FAST EnergyCam Wireless M-Bus

Protocol description

Table of Contents

Index of Illustration

Index of Tables

General

EnergyCam (named EC later on) radio communication is according to Wireless M-Bus (wM-Bus) standard (EN 13757-4:2011) including Open Metering System Specification (OMS) extension.

Data collector

A data collector (collector) is used to communicate with one or several wM-Bus sensors, like our EC. The Collector is not able to send data directly to EC since EC is usually in power-down mode and cannot receive radio data at this time. A radio communication to collector is always initiated by the EC, in this situation the Collector is able to send data to EC back again (when a bidirectional mode is used).

Default mode is T2 which has bidirectional communication features. Alternatively mode S2 can be used, too. Even collectors running only T1 or S1 mode are supported but lacks bidirectional communication, thus automatic installation via (SND_IR = Installation Request) cannot be used. Collectors running T1 (S1) mode are able to receive data from EC running T2 (S2) mode.

wM-Bus installation procedure

A unique serial number (M-Bus ID) is written on the EC body. This number is used in A-Field named Ident Number of address and cannot be changed. The used Manufacturer ID is "FFD". This allows to identify a certain EC at the data collector once a data packet has been successfully received.

Note that a complete wM-Bus address comprises four elements: ManufacturerID, IdentNumber, Version and Type. Usually all these elements are needed to identify a certain EC by a collector. The type reflects the currently setup medium like energy (electricity), gas and water meter. Only type can be changed by the user, the others are preset during manufacturing and cannot be changed.

The EC can be configured either for manual or automatic wM-Bus installation:

Manual Installation

EC can be advices to send a single radio packet which can be received by a collector. This allows an easy installation. In this case the collector must allow to add EC address manually.

Automatic Installation (SND_IR according to OMS)

Use this method if the collector is able to reply to an installation request according to OMS (see SND_IR). An installation request is sent several times. A collector which is allowed to reply correctly is used for later communication. Note a probably used encryption key is never automatically setup in collector and has to be always added manually.

Encryption

Encryption with a static 128 bit AES key is the only encryption which is currently defined by the OMS group and therefore used in current firmware of EC. This method needs that the same key is manually configured in both, the EC and the Collector.

Once the norm is published with further security mechanism like TLS a new EC firmware will be released to support them.

Examples of not encrypted radio packet

EC will transmit the following format on every OCR reading. Note that the field **Type** depends on current configuration of EC (either electricity, gas or water) and is the only part of address which is changeable.

wM-Bus defines that multi-byte fields are transmitted least significant byte first.

Example for not encrypted data for electricity meter

What	Value
Manufacturer ID	"FFD" (Three letter code), not changeable by user
Ident Number	0x23101664, not changeable by user
Version	0x01, not changeable by user
Type	0x02 (electricity), changeable by user
Value	68966.1 kWh

Table 1: Example for not encrypted data for electricity meter (energy in Wh)

Raw data on air [hex] in transmission order.

```
19 44 C4 18 64 16 10 23 01 02 B9 2F 7A 27 00 00 80 04 05 FD 85 0A 00 02 FD 08 0B 16 C3 AF
```

Example for not encrypted data for gas meter

What	Value
Manufacturer ID	"FFD" (Three letter code), not changeable by user
Ident Number	0x23101664, not changeable by user
Version	0x01, not changeable by user
Type	0x03 (gas), changeable by user
Value	68966.1 m^3

Table 2: Example for not encrypted data for gas meter (volume in m^3)

Raw data on air [hex] in transmission order.

```
19 44 C4 18 64 16 10 23 01 03 84 4A 7A 2B 00 00 80 04 15 FD 85 0A 00 02 FD 08 0F 0B AA 9C
```

Example for not encrypted data for water meter

What	Value
Manufacturer ID	"FFD" (Three letter code), not changeable by user
Ident Number	0x23101664, not changeable by user
Version	0x01, not changeable by user
Type	0x07 (water), changeable by user
Value	68966.1 m^3

Table 3: Example for not encrypted data for water meter (volume in m^3)

Raw data on air [hex] in transmission order.

```
19 44 C4 18 64 16 10 23 01 07 71 DE 7A 2D 00 00 80 04 15 FD 85 0A 00 02 FD 08 11 0C 23 A0
```

Detailed description of example for not encrypted data for electricity (energy in Wh)

Bold values are current values in example.

Byte #	Value [hex]	Name	Layer	Description
0	19	L-Field	Link Layer (DLL)	0x19=25 bytes following (without both CRCs a 2 bytes), in sum 1+25+2*2=30 bytes
1	44	C-Field		SND_NR (Send, No Response)
2	C4	M-Field		ManufacturerID [0]: **0x18C4**="FFD" for FastForwarD (see www.dlms.com/flag)
3	18	M-Field		ManufacturerID [1]: Multibyte fields are transmitted lower byte first (except CRC)
4	64	A-Field		IdentNumber [0]: **0x23101664** (read as 8 BCD numbers)
5	16	A-Field		IdentNumber [1]:
6	10	A-Field		IdentNumber [2]
7	23	A-Field		IdentNumber [3]
8	01	A-Field		Version of Meter (constant, always 1 for EC), forms together with Manufacturer-ID, IdentNumber and Type a unique wM-Bus address
9	02	A-Field		Type of Meter (medium) 0x01 Oil **0x02** Energy (electricity) 0x03 Gas 0x07 Water 0x0F UNKNOWN (see VIF dimensionless)
10	B9	CRC0 [1]		Higher byte of CRC is transmitted first. Generated over preceding bytes
11	2F	CRC0 [0]		Polynom: $x^{16}+x^{13}+x^{12}+x^{11}+x^{10}+x^{8}+x^{6}+x^{5}+x^{2}+1$, start with 0, result inverted
12	7A	CI-Field	Appli-cation Layer (APL)	ControlInfo: **0x7A**=Telegram with short Header (e.g. AC, ST, CW)
13	27	AC		ACcess number (incremented every synchronous packet which is not repeated)
14	00	ST		Signals error StaTe of meter ST **0x00**: no errors ST 0x04: battery low
15	00	CW [0]		CW=ConfigurationWord 0x8000 (controls encryption, here not encrypted) [7:4] 0x**0** (# encrypted blocks) [3:2] **00** (standard telegram) [1:0] **00** (hop counter=0)
16	80	CW [1]		[7:0]=[BAS0MMMM] [7] **1**: bidirectional because of T2 mode [6] **0**: accessible (meter remains after tx in rx state for a short period of time) 1: not accessible [5] **0**: asynchronous packet (spontaneous tx, e.g. by button press “--1--”) 1: synchronous packet (send by periodical timer) [4] **0:** reserved [3:0] 0x**0**: no AES encryption
17	04	DIF		DataInformationField for OCR value [7] Extension bit **0**: no DIF byte following (i.e. next byte is VIF) 1: next byte is also a DIF byte [6] Storage number (not used for EC) [5:4] Function field: **0**: instantaneous value (default for EC) 3: value during error state (e.g. repeating last OCR value due to error detection by EC)

Byte #	Value [hex]	Name	Layer	Description
				x: other codes not used by EC [3:0] **4**: 32 bit integer (means 4 bytes following after VIF). Other codes here not used by EC
18	05	VIF		ValueInformationField for Exponent and OCR value (here energy in Wh) [7] Extension bit **0**: no VIF byte following (i.e. next byte is value) 1: next byte is also a VIF byte, in case of medium UKNOWN two VIF bytes follows, see Table 5: VIF bytes coding for medium UNKNOWN for details [6:3] **0**: Energy [Wh] used for electricity 2: Volume[m^3] used for gas and water x: other codes not used by EC [2:0] Exponent with bias (bias depends on unit, -3 for energy, -6 for volume) **5: Exponent = 5-3 = 2 -> 10^2 factor; 689661*100 Wh = 68966100 Wh -> 68966.1 kWh** 5: Exponent = **5**-6 = -1 -> 10^{-1} factor; 689661*0.1 m^3 -> 68966.1 m^3 (used for gas and water)
19	FD	Value [0]		Value 32 bit integer **0x000A85FD**= 689661 (has to be scaled by exponent specified in VIF)
20	85	Value [1]		
21	0A	Value [2]		
22	00	Value [3]		
23	02	DIF		DataInformationField for Unique telegram identification [7] Extension bit **0**: no DIF byte following (i.e. next byte is VIF) 1: next byte is also a DIF byte [6] **0**: Storage number (not used for EC) [5:4] **0**: Function field: (not used by EC) [3:0] **2**: 16 bit integer (means 2 bytes following after VIF). Other codes here not used by EC
24	FD	VIFE		ValueInformationField for Unique telegram identification [7] Extension bit 0: no VIF byte following **1**: next byte is also a VIF byte [6:0] 0x7D: VIF Extension (next byte is VIF)
25	08	VIF		ValueInformationField for Unique telegram identification [7] Extension bit **0**: no more VIF following (i.e. next byte is value) **1**: next byte is also a VIF [6:0] **8**: Unique Telegram identifiCation
26	0B	UTI[0]		Unique telegram identification (Byte 0, EC tx counter)
27	16	UTI[1]		Unique telegram identification (Byte 1, EC pic counter)
28	C3	CRC1 [1]	DLL	Higher byte of CRC is transmitted first. Generated over preceding bytes
29	AF	CRC1 [0]		Same format as CRC0

Table 4: Packet SND_NR (not encrypted)

VIF Byte #	Value [hex]	Description
0	FD	Value 0xFD declares an “extension indicator”. This means that VIF coding is done in the following byte
1	BA	Value 0x3A defines that the data value should be interpreted “dimensionless”. MSB is set which means that another VIF byte is following
2	75	ValueInformationField for Exponent and OCR value [7] Extension bit **0**: no VIF byte following (i.e. next byte is value) 1: next byte is also a VIF byte [6:3] **14**: Multiplicative correction factor: 10**[6:3]-6 [2:0] Exponent with bias (bias -6) 5: Exponent = **5**-6 = -1 -> 10^{-1} factor; 689661*0.1 m^3 -> 68966.1 [dimensionless]

Table 5: VIF bytes coding for medium UNKNOWN

Examples of encrypted radio packet

Same data as example before using:

```
KEY   0x12345678912345678912345678 9ABCDE
IV    0xC4186416102301022727272727272727
```

IV = AES CBC Initial Vector according to FIPS 197 (LSB first): = M Field + A Field + 8 bytes Acces No

Raw data on air [hex] in transmission order (Bytes named Rx refers to CRCx).

```
1E 44 C4 18 64 16 10 23 01 02 1B D6 7A 27 00 10 85 57 00 05 A6 C1 31 B9 7B 26 68 C3 B9 D7 CF 29 0D
3B 7A DF 8F
```

Detailed description of example for encrypted data for electricity (energy in Wh)

Bold values are current values in example.

Byte #	Value encrypted [hex]	Value decrypted [hex]	Name	Layer	Description
0	1E	1E	L-Field	Link Layer (DLL)	0x1E=30 bytes following (without three CRCs a 2 bytes), in sum 1+25+3*2=37 bytes
1	44	44	C-Field		SND_NR (Send, No Response)
2	C4	C4	M-Field		ManufacturerID [0]: **0x18C4**="FFD" for FastForwarD (see www.dlms.com/flag)
3	18	18	M-Field		ManufacturerID [1]: Multibyte fields are transmitted lower byte first (except CRC)
4	64	64	A-Field		IdentNumber [0]: **0x23101664** (read as 8 BCD numbers)
5	16	16	A-Field		IdentNumber [1]:
6	10	10	A-Field		IdentNumber [2]
7	23	23	A-Field		IdentNumber [3]
8	01	01	A-Field		Version of Meter (constant, always 1 for EC), forms together with Manufacturer-ID, IdentNumber and Type a unique wM-Bus address
9	02	02	A-Field		Type of Meter (medium) 0x01 Oil **0x02** Energy (electricity) 0x03 Gas 0x07 Water 0x0F UNKNOWN (see VIF dimensionless)
10	1B	1B	CRC0 [1]		Higher byte of CRC is transmitted first. Generated over preceding bytes
11	D6	D6	CRC0 [0]		Polynom: $x^{16}+x^{13}+x^{12}+x^{11}+x^{10}+x^8+x^6+x^5+x^2+1$, start with 0, result inverted
12	7A	7A	CI-Field	Application Layer (APL)	ControlInfo: **0x7A**=Telegram with short Header (e.g. AC, ST, CW)
13	27	27	AC		ACcess number (incremented every synchronous packet which is not repeated)
14	00	00	ST		Signals error StaTe of meter ST **0x00**: no errors ST 0x04: battery low
15	10	10	CW [0]		CW=ConfigurationWord 0x8510 (controls encryption, here encrypted) [7:4] **0x1** (# encrypted blocks) [3:2] 00 (standard telegram) [1:0] 00 (hop counter=0)
16	85	85	CW [1]		[7:0]=[BAS0MMMM] [7] **1**: bidirectional because of T2 mode [6] **0**: accessible (meter remains after tx in rx state for a short period of time) 1: not accessible [5] **0**: asynchronous packet (spontaneous tx, e.g. by button press "--1--") 1: synchronous packet (send by periodical timer) [4] 0**:** reserved [3:0] 0x**5**: AES encryption
17	57	2F		AES Encryption Block	Decryption verification, should be 0x2F after successful decryption
18	00	2F			Decryption verification, should be 0x2F after successful decryption

Byte #	Value encrypted [hex]	Value decrypted [hex]	Name	Layer	Description
19	05	04	DIF	#0	DataInformationField for OCR value [7] Extension bit **0**: no DIF byte following (i.e. next byte is VIF) 1: next byte is also a DIF byte [6] Storage number (not used for EC) [5:4] Function field: **0**: instantaneous value (default for EC) 3: value during error state (e.g. repeating last OCR value due to error detection by EC) x: other codes not used by EC [3:0] **4**: 32 bit integer (means 4 bytes following after VIF). Other codes here not used by EC
20	A6	05	VIF		ValueInformationField for Exponent and OCR value (here energy in Wh) [7] Extension bit **0**: no VIF byte following (i.e. next byte is value) 1: next byte is also a VIF byte, in case of medium UKNOWN two VIF bytes follows, see Table 5: VIF bytes coding for medium UNKNOWN for details [6:3] **0**: Energy [Wh] used for electricity 2: Volume[m^3] used for gas and water x: other codes not used by EC [2:0] Exponent with bias (bias depends on unit, -3 for energy, -6 for volume) **5: Exponent = 5-3 = 2 -> 10^2 factor; 689661*100 Wh = 68966100 Wh -> 68966.1 kWh** 5: Exponent = **5**-6 = -1 -> 10^{-1} factor; 689661*0.1 m^3 -> 68966.1 m^3 (used for gas and water)
21	C1	FD	Value [0]		Value 32 bit integer **0x000A85FD**= 689661 (has to be scaled by exponent specified in VIF)
22	31	85	Value [1]		
23	B9	0A	Value [2]		
24	7B	00	Value [3]		
25	26	02	DIF		DataInformationField for Unique telegram identification [7] Extension bit **0**: no DIF byte following (i.e. next byte is VIF) 1: next byte is also a DIF byte [6] **0**: Storage number (not used for EC) [5:4] **0**: Function field: (not used by EC) [3:0] **2**: 16 bit integer (means 2 bytes following after VIF). Other codes here not used by EC
26	68	FD	VIFE		ValueInformationField for Unique telegram identification [7] Extension bit 0: no VIF byte following **1**: next byte is also a VIF byte [6:0] 0x7D: VIF Extension (next byte is VIF)
27	C3	08	VIF		ValueInformationField for Unique telegram identification [7] Extension bit

Byte #	Value encrypted [hex]	Value decrypted [hex]	Name	Layer	Description
					0: no more VIF following (i.e. next byte is value) **1**: next byte is also a VIF [6:0] **8**: Unique Telegram identifiCation
28	B9	B9	CRC1 [1]	DLL	Higher byte of CRC is transmitted first. Generated over preceding bytes
29	D7	D7	CRC1 [0]		Same format as CRC0
30	CF	0B	UTI[0]		Unique telegram identification (Byte 0, EC tx counter)
31	29	16	UTI[1]		Unique telegram identification (Byte 1, EC pic counter)
32	0D	0x2F	Filler		Filler byte to complete a 16 byte block
33	3B	0x2F	Filler		Filler byte to complete a 16 byte block
34	7A	0x2F	Filler		Filler byte to complete a 16 byte block
35	DF	DF	CRC2 [1]	DLL	Higher byte of CRC is transmitted first. Generated over preceding bytes
36	8F	8F	CRC2 [0]		Same format as CRC0

Table 6: Packet SND_NR (encrypted)

Unique Telegram Identification

After the OCR value a “Unique Telegram Identification” (UTI) is added. This value is not necessarily incremented or decremented it is just important that it varies for every RF packet sent. This guarantees that even when the OCR value remains stable the payload is different. This is only important when the RF packet is encrypted, otherwise the used encryption key data could be easier reengineered. The UTI is added to the data packet even when encryption is not used.

Here are the relevant parts as citations from the specifications.

OMS-Spec_Vol2_Primary_v301.pdf

http://oms-group.org/fileadmin/pdf/OMS-Spec_Vol2_Primary_v301.pdf

4.2.5.3 Initialisation Vector for Encryption Mode 5

To make sure that the encrypted and the unencrypted section of the telegram came from the same meter, this initialisation vector contains in its lower 8 bytes the meter identification (from link or application layer, depending on the CI-field (refer to chapter 4.2.1)).

When the consumption value does not change, this could be detected by reception of periodical telegrams from the meter. To protect the consumer from unauthorised observation of such a situation with zero consumption, each generated telegram shall change with every periodical transmission. This can be implemented either by a timestamp or a counter in the first block or by an increased access number (Acc. no.), which is part of the initialisation vector (copy 8 times the access number to the upper 8 bytes). Due to the block chaining mode CBC both methods will influence all other encrypted blocks. Note that after 255 transmissions the zero consumption is detectable again even if the access number was used. The access number will be incremented with each synchronous transmission only.

Therefore it is recommended to add a time stamp or a sequence number (VIFE "Unique telegram identification (previously named 'Access Number (transmission count)')") to the telegram content.

EN_13757-3_2011_05.pdf (The norm is not publicly available, here German's version)

Tabelle 27 — Besondere VIF-Codes

1111 1101 (FDh): zweite Erweiterung der VIF-Codes, tatsächliches VIF ist im ersten VIFE gegeben und mit Hilfe von Tabelle 28 in 7.4 codiert

Tabelle 28 — Haupttabelle für die VIFE-Code-Erweiterung

E000 1000 (08h): eindeutige Telegrammidentifikation (ehemals bezeichnet als „Zugriffsnummer (Übertragungszahl)“)

Examples of an installation request (SND_IR)

After successful OCR installation a SND_IR (SeND Installation Request) procedure is started.

Raw data on air [hex] in transmission order.

```
0E 46 C4 18 64 16 10 23 01 07 31 7D 7A 00 00 00 80 2B 33
```

Byte #	Value [hex]	Name	Layer	Description
0	0E	L-Field	Link Layer (DLL)	0x0E=14 bytes following (without two CRCs a 2 bytes), in sum 1+14+2*2=19 bytes
1	46	C-Field		SND_IR (SeND Installation Request)
2	C4	M-Field		ManufacturerID [0]: **0x18C4**="FFD" for FastForwarD (see www.dlms.com/flag)
3	18	M-Field		ManufacturerID [1]: Multibyte fields are transmitted lower byte first (except CRC)
4	64	A-Field		IdentNumber [0]: **0x23101664** (read as 8 BCD numbers)
5	16	A-Field		IdentNumber [1]
6	10	A-Field		IdentNumber [2]
7	23	A-Field		IdentNumber [3]
8	01	A-Field		Version of Meter (constant, always 1 for EC), forms together with Manufacturer-ID, IdentNumber and Type a unique wM-Bus address
9	07	A-Field		Type of Meter (medium) 0x02 Energy (electricity) 0x03 Gas **0x07** Water
10	31	CRC0 [1]		Higher byte of CRC is transmitted first. Generated over preceding bytes
11	7D	CRC0 [0]		Polynom: $x^{16}+x^{13}+x^{12}+x^{11}+x^{10}+x^{8}+x^{6}+x^{5}+x^{2}+1$, start with 0, result inverted
12	7A	CI-Field	Appli-cation Layer (APL)	ControlInfo: **0x7A**=Telegram with short Header (e.g. AC, ST, CW)
13	00	AC		ACcess number (not changed, because no synchronous packet)
14	00	ST		Signals error StaTe of meter ST **0x00**: no errors
15	00	CW [0]		CW=ConfigurationWord 0x8000 (controls encryption, SND_IR is never encrypted) [7:4] 0x**0** (# encrypted blocks) [3:2] **00** (standard telegram) [1:0] **00** (hop counter=0)
16	80	CW [1]		[7:0]=[BAS0MMMM] [7] **1**: bidirectional because of T2 or S2 mode [6] **0**: accessible (meter remains after tx in rx state for a short period of time) 1: not accessible [5] **0**: asynchronous packet (spontaneous tx) 1: synchronous packet [4] **0:** reserved [3:0] 0x**0**: no AES encryption
17	2B	CRC1 [1]	DLL	Higher byte of CRC is transmitted first. Generated over preceding bytes
18	33	CRC1 [0]		Same format as CRC0

Table 7: Packet SND_IR

Examples of an installation confirmation (CNF_IR)

A collector may respond to a SND_IR with a CNF_IR (CoNFimation Installation Request) when the sensor is granted.

Raw data on air [hex] in transmission order. “??” bytes are the CRC bytes which is not calculated in example.

```
16 06 A2 5C 99 40 13 00 4A 31 ?? ?? 80 64 16 10 23 C4 18 01 02 00 2C 00 C0 ?? ??
```

Byte #	Value [hex]	Name	Layer	Description
0	16	L-Field	Link Layer (DLL)	0x16=22 bytes following (without two CRCs a 2 bytes), in sum 1+22+2*2=27 bytes
1	06	C-Field		CNF_IR (CoNFirm Installation Request)
2	A2	M-Field		ManufacturerID [0]: **0x5CA2**="WEB" for Webolution GmbH & Co. KG (see www.dlms.com/flag)
3	5C	M-Field		ManufacturerID [1]: Multibyte fields are transmitted lower byte first (except CRC)
4	99	A-Field		IdentNumber [0]: **0x4A991349** (read as 8 BCD numbers), A-Field of collector
5	40	A-Field		IdentNumber [1]:
6	13	A-Field		IdentNumber [2]
7	00	A-Field		IdentNumber [3]
8	4A	A-Field		Version of collector
9	31	A-Field		Type of collector 0x31 “Reserved for communication controller”
10	??	CRC0 [1]		Higher byte of CRC is transmitted first. Generated over preceding bytes
11	??	CRC0 [0]		Polynom: $x^{16}+x^{13}+x^{12}+x^{11}+x^{10}+x^{8}+x^{6}+x^{5}+x^{2}+1$, start with 0, result inverted
12	80	CI-Field	Appli-cation Layer (APL)	ControlInfo: **0x80**=Telegram long transport layer (e.g. A-Field, M-Field, A-Field, AC, ST, CW)
13	64	A-Field		IdentNumber [0]: **0x23101664** (read as 8 BCD numbers), A/M-Field of meter which sent SND_IR
14	16	A-Field		IdentNumber [1]
15	10	A-Field		IdentNumber [2]
16	23	A-Field		IdentNumber [3]
17	C4	M-Field		ManufacturerID [0]: **0x18C4**="FFD" for FastForwarD (see www.dlms.com/flag)
18	18	M-Field		ManufacturerID [1]
19	01	A-Field		Version of Meter (constant, always 1 for EC), forms together with Manufacturer-ID, IdentNumber and Type a unique wM-Bus address
20	02	A-Field		Type of Meter (medium) 0x02 Energy (electricity) 0x03 Gas **0x07** Water
21	00	AC		ACcess number (not changed, because no synchronous packet)
22	2C	ST		Signals error StaTe of meter ST **0x2C**: RSSI (Receive signal strength indication) of received SND_IR ST = 0: no RSSI value available or wired communication ST = 1: RSSI is <= -128 dBm ST = 2..62: -130 dbm + 2 * RSSI ST = 63: RSSI > -6 dBm RSSI = (-ST+130dbm)/2 * -(1); here RSSI = (-44+130dbm)/2 * -(1) = -43 dbm
23	00	CW [0]		CW=ConfigurationWord 0xC000 (controls encryption, CNF_IR is never encrypted) [7:4] 0x**0** (# encrypted blocks) [3:2] **00** (standard telegram) [1:0] **00** (hop counter=0)
24	C0	CW [1]		[7:0]=[BAS0MMMM] [7] **1**: bidirectional because of T2 or S2 mode

Byte #	Value [hex]	Name	Layer	Description
				[6] **1**: accessible 0: not accessible [5] **0**: asynchronous packet (spontaneous tx) 1: synchronous packet [4] **0:** reserved [3:0] 0x**0**: no AES encryption
25	??	CRC1 [1]	DLL	Higher byte of CRC is transmitted first. Generated over preceding bytes
26	??	CRC1 [0]		Same format as CRC0

Table 8: Packet CNF_IR

Manual decryption

For en- or decryption experiments one can use online tools like http://aes.online-domain-tools.com

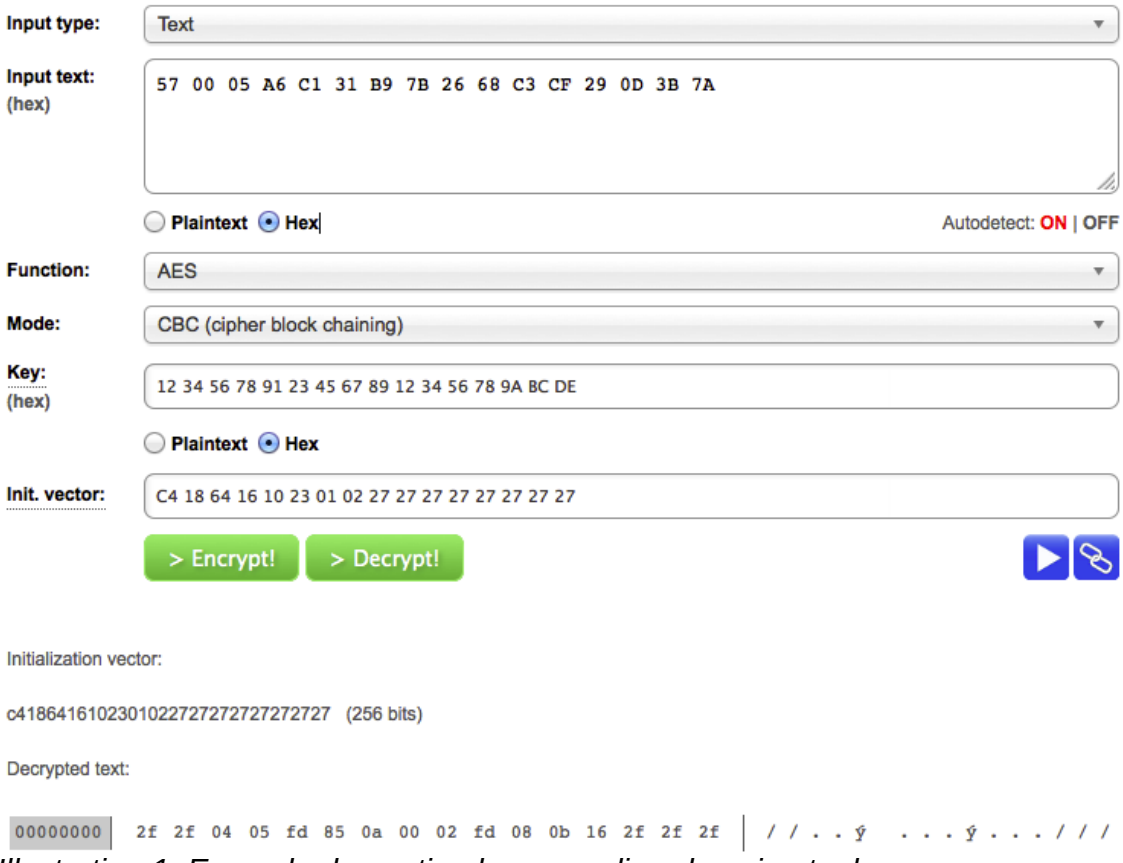

Illustration 1: Example decryption by aes.online-domains-tools.com

References

“Communication systems for meters and remote reading of meters.” Part 4: Wireless meter readout (Radio meter reading for operation in the 868 MHz to 870 MHz SRD band); German version DIN EN 13757-4, 2011.

“Open metering system specification - OMS": http://www.oms-group.org

Communication systems for and remote reading of meters - Part 3: Dedicated application layer; German version EN 13757-3:2011

History

Date	Author	Version	Changes
28th Jun 2012	SPR	0.1	Initial
26th Nov 2012	SPR	0.2	Replaced SA by EC (EnergyCam). Added manual installation procedure. Added example packet
14th Jan 2013	SPR	0.3	Better text contrast
10th Oct 2013	SPR	0.4	Removed preliminary status, added more installation docu
29th Jan 2014	SPR	0.5	Added unique transmission counter (Used starting from Firmware version “Beta12”)
17th Mar 2014	SPR	0.6	Cosmetic
4th Apr 2014	SPR	0.7	Renamed “Unique transmission counter” to “Unique Telegram Identification”. Added chapter “Unique telegram identifier”
4th Jun 2014	SPR	1.0	Added “battery low” and “value during error state” codings. Added many comments
3rd Jul 2014	SPR	1.1	Corrected CRC polynom (exponents are not superscript)
20th Jul 2014	SPR	1.2	Fixed typo: OCR value 869661 instead of 689661
2nd Oct 2014	SPR	1.3	Added example for AES encrypted packet
28th Oct 2014	SPR	1.4	Document renamed,
16th Dec 2014	SPR	1.5	Corrected comment for ConfigurationWord in encrypted case
07th Jan 2015	SPR	1.6	Added packet SND_IR
15th Jan 2015	SPR	1.7	Added packet CNF_IR
29th Jan 2015	SPR	1.8	Added medium Oil and UNKNOWN. Added VIF bytes coding for medium UNKNOWN (VIF dimensionless)

Table 9: History