note that CO_WR_IDBASE writes the Base ID to persistent flash memory; the Base ID will survive reset and power loss. This also means that once the allocated space is full, the Base ID cannot be changed anymore!

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0005	5 bytes of data
	3	1	Optional Length	0x00	No optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x02	CO_WR_RESET = 2
	7	4	Base ID	0xFFnn’nnn0	Range between 0xFF80’0000 and 0xFFFF’FF80
	11	1	CRC8		

Possible return codes:

0x00	RET_OK	
0x02	RET_NOT_SUPPORTED	
0x82	FLASH_HW_ERROR	The write/erase/verify process failed, the flash page seems to be corrupted.
0x90	BASEID_OUT_OF_RANGE	Base ID is not in range between 0xFF80’0000 and 0xFFFF’FF80.
0x91	BASEID_MAX_REACHED	Base ID was changed 10 times, no more changes allowed.

Code 08: CO_RD_IDBASE

Reads the Base ID range. See Addressing for details of this feature.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0001	1 byte of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x08	CO_RD_IDBASE = 8
	7	1	CRC8		

Possible return codes:

00	RET_OK

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0005	5 byte of data
	3	1	Optional Data Length	0x01	1 byte of optional data
	4	1	Packet Type	0x02	RESPONSE = 2
	5	1	CRC8		
Data	6	1	RETURN Code	0x00	RET_OK = 0
	7	4	Base ID	0xFFnn’nnn0	Currently set Base ID in range between 0xFF80’0000 and 0xFFFF’FF80
Optional Data	11	1	Remaining Writes	0x0n	Remaining write cycles for Base ID (max. 10).
	12	1	CRC8		

02	RET_NOT_SUPPORTED

Code 09: CO_WR_REPEATER

Sets the repeater level and mode. See Repeater for details of this feature.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0003	3 bytes of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x09	CO_WR_REPEATER = 9
	7	1	REP_MODE	0x00 − 0x02	0x00: OFF 0x01: ON, repeat all 0x02: ON, repeat only filtered
	8	1	REP_LEVEL	0x00 − 0x02	0x00: must be 0 when MODE is OFF 0x01: Level 1 0x02: Level 2
	9	1	CRC8		

Possible return codes:

00	RET_OK
02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

Code 10: CO_RD_REPEATER

Get the currently set repeater level and mode. See Repeater for details of this feature.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0001	1 byte of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x0A	CO_RD_REPEATER = 10
	7	1	CRC8		

Possible return codes:

00	RET_OK

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0003	3 bytes of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x02	RESPONSE = 2
	5	1	CRC8		

Group	Offset	Size	Field	Value	Description
Data	6	1	RETURN Code	0x00	RET_OK = 0
	7	1	REP_MODE	0x00 – 0x02	0x00: OFF 0x01: ON, repeat all 0x02: ON, repeat only filtered
	8	1	REP_LEVEL	0x00 – 0x02	0x00: must be 0 when MODE is OFF 0x01: Level 1 0x02: Level 2
	9	1	CRC8		

02	RET_NOT_SUPPORTED

Code 11: CO_WR_FILTER_ADD

Adds a filter. AX−ENOCEAN modules can discard received telegrams depending on their ID, R−ORG or dBm value. Note that filters have to be enabled generally to work (use Code 14: CO_WR_FILTER_ENABLE for application filters, and Code 09: CO_WR_REPEATER for repeater filters). See chapter ‘Filter’ for details of this feature.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0007	7 bytes of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x0B	CO_WR_FILTER_ADD = 11
	7	1	Filter Type	0x00 – 0x03	Which value to filter on 0x00: Source ID 0x01: R−ORG 0x02: dBm (RSSI of received telegram) 0x03: Destination ID
	8	4	Filter Value	0xnnnn’nnnn	Value to filter with: • Source or destination ID • R−ORG • dBm value (unsigned, interpreted as negative value)
	12	1	Filter Kind	0x00 0x80 0x40 0xC0	0x00: application filter (band−rejection) 0x80: application filter (band−pass) 0x40: repeater filter (band−rejection) 0xC0: repeater filter (band−pass)
	13	1	CRC8		

Possible return codes:

00	RET_OK
01	RET_ERROR (memory full)
02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

FILTER EXAMPLES

Filter Type		Filter Value		Filter Kind		Description
0x00	Source ID	0x1234’5678	Source ID	0x00	App. reject	Block specified ID
0x00	Source ID	0x1234’5678	Source ID	0x80	App. pass	Block all other IDs
0x01	R−ORG	0x0000’00A5	4BS	0x00	App. reject	Block telegrams with R−ORG 0xA5

FILTER EXAMPLES

Filter Type		Filter Value		Filter Kind		Description
0x01	R−ORG	0x0000’00A5	4BS	0x80	App. pass	Block telegrams with R−ORG other than 0xA5
0x02	dBm	0x0000’0046	−70 dBm	0x00	App. reject	Block signals weaker than −70 dBm
0x02	dBm	0x0000’0046	−70 dBm	0x80	App. pass	Block signals stronger than −70 dBm
0x00	Source ID	0x1234’5678	Source ID	0xC0	Rep. pass	Repeat only telegrams from specific source.
0x01	R−ORG	0x0000’00A5	4BS	0x40	Rep. reject	Do not repeat 4BS telegrams.
0x02	dBm	0x0000’0046	−70 dBm	0xC0	Rep. pass	Do not repeat signals stronger than −70 dBm.

Code 12: CO_WR_FILTER_DEL

Remove a filter.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0006	6 bytes of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x0C	CO_WR_FILTER_DEL = 12
	7	1	Filter Type	0x00 – 0x03	Which value to filter on 0x00: Source ID 0x01: R−ORG 0x02: dBm (RSSI of received telegram) 0x03: Destination ID
	8	4	Filter Value	0xnnnn’nnnn	Value to filter with: • Source or destination ID • R−ORG • dBm value (unsigned, interpreted as negative value)
	12	1	CRC8		

Possible return codes:

00	RET_OK
01	RET_ERROR (filter could not be found)
02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

Code 13: CO_WR_FILTER_DEL_ALL

Remove all filters.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0001	1 byte of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x0D	CO_WR_FILTER_DEL_ALL = 13
	7	1	CRC8		

Possible return codes:

00	RET_OK
02	RET_NOT_SUPPORTED

Code 14: CO_WR_FILTER_ENABLE

Enable or disable all application filters (repeater filters are enabled/disabled with Code 09: CO_WR_REPEATER). Also, configure how multiple filters act together (consult Chapter 'Filter' for an explanation of AND/OR composition).

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0003	3 bytes of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x0E	CO_WR_FILTER_DEL = 14
	7	1	Filter Enable	0x00 0x01	0x00: All application filters disabled (OFF) 0x01: All application filters enabled (ON)
	8	1	Filter Operator	0x00 0x01 0x08 0x09	0x00: OR composition of all filters 0x01: AND composition of all filters 0x08: OR for app. Filters, AND for repeater filters 0x09: AND for app. Filters, OR for repeater filters
	9	1	CRC8		

Possible return codes:

00	RET_OK
02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

Code 15: CO_RD_FILTER

List all filters.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0001	1 byte of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x0F	CO_WR_FILTER_DEL = 15
	7	1	CRC8		

Possible return codes:

00	RET_OK

Group	Offset	Size	Field	Value	Description	
	0	1	Sync. Byte	0x55		
Header	1	2	Data Length	0x0006	1 + 5*N bytes of data	
	3	1	Optional Data Length	0x00	No optional data	
	4	1	Packet Type	0x02	RESPONSE = 2	
	5	1	CRC8			
Data	6	1	RETURN Code	0x00	RET_OK = 0	
	7 + 5*N	1	Filter Type	0x00 – 0x03	Which value to filter on 0x00: Source ID 0x01: R−ORG 0x02: dBm (RSSI of received telegram) 0x03: Destination ID	Repeated N times (one entry per filter)
	8 + 5*N	4	Filter Value	0xnnnn’nnnn	Value to filter with: • Source or destination ID • R−ORG • dBm value (unsigned, interpreted as negative value)	
	12 + 5*N	1	CRC8			

02	RET_NOT_SUPPORTED

Code 16: CO_WR_WAIT_MATURITY

Wait until the end of the receive maturity time (100 ms) before received radio telegrams are forwarded over the serial link. This is disabled by default. If disabled, telegrams are forwarded faster, but the subtelegram count will always be one and only the RSSI of the first subtelegram is measured. This does not influence the repeater’s operation in any way, telegrams are always repeated as soon as possible, following the ERP1 timing standards.

See also chapter ‘Wait Maturity’ for details.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0001	2 bytes of data
	3	1	Optional Data Length	0x00	No optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x10	CO_WR_FILTER_DEL = 16
	7	1	Wait End Maturity	0x00 0x01	0x00: Telegrams are forwarded immediately 0x01: Telegrams are sent after the maturity time elapsed
	8	1	CRC8		

Possible return codes:

00	RET_OK
02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

Unsupported Commands

The following commands are explicitly not supported by the AX−ENOCEAN modules at the moment:

04	CO_RD_SYS_LOG
05	CO_WR_SYS_LOG
06	CO_WR_BIST
17	CO_WR_SUBTEL
18	CO_WR_MEM
19	CO_RD_MEM
20	CO_RD_MEM_ADDRESS
21	CO_RD_SECURITY *(deprecated)*
22	CO_WR_SECURITY *(deprecated)*
23	CO_WR_LEARNMODE
24	CO_RD_LEARNMODE
25	CO_WR_SECUREDEVICE_ADD
26	CO_WR_SECUREDEVICE_DEL
27	CO_RD_SECUREDEVICE_BY_INDEX
28	CO_WR_MODE
29	CO_RD_NUMSECUREDEVICES
30	CO_RD_SECUREDEVICE_BY_ID
31	CO_WR_SECUREDEVICE_ADD_PSK
32	CO_WR_SECUREDEVICE_SENDTEACHIN
33	CO_WR_TEMPORARY_RLC_WINDOW
34	CO_RD_SECUREDEVICE_PSK
35	CO_RD_DUTYCYCLE_LIMIT

These and all undefined common commands (such as possible future commands) are answered with a standard ESP3: Packet Type 2: RESPONSE with code RET_NOT_SUPPORTED.

Code 128: CO_WR_RX_MODE

This command activates or deactivates the receiver. The receiver is enabled by default and will forward all radio packets over the serial link.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0002	2 bytes of data
	3	1	Optional Data Length	0x00	0 bytes optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x80	CO_WR_RX_MODE = 128
	7	1	Wait End Maturity	0x00 0x01	0x00: Disable receiver 0x01: Enable receiver
	9	1	CRC8		

Possible return codes:

00	RET_OK
03	RET_WRONG_PARAM

Code 129: CO_WR_TX_TEST

This command activates or deactivates various transmission tests for regulation certification. To switch between different test modes, disable TX test mode first.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0002	2 bytes of data
	3	1	Optional Data Length	0x01	1 byte optional data
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
	5	1	CRC8		
Data	6	1	COMMAND Code	0x81	CO_WR_TX_TEST = 129
	7	1	TX Test mode	0x00 0x01 0x02 0x03	0x00: Disable TX test mode 0x01: Send CW 0x02: Send Constant Byte (default: 0xAA) 0x03: Send Random Data
Optional Data	8	1	Constant Byte	0xnn	Constant Byte to send in Mode 2 (default: 0xAA)
	9	1	CRC8		

Possible return codes:

00	RET_OK
01	RET_ERROR (already transmitting)
03	RET_WRONG_PARAM

ESP3: PACKET TYPE 7: REMOTE MANAGEMENT

Send or receive a remote management message. Received remote management telegrams (R−ORG SYS_EX = 0xC7) are merged and forwarded as type 7 serial packets. While transmitting remote management messages with a ESP3: Packet Type 1: RADIO_ERP1 packet is possible as well, the REMOTE_MAN_COMMAND additionally takes care of the correct assembly of the data fields and automatically splits longer commands into several radio telegrams.

See Remote Management for a thorough explanation on remote management.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0xnnnn	4 + N bytes of data
	3	1	Optional Data Length	0x01	10 bytes of optional data
	4	1	Packet Type	0x07	REMOTE_MAN_COMMAND = 7
	5	1	CRC8		
Data	6	2	Function Number	0x0nnn	Range: 0x0000 – 0x0FFF
	8	2	Manufacturer ID	0x0nnn	Range: 0x0000 – 0x07FF
	10	N	Message Data	...	0 to 255 bytes
Optional Data	10+N	4	Destination ID	0xnnnn’nnnn	Address of destination device. Default: 0xFFFF’FFFF = broadcast
	14+N	4	Source ID	0xnnnn’nnnn	Address of source device (unique ID or base ID).
	18+N	1	dBm	0xnn	Receive: Best RSSI of all received subtelegrams (value is dBm, without minus) Transmit: Field is ignored.
	19+N	1	Send With Delay	0x00 0x01	0x00: Default, send radio telegram immediately 0x01: Send the first message with a random delay. This must be set when answering to broadcast message to prevent collisions.
	20+N	1	CRC8		

Possible return codes:

00	RET_OK
02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

ESP3: PACKET TYPE 0X81: GPIO

Use these commands to output or input data and control the general−purpose pins of an AX−ENOCEAN device. See chapter 'GPIO' for more.

Code 00: AX_GPIO_RD_HW_INFO

Returns list of pins available and their capabilities.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0001	1 byte of data
	3	1	Optional Data Length	0x00	0 bytes optional data
	4	1	Packet Type	0x81	AX_GPIO = 129
	5	1	CRC8		
Data	6	1	GPIO Command	0x00	AX_GPIO_RD_HW_INFO = 0
	7	1	CRC8		

Possible return codes:

00	RET_OK

Group	Offset	Size	Field	Value	Description	
	0	1	Sync. Byte	0x55		
Header	1	2	Data Length	0x0002+N	2+N bytes of data	
	3	1	Optional Data Length	0x00	0 bytes of optional data	
	4	1	Packet Type	0x02	RESPONSE = 2	
	5	1	CRC8			
Data	6	1	RETURN Code	0x00	RET_OK = 0	
	7	1	Pin Count	0xnn	Number of pins available on this device.	
	8	1*N	Possible Modes	0b0000'00xx	Bit field indicating which analog modes are possible for each pin (digital is always possible):	
					Bit 0	Analog Input (ADC)
					Bit 1	Analog Output (DAC, clock)
	8+N	1	CRC8			

02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

Code 01: AX_GPIO_WR_PIN_MODE

Sets GPIO pin to specific mode (e.g. output high or an analog input).

Group	Offset	Size	Field	Value	Description	
	0	1	Sync. Byte	0x55		
Header	1	2	Data Length	0x0003	3 bytes of data	
	3	1	Optional Data Length	0x00	0 bytes of optional data	
	4	1	Packet Type	0x81	AX_GPIO = 129	
	5	1	CRC8			
Data	6	1	GPIO Command	0x01	AX_GPIO_WR_PIN_MODE = 1	
	7	1	Pin Number	0x0n	GPIO Pin 0−14 (see Pinout)	
	8	1	Pin Mode	0x0n	Bit field specifying GPIO pin mode:	
					0x00	Digital Input, floating
					0x04	Digital Input, with pull−up
					0x02	Digital Output, LOW
					0x06	Digital Output HIGH
					0x01	Analog Input
					0x03	Analog Output (driven by clock or DAC)
	9	1	CRC8			

Possible return codes:

00	RET_OK
02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

Code 02: AX_GPIO_RD_PIN_MODE

Reads current pin mode and returns possible GPIO modes.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0002	2 bytes of data
	3	1	Optional Data Length	0x00	0 bytes optional data
	4	1	Packet Type	0x81	AX_GPIO = 129
	5	1	CRC8		
Data	6	1	GPIO Command	0x02	AX_GPIO_RD_PIN_MODE = 2
	7	1	Pin Number	0x0n	GPIO Pin 0−14 (see Pinout)
	8	1	CRC8		

Possible return codes:

00	RET_OK

Group	Offset	Size	Field	Value	Description	
	0	1	Sync. Byte	0x55		
Header	1	2	Data Length	0x0002	2 bytes of data	
	3	1	Optional Data Length	0x01	1 byte of optional data	
	4	1	Packet Type	0x02	RESPONSE = 2	
	5	1	CRC8			
Data	6	1	RETURN Code	0x00	RET_OK = 0	
	7	1	Current Mode	0x0n	Currently set mode of pin, set by Code 01: AX_GPIO_WR_PIN_MODE.	
					0x00	Digital Input, floating
					0x04	Digital Input, with pull−up
					0x02	Digital Output, LOW
					0x06	Digital Output HIGH
					0x01	Analog Input
					0x03	Analog Output (driven by clock or DAC)
Optional Data	8	1	Possible Modes	0b0000’00xx	Bit field indicating which analog modes are possible on this pin (digital is always possible):	
					Bit 0	Analog Input (ADC)
					Bit 1	Analog Output (DAC, clock)
	9	1	CRC8			

02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

Code 03: AX_GPIO_RD_PIN_VALUE

Measure the voltage applied to a GPIO pin. If the pin is configured as digital, this command will return a binary value; otherwise, the voltage will be sampled and returned as a fraction of the measurement range (1 V if none is specified). The command also allows measurement of the voltage difference across two GPIO pins. In differential mode, the full scale range may also be specified as 0.1 V, 1 V or 10 V. Note however that the pin input voltages must not exceed the range 0..VDD.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0003	2 bytes of data
	3	1	Optional Data Length	0x00	2 bytes of optional data
	4	1	Packet Type	0x81	AX_GPIO = 129
	5	1	CRC8		
Data	6	1	GPIO Command	0x03	AX_GPIO_RD_PIN_VALUE = 3
	7	1	Pin Number	0x0n	GPIO Pin 0−14 (see Pinout)

Group	Offset	Size	Field	Value	Description
Optional Data	8	1	ADC Mode	0x01 0x02 0x03 0x04	When reading an analog pin, it is also possible to measure the voltage difference across two pins. Both pins must be configured as analog input. 0x01 | Single Ended (default) | 1 V 0x02 | Differential | 0.1 V 0x03 | Differential | 1 V 0x04 | Differential | 10 V
	9	1	Differential Pin	0x0n	If a differential mode is specified, this is the secondary pin (must also be an analog input). Measured voltage is V(pin) – V(differential pin).
	10	1	CRC8		

Possible return codes:

00	RET_OK

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0002	3 bytes of data
	3	1	Optional Data Length	0x00	0 bytes optional data
	4	1	Packet Type	0x02	RESPONSE = 2
	5	1	CRC8		
Data	6	1	RETURN Code	0x00	RET_OK = 0
	7	2	Pin Value	0xnnnn	Measured value at pin 0x0000/0x0001 for digital pins fraction of full scale range (1 V by default) for analog input pins unsigned 16bit integer for single ended signed 16bit integer for differential measurements
	9	1	CRC8		

02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

Code 04: AX_GPIO_WR_ANALOG_OUT

This command outputs a square wave or a ΣΔ DAC value on all pins configured as analog output with Code 01: AX_GPIO_WR_PIN_MODE. If not specified, the reference frequency for square wave output defaults to 20 MHz, otherwise the value is rounded to the next possible value.

Possible Reference Frequencies			
40 000 000	02 62 5A 00	1 280	00 00 05 00
20 000 000	01 31 2D 00	640	00 00 02 80
10 000 000	00 98 96 80	320	00 00 01 40
5 000 000	00 4C 4B 40	160	00 00 00 A0
2 500 000	00 26 25 A0	80	00 00 00 50
1 250 000	00 13 12 D0	40	00 00 00 28
625 000	00 09 89 68	20	00 00 00 14
312 500	00 04 C4 B4	10	00 00 00 0A

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0003	4 bytes of data
	3	1	Optional Data Length	0x00	4 bytes of optional data
	4	1	Packet Type	0x81	AX_GPIO = 129
	5	1	CRC8		
Data	6	1	GPIO Command	0x04	AX_GPIO_WR_ANALOG_OUT = 4
	7	1	Mode	0x00 0x01 0x02	Output a signal on pins configured as analog output. Mode 0x00: OFF Disable clock generator and DAC. Mode 0x01: Square Wave Output a square wave on analog output pins with frequency $(freq / 2^{16}) \cdot reffreq$ Mode 0x02: ΣΔ DAC Output a ΣΔ DAC value on the pin(s) configured as analog output. Value may be in the range −32768 ... 32767. The average output voltage is $(1/2 + value/2^{17}) \cdot VDD$ A low pass filter is required to get smooth voltages.
	8	2	Frequency / Value	0xnnnn	The (unsigned 16−bit) frequency of the square wave, respective the (signed 16−bit) *value* of the DAC output.
Optional Data	10	4	Reference Frequency	0xnnnnnnnn	Reference frequency for square wave output, rounded to nearest possible value.
	14	1	CRC8		

Possible return codes:

00	RET_OK
02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

Code 05: AX_GPIO_RD_ANALOG_OUT

Return the settings of the clock generator respective the DAC.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0002	1 byte of data
	3	1	Optional Data Length	0x00	0 bytes of optional data
	4	1	Packet Type	0x81	AX_GPIO = 129
	5	1	CRC8		
Data	6	1	GPIO Command	0x05	AX_GPIO_RD_ANALOG_OUT = 5
	8	1	CRC8		

Possible return codes:

00	RET_OK

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0002	2 bytes of data
	3	1	Optional Data Length	0x01	6 bytes of optional data
	4	1	Packet Type	0x02	RESPONSE = 2
	5	1	CRC8		
Data	6	1	RETURN Code	0x00	RET_OK = 0
	7	1	Mode	0x00 0x01 0x02	Current setting of the analog output(s). 0x00 OFF 0x01 Square Wave 0x02 DAC
Optional Data	8	2	Frequency / DAC Value	0xnnnn	Frequency of the square wave or value of the DAC.
	10	4	Reference Frequency	0xnnnn nnnn	Reference frequency for the square wave generator.
	14	1	CRC8		

02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

Code 06/07: AX_GPIO_RD_TEMPL/VDD

These commands read the internal temperature sensor, respective the VDDIO voltage of an AX−ENOCEAN module.

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0001	1 byte of data
	3	1	Optional Data Length	0x00	0 bytes of optional data
	4	1	Packet Type	0x81	AX_GPIO = 129
	5	1	CRC8		
Data	6	1	GPIO Command	0x06 0x07	AX_GPIO_RD_TEMP = 6 AX_GPIO_RD_VCC = 7
	7	1	CRC8		

Possible return codes:

00	RET_OK

<table>
<tr><th>Group</th><th>Offset</th><th>Size</th><th>Field</th><th>Value</th><th colspan="2">Description</th></tr>
<tr><td></td><td>0</td><td>1</td><td>Sync. Byte</td><td>0x55</td><td colspan="2"></td></tr>
<tr><td rowspan="3">Header</td><td>1</td><td>2</td><td>Data Length</td><td>0x0003</td><td colspan="2">3 bytes of data</td></tr>
<tr><td>3</td><td>1</td><td>Optional Data Length</td><td>0x00</td><td colspan="2">0 bytes of optional data</td></tr>
<tr><td>4</td><td>1</td><td>Packet Type</td><td>0x02</td><td colspan="2">RESPONSE = 2</td></tr>
<tr><td></td><td>5</td><td>1</td><td>CRC8</td><td></td><td colspan="2"></td></tr>
<tr><td rowspan="4">Data</td><td>6</td><td>1</td><td>RETURN Code</td><td>0x00</td><td colspan="2">RET_OK = 0</td></tr>
<tr><td rowspan="3">7</td><td rowspan="3">2</td><td rowspan="3">Temperature / VDDIO</td><td rowspan="3">0xnnnn</td><td><b>Measurement</b></td><td><b>Unit</b></td></tr>
<tr><td>Internal Temperature</td><td>1/10th °C</td></tr>
<tr><td>VDDIO Voltage</td><td>mV</td></tr>
<tr><td></td><td>9</td><td>1</td><td>CRC8</td><td></td><td colspan="2"></td></tr>
</table>

02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

Code 08: AX_GPIO_WR_SPI

This command clocks out bytes on the SPI port. The clock frequency is 312.5 kHz. The command returns the bytes read on MISO during output. Optionally the clocking mode may be specified. SPI pins should be configured as digital inputs/outputs before using SPI.

Number	IC Pin	Additional Functionality
GPIO1	11	SPI MISO
GPIO2	12	SPI MOSI
GPIO3	13	SPI clock
GPIO4	14	SPI chip select, analog out

<table>
<tr><th>Group</th><th>Offset</th><th>Size</th><th>Field</th><th>Value</th><th colspan="3">Description</th></tr>
<tr><td></td><td>0</td><td>1</td><td>Sync. Byte</td><td>0x55</td><td colspan="3"></td></tr>
<tr><td rowspan="3">Header</td><td>1</td><td>2</td><td>Data Length</td><td>0x00nn</td><td colspan="3">1+N bytes of data</td></tr>
<tr><td>3</td><td>1</td><td>Optional Data Length</td><td>0x01</td><td colspan="3">1 byte of optional data</td></tr>
<tr><td>4</td><td>1</td><td>Packet Type</td><td>0x81</td><td colspan="3">AX_GPIO = 129</td></tr>
<tr><td></td><td>5</td><td>1</td><td>CRC8</td><td></td><td colspan="3"></td></tr>
<tr><td rowspan="2">Data</td><td>6</td><td>1</td><td>GPIO Command</td><td>0x08</td><td colspan="3">AX_GPIO_WR_SPI = 8</td></tr>
<tr><td>7</td><td>N</td><td>Transmit Data</td><td>0xnn</td><td colspan="3">N bytes of data to transmit (max. 90 bytes).</td></tr>
<tr><td rowspan="6">Optional Data</td><td rowspan="6">7+N</td><td rowspan="6">1</td><td rowspan="6">SPI Mode</td><td rowspan="6">0x00
0x01
0x02
0x03</td><td colspan="3">Specifies the clocking mode. Mode 0 is default.</td></tr>
<tr><td><b>Mode</b></td><td><b>Clock Polarity</b></td><td><b>Clock Phase</b></td></tr>
<tr><td>0x00</td><td>normal (0)</td><td>normal (0)</td></tr>
<tr><td>0x01</td><td>normal (0)</td><td>alternate (1)</td></tr>
<tr><td>0x02</td><td>inverted (1)</td><td>normal (0)</td></tr>
<tr><td>0x03</td><td>inverted (1)</td><td>alternate (1)</td></tr>
<tr><td></td><td>8+N</td><td>1</td><td>CRC8</td><td></td><td colspan="3"></td></tr>
</table>

Possible return codes:

00	RET_OK

Group	Offset	Size	Field	Value	Description
	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x00nn	1 + N bytes of data
	3	1	Optional Data Length	0x00	0 bytes of optional data
	4	1	Packet Type	0x02	RESPONSE = 2
	5	1	CRC8		
Data	6	1	RETURN Code	0x00	RET_OK = 0
	7	N	Received Data	0xnn	N bytes of data read on MISO during transmit.
	7 + N	1	CRC8		

02	RET_NOT_SUPPORTED
03	RET_WRONG_PARAM

ESP3: CRC8 CALCULATION

The polynomial $G(x) = x^8 + x^2 + x^1 + x^0$ is used to generate the CRC8 table for the CRC8 computation. The following C code illustrates how the CRC8 value is calculated:

```
  uint8_t crc8_table[256] = {
            0x00, 0x07, 0x0e, 0x09, 0x1c, 0x1b, 0x12, 0x15,
            0x38, 0x3f, 0x36, 0x31, 0x24, 0x23, 0x2a, 0x2d,
            0x70, 0x77, 0x7e, 0x79, 0x6c, 0x6b, 0x62, 0x65,
            0x48, 0x4f, 0x46, 0x41, 0x54, 0x53, 0x5a, 0x5d,
            0xe0, 0xe7, 0xee, 0xe9, 0xfc, 0xfb, 0xf2, 0xf5,
            0xd8, 0xdf, 0xd6, 0xd1, 0xc4, 0xc3, 0xca, 0xcd,
            0x90, 0x97, 0x9e, 0x99, 0x8c, 0x8b, 0x82, 0x85,
            0xa8, 0xaf, 0xa6, 0xa1, 0xb4, 0xb3, 0xba, 0xbd,
            0xc7, 0xc0, 0xc9, 0xce, 0xdb, 0xdc, 0xd5, 0xd2,
            0xff, 0xf8, 0xf1, 0xf6, 0xe3, 0xe4, 0xed, 0xea,
            0xb7, 0xb0, 0xb9, 0xbe, 0xab, 0xac, 0xa5, 0xa2,
            0x8f, 0x88, 0x81, 0x86, 0x93, 0x94, 0x9d, 0x9a,
            0x27, 0x20, 0x29, 0x2e, 0x3b, 0x3c, 0x35, 0x32,
            0x1f, 0x18, 0x11, 0x16, 0x03, 0x04, 0x0d, 0x0a,
            0x57, 0x50, 0x59, 0x5e, 0x4b, 0x4c, 0x45, 0x42,
            0x6f, 0x68, 0x61, 0x66, 0x73, 0x74, 0x7d, 0x7a,
            0x89, 0x8e, 0x87, 0x80, 0x95, 0x92, 0x9b, 0x9c,
            0xb1, 0xb6, 0xbf, 0xb8, 0xad, 0xaa, 0xa3, 0xa4,
            0xf9, 0xfe, 0xf7, 0xf0, 0xe5, 0xe2, 0xeb, 0xec,
            0xc1, 0xc6, 0xcf, 0xc8, 0xdd, 0xda, 0xd3, 0xd4,
            0x69, 0x6e, 0x67, 0x60, 0x75, 0x72, 0x7b, 0x7c,
            0x51, 0x56, 0x5f, 0x58, 0x4d, 0x4a, 0x43, 0x44,
            0x19, 0x1e, 0x17, 0x10, 0x05, 0x02, 0x0b, 0x0c,
            0x21, 0x26, 0x2f, 0x28, 0x3d, 0x3a, 0x33, 0x34,
            0x4e, 0x49, 0x40, 0x47, 0x52, 0x55, 0x5c, 0x5b,
            0x76, 0x71, 0x78, 0x7f, 0x6A, 0x6d, 0x64, 0x63,
            0x3e, 0x39, 0x30, 0x37, 0x22, 0x25, 0x2c, 0x2b,
            0x06, 0x01, 0x08, 0x0f, 0x1a, 0x1d, 0x14, 0x13,
            0xae, 0xa9, 0xa0, 0xa7, 0xb2, 0xb5, 0xbc, 0xbb,
            0x96, 0x91, 0x98, 0x9f, 0x8a, 0x8D, 0x84, 0x83,
            0xde, 0xd9, 0xd0, 0xd7, 0xc2, 0xc5, 0xcc, 0xcb,
            0xe6, 0xe1, 0xe8, 0xef, 0xfa, 0xfd, 0xf4, 0xf3
};
uint8_t process_crc8(const uint8_t crc8, const uint8_t data)
{
    return crc8_table[(crc8) ^ (data)];
}
uint8_t crc_8bit(const uint8_t * data, uint8_t length)
{
            uint8_t sum = 0;
            while (length-- > 0) {
                        sum = process_crc8(sum, *data);
                        data++;
            }
            return sum;
}
```

See also ISO/IEC:2012 14543−3−10 (Annex A.3).

Life Support Application

This product is not designed for use in life support appliances, devices, or in systems where malfunction of this product can reasonably be expected to result in personal injury. ON Semiconductor customers using or selling this product for use in such applications do so at their own risk and agree to fully indemnify ON Semiconductor for any damages resulting from such improper use or sale.

Table 3. DEVICE VERSIONS
ENOCEAN EU (868 MHz ASK)

		CO_RD_VERSION				
		APP Version			Chip Version	
Product	Part Number	[0]	[1]	[2]	[0]	[1]
MiniStamp	AX−EO10−MINI21−868−B1	0x01	0x00	0x00	0x8F	0x51
AntStamp	AX−EO10−ANT21−868−B1					
USB−Stick	AX−EO01−USB01−868−GEVB					

NOTE: An AX−ENOCEAN IC is also available. Please see AX−ENOCEAN−IC/D on our website at www.onsemi.com for additional information.

EnOcean is a registered trademark of EnOcean GmbH.

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Literature Distribution Center for ON Semiconductor
19521 E. 32nd Pkwy, Aurora, Colorado 80011 USA
Phone: 303−675−2175 or 800−344−3860 Toll Free USA/Canada
Fax: 303−675−2176 or 800−344−3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800−282−9855 Toll Free USA/Canada
Europe, Middle East and Africa Technical Support:
Phone: 421 33 790 2910
Japan Customer Focus Center
Phone: 81−3−5817−1050

ON Semiconductor Website: **www.onsemi.com**

Order Literature: http://www.onsemi.com/orderlit

For additional information, please contact your local Sales Representative