OpenlinkIQ® Specification

Document Content

This document specifies the OpenlinkIQ® communication protocol. This information is published and available on openlinkiq.org.

The information within this document ("OpenlinkIQ Specification") is the property of KAMSTRUP A/S. The OpenlinkIQ Specification specifies the OpenlinkIQ Communication Protocol and is made available by KAMSTRUP A/S, free of charge, to any IMPLEMENTER for the purpose of implementing products including the OpenlinkIQ Communication Protocol. KAMSTRUP A/S reserves the right to release and publish new versions of the OpenlinkIQ Specification without further notice.

THE OPENLINKIQ SPECIFICATION IS PROVIDED "AS IS" WITHOUT ANY WARRANTIES. Any use of the OpenlinkIQ Specification is on the IMPLEMENTER'S own risk. The OpenlinkIQ Specification is provided with no warranty of quality or performance. KAMSTRUP A/S AND ITS AFFILIATES SHALL HAVE NO LIABILITY TO ANY IMPLEMENTER OR THIRD PARTY FOR DAMAGES DIRECTLY OR INDIRECTLY ARISING FROM THE USE OF THE OPENLINKIQ SPECIFICATION OR PRODUCTS INCLUDING IMPLEMENTATIONS OF THE OPENLINKIQ COMMUNICATION PROTOCOL.

The OpenlinkIQ Specification does not contain or create any warranties by Kamstrup A/S and its affiliates regarding compliance to national or international laws, regulations, directives and standards such as, but not limited to, telecommunication regulations, safety regulations and metering regulations. KAMSTRUP A/S AND ITS AFFILIATES DISCLAIM ALL LIABILITY TO ANY IMPLEMENTER OR THIRD PARTY FOR NONCOMPLIANCE WITH SUCH LAWS, REGULATIONS, DIRECTIVES AND STANDARDS.

The OpenlinkIQ Specification may in whole or in part be subject to third party intellectual property rights including, but not limited to, patent rights, copyrights and/or trademark rights. KAMSTRUP A/S AND ITS AFFILIATES SHALL NOT BE HELD RESPONSIBLE FOR INFORMING OR FOR NOT INFORMING ABOUT SUCH THIRD PARTY INTELLECTUAL PROPERTY RIGHTS. KAMSTRUP A/S AND ITS AFFILIATES DISCLAIM ALL LIABILITY TO ANY IMPLEMENTER OR THIRD PARTY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS RELATED TO THE USE OF THE OPENLINKIQ SPECIFICATION OR PRODUCTS INCLUDING IMPLEMENTATIONS OF THE OPENLINKIQ COMMUNICATION PROTOCOL, INCLUDING DERIVED OR MODIFIED VERSIONS OF THE OPENLINKIQ COMMUNICATION PROTOCOL.

BY USE OF THE OPENLINKIQ SPECIFICATION AND/OR THE OPENLINKIQ COMMUNICATION PROTOCOL ANY IMPLEMENTER OR THIRD PARTY WAIVES ANY CLAIM AGAINST KAMSTRUP A/S AND ITS AFFILIATES RELATED TO ANY USE OF THE OPENLINKIQ SPECIFICATION AND OPENLINKIQ COMMUNICATION PROTOCOL.

KAMSTRUP A/S as the proprietor of the following patents: EP 3 072 308 B1, US 9 912 441 B2, CN 105 765 991 B and EP 3 482 511 A1 hereby declares to grant to any IMPLEMETOR a worldwide, irrevocable, non-exclusive and free of charge license on fair, reasonable and nondiscriminatory terms and conditions to the mentioned patents to make, use and sell products including implementations of the OpenlinkIQ Communication Protocol according to the OpenlinkIQ Specification.

KAMSTRUP A/S as the proprietor of the following trademarks: OpenlinkIQ® and the OpenlinkIQ logo declares to grant a worldwide, irrevocable, non-exclusive and free of charge license to the mentioned trademarks to any IMPLEMENTER to mark and promote products including implementations of the OpenlinkIQ Communication Protocol according to the OpenlinkIQ Specification.

A license to the abovementioned patents and trademarks to an IMPLEMENTER is granted under the condition that all products including the OpenlinkIQ Communication Protocol made, used or sold by the IMPLEMENTER complies to the OpenlinkIQ Specification in its entirety.

FURTHER, A LICENSE TO THE ABOVEMENTIONED PATENTS AND TRADEMARKS IS GRANTED UNDER THE CONDITION OF THE IMPLEMENTER'S WILLINGNESS TO GRANT A LICENSE ON THE ABOVEMENTIONED TERMS AND CONDITIONS TO ANY IMPLEMENTER OWNED INTELLECTUAL PROPERTY RIGHT WHICH IS ESSENTIAL TO THE IMPLEMENTATION OR TO THE OPTIMAL IMPLEMENTATION OF THE OPENLINKIQ COMMUNICATION PROTOCOL ACCORDING TO THE OPENLINKIQ SPECIFICATION.

KAMSTRUP A/S reserves the right to introduce a product certification procedure to ascertain that certified products including the OpenlinkIQ Communication Protocol complies with the OpenlinkIQ Specification and to make certification of products including implementations of the OpenlinkIQ Communication Protocol a further condition for granting licenses to the abovementioned intellectual property rights.

This declaration by KAMSTRUP A/S of a license to intellectual property rights does not include any other IPR than the abovementioned. Explicitly the trademarks Kamstrup® and LinkIQ® and any patents owned by KAMSTRUP A/S other than EP 3 072 308 B1, US 9 912 441 B2, CN 105 765 991 B and EP 3 482 511 A1 are excluded from the license.

KAMSTRUP A/S reserves the right to change the licensing terms and conditions without notice.

Kamstrup A/S
Industrivej 28
8660 Skanderborg
Denmark

Contents

1 Introduction

There is a need in the industry to get access to a more robust long range solution compared to existing standard solutions, both within the utility segment and the submetering segment. The advantages of such a radio communication technology will also be of interest for a variety of new, small battery powered sensor applications.

OpenlinkIQ® offers a robust radio communication network, by using intelligent coding of the transmitted signal, which again ensures reliable data delivery, even if communication is disturbed by interference from radio transmissions from other devices.

Built upon open metering communication standards (M-Bus), combined with many years of experience with battery operated solutions, OpenlinkIQ® takes the full benefit from both interoperability and advanced radio communication technology.

With OpenlinkIQ® it is the intention to make this advanced technology available to the community as a genuine open and free-to-use long range radio technology, with a performance matching many other new LPWAN technologies on the market, but with a very low power consumption. Often these new LPWAN technologies are either proprietary solutions or covered by IPR. OpenlinkIQ® will demonstrate a path for enabling any end device with efficient and advanced communication technology.

OpenlinkIQ® is designed to coexist with current deployed solutions based on known LPWAN technologies, such as wireless M-Bus, LoRaWAN®, SigFox®, etc.

This document specifies the OpenlinkIQ® communication protocol, consisting of a radio interface, physical layer and frame structure. OpenlinkIQ® is a long range protocol used for efficient transfer of data from IoT devices (e.g. meter readings or sensor data) to data collectors.

This specification details a format, specifically designed to carry M-Bus data. OpenlinkIQ® facilitates transmission of M-Bus data via a standardized M-Bus adaptation layer (MBAL).

A system description of the typical application environment is shown in chapter 2. A technical brief of which radio-band(s) are used for transmissions is shown in chapter 3.

Chapters 4 through 11 specifies the requirements for fulfilling the OpenlinkIQ® communication protocol.

1.1 References

Reference	Description/Link
[1]	FILE100000553-55123278 OpenlinkIQ® Implementation Guide, Release D, September 2023 – openlinkiq.org
[2]	EN 13757-3 Communication systems for meters – *Part 3: Application protocols*, CEN/TC294:2018
[3]	EN 13757-4 Communication systems for meters – *Part 4: Wireless M-Bus communication*, CEN/TC294:2019
[4]	EN 13757-7 Communication systems for meters – Part 7: *Transport and security services*, CEN/TC294:2018
[5]	EN 13757-8 Communication systems for meters – Part 8: *Adaptation layer*, CEN/TC294:2023
[6]	EN 300 220-2 Short Range Devices (SRD) operating in the frequency range 25 MHz to 1 000 MHz; Part 2: *Harmonised Standard for access to radio spectrum for non-specific radio equipment*, ETSI:2018-06
[7]	EN 301 489-1 ElectroMagnetic Compatibility (EMC) standard for radio equipment and services; Part 1: *Common technical requirements; Harmonised Standard for ElectroMagnetic Compatibility*, ETSI:2019-11

[8]	ETSI – EN 301 489-3 ElectroMagnetic Compatibility (EMC) standard for radio equipment and services; Part 3: *Specific conditions for Short Range Devices (SRD) operating on frequencies between 9 kHz and 246 GHz; Harmonised Standard for ElectroMagnetic Compatibility*, ETSI:2023-01
[9]	IEC 62 311 *Assessment of electronic and electrical equipment related to human exposure restrictions for electromagnetic fields (0 Hz to 300 GHz)*, IEC:2020
[10]	Liang Li, *Analysis of Low Power Implementational Issues of Turbo-like Codes in Body Area Networks, chapter 2: "Turbo-like Code Solutions in BANs"*, nine-month report, University of Southampton, November 2009.
[11]	*oms-group.org – Open Metering System organisation*

1.2 Change Log

Rev	Date	Description
A0	2022-05-01	Initial release
C	2023-07-04	Added section 3.3 for regional interfaces and modified test vectors in section 8.2.
D	2023-09-29	Updated references. Precision of octal values in section 7.1

1.3 Abbreviations

3G	3rd generation of wireless mobile telecommunications technology
AFL	M-bus Authentication and Fragmentation Layer
AMI	Advanced Metering Infrastructure
APL	M-Bus Application Layer
CEN	European Committee for Standardization
CH	Coded Header
CI	Control Information
CRC	Cyclic Redundancy Check
DC	Duty Cycle
ERP	Effective Radiated Power
ETSI	European Telecommunications Standards Institute
FEC	Forward Error Correction
FSK	Frequency-Shift Keying
GMSK	Gaussian Minimum-Shift Keying
HES	Head End System
HTTPS	Hypertext Transfer Protocol Secure
IEC	International Electrotechnical Commission
IP	Internet Protocol
ISM	Industrial, Scientific, and Medical
LPWAN	Low Power Wide Area Network
LSB	Least Significant Bit (or byte)
LTE	Long Term Evolution (4G)
MBAL	M-Bus Adaptation Layer
MSB	Most Significant Bit (or byte)
MSK	Minimum-Shift Keying
OMS	Open Metering System
RF	Radio Frequency
RSC	Recursive Systematic Convolution
SRD	Short Range Devices
TCP	Transmission Control Protocol
TI	Transmission Interval
TPL	M-Bus Transport Layer
UMTS	Universal Mobile Telecommunications System

2 System description

OpenlinkIQ® is optimized for providing long range robust connectivity and an ultra-long battery life for large scale advanced meter and sensor infrastructure (AMI) solutions. The OpenlinkIQ® protocol is designed for a star topology radio network which comprises a large number of end devices (sensors, utility meters and alike) and a small number of infrastructure components (data collectors). OpenlinkIQ® is thus characterised as a genuine LPWAN technology.

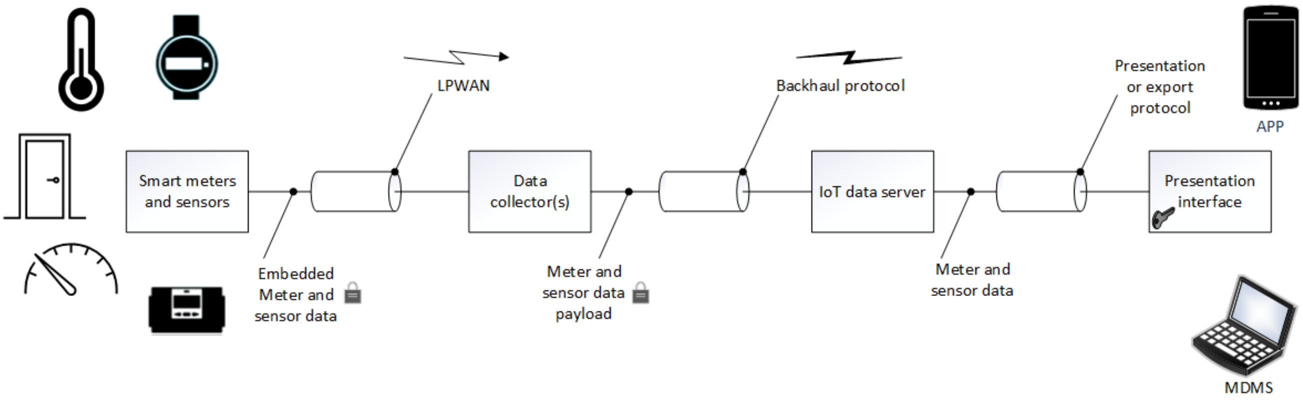

Figure 1 Eco-system example.

Figure 1 shows an example of an eco-system for collecting smart meter and sensor data. The smart meters typically communicate via LPWAN to a number of data collectors or base stations. The meter or sensor readings are often encrypted and then embedded in the radio communication telegram defined by the LPWAN.

The data collectors forward meter and sensor data to IoT data servers or head-end systems (HES) responsible for the storage of data. Meter and sensor data can then be retrieved with the use of a valid key, or data can be exported to a meter data management system (MDMS) thus receiving meter data for the particular end devices they are responsible for. The typical customer is a utility that uses different services and applications to read consumption or status values for their business.

The OpenlinkIQ® protocol is suitable to be used for any sensor types needing long range transmissions in a power efficient way. From the data collector level and up, OpenlinkIQ® systems do not prescribe specific IoT data servers to be used. Any open IoT platform can be supported.

End device transmissions are distributed in frequency and time throughout the occupied bandwidth. Distribution of transmission frequency and transmission time among the end devices enables an advanced receiver to process multiple uplink transmissions simultaneously. This ensures that even very large AMI systems can support a high quality of service while operating in license exempt ISM bands with multiple other actors.

2.1 Protocol stack

The OpenlinkIQ® protocol takes care of all lower layers of the radio communication link (physical layer and data link layer). The M-Bus upper layers, APL [2] and TPL/AFL (EN 13757-7 [4]), are concatenated with the 'MBAL' field (EN 13757-8 [5]) and embedded in OpenlinkIQ® for transmission. Figure 2 shows the layers across the system components.

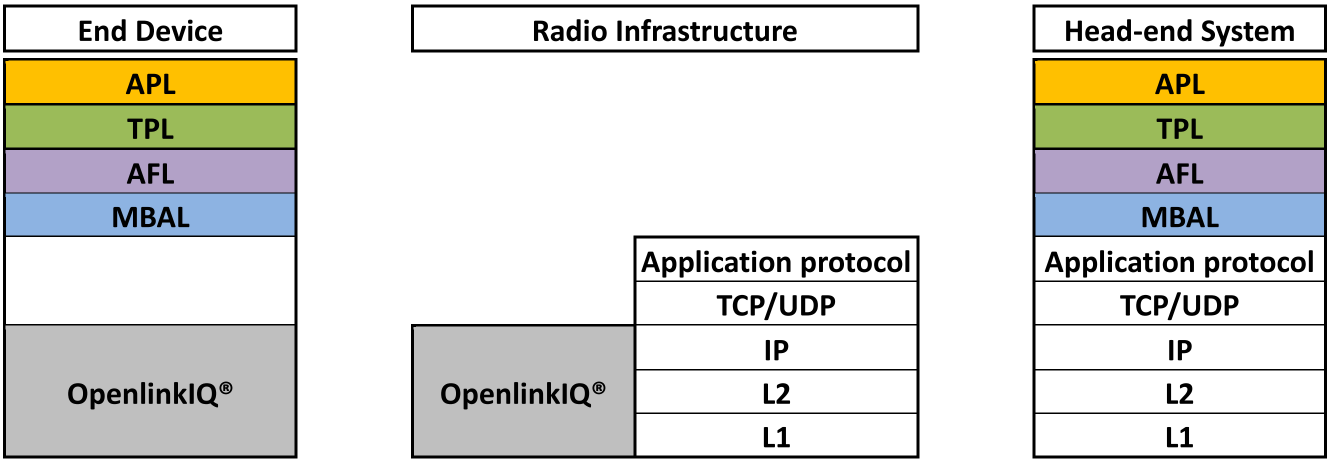

Figure 2 M-Bus over OpenlinkIQ® encapsulation.

OpenlinkIQ® constitutes the LPWAN technology to transmit data from end devices to infrastructure components. The upper M-Bus layers (APL/TPL/AFL) are enabled via a 'CI' field mechanism detailed in EN 13757-7 [4]. End-to-end security and replay protection is handled entirely in those M-Bus layers.

The connection between the infrastructure component(s) and HES is handled by other protocols (e.g. TCP/IP). The M-Bus upper layers and 'MBAL' field is encapsulated in a suitable application protocol (e.g. HTTPS) and transmitted to the HES (not further discussed in this specification).

3 Technical Brief

The physical layer of OpenlinkIQ® conforms to EU Commission Decision (EU) 2022/180 (amendment to EU Commission Decision 2006/771/EC on harmonization of the radio spectrum for use by short-range devices).

OpenlinkIQ® uses radio communication in a shared frequency band that is license exempt and free to use under the conditions depicted in the following.

3.1 Radio interface

Of the available bands defined in EU Commission Decision (EU) 2022/180 the following has been selected for OpenlinkIQ® radio transmissions.

Band number: 48 (EU wide harmonised)

- Frequency range: 868,0 MHz to 868,6 MHz (license free ISM band)
- Permitted duty cycle: 1 % (measured over any 1-hour interval)
- Max ERP: 14 dBm (25 mW)

Other RF parameters and requirements shall be derived from [6].

3.2 Conformity

Any SRDs, operating on the radio interface defined in section 3.1, shall be in conformance with the following standards:

Radio Equipment Directive (RED – 2014/53/EU)

- EN 300 220-2 [6]
- EN 301 489-1 [7]
- EN 301 489-3 [8]

Low Voltage Directive (LVD – 2014/35/EU):

- IEC 62 311 [9]

For an end device to be genuine OpenlinkIQ® compliant, the content of the present document shall be implemented.

3.3 Regional radio interfaces

For implementing OpenlinkIQ® in other regions outside EU, please contact openlinkiq.org.

4 Physical constraints

A visualization of the channel allocation used for OpenlinkIQ® is shown in Figure 3. The channel allocation for the transmission of OpenlinkIQ® is shown in the lower band. It is also shown that OpenlinkIQ® operates in a frequency band apart from the wireless M-Bus mode C channel (see [3] for more information).

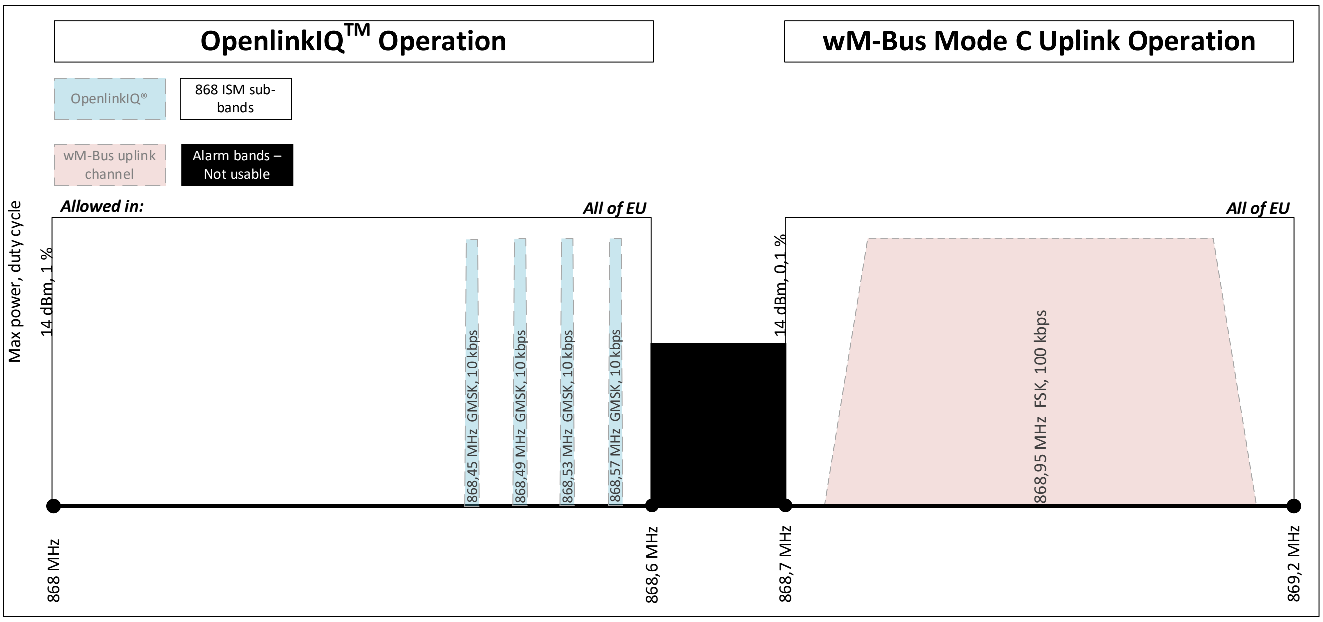

Figure 3 OpenlinkIQ® band and channel allocation visualization.

4.1 Centre frequencies

Four narrowband channels have been selected to support OpenlinkIQ® in the selected band. These are shown in Table 1.

Channel number	Centre Frequency Set
0	868,45 MHz
1	868,49 MHz
2	868,53 MHz
3	868,57 MHz

Table 1 OpenlinkIQ® channel definitions.

Each of the channels shall have a maximum channel width of 40 kHz. To be compliant to this specification the transmitter parameters shown in Table 2 must be met.

4.2 Transmitter parameters

For the end device transmissions to be efficiently received in data collectors using coherent receivers the requirements in Table 2 shall be implemented.

Parameter	Requirement	Comment
Centre frequency precision	± 10,5 ppm	Over product lifetime

<table>
<tr><td></td><td colspan="2"></td><td>The precision is over XTAL temperature range, aging and initial XTAL tolerance.
The resolution frequency for the transmitter synthesizer is included in the centre frequency precision.</td></tr>
<tr><td>Centre frequency stability Transitional Frequency Dynamic Drift</td><td colspan="2">± 300 Hz/sec</td><td>The (peak) drift allowed in the transmitter centre frequency during transmission of the preamble and synchronization word.</td></tr>
<tr><td>Centre frequency stability Established Frequency Dynamic Drift</td><td colspan="2">± 200 Hz/sec</td><td>The (peak) drift allowed in the transmitter centre frequency during all of the transmission following the synchronization bits.</td></tr>
<tr><td>Modulation</td><td colspan="2">Precoded GMSK</td><td>The data bits are required to be precoded as specified in 9.</td></tr>
<tr><td>Line coding</td><td colspan="2">Differential precoding</td><td>See annex 9 for more details.</td></tr>
<tr><td>Bit constellation</td><td>-dev
0b0</td><td>+dev
0b1</td><td></td></tr>
<tr><td>Gaussian filter time bandwidth product</td><td colspan="2">0,5</td><td></td></tr>
<tr><td>Symbol rate</td><td colspan="2">10 ksps
(± 30 PPM)</td><td></td></tr>
<tr><td>Data rate</td><td colspan="2">10 kbps
(± 30 PPM)</td><td></td></tr>
<tr><td>Deviation</td><td colspan="2">2,5 kHz</td><td>Symbol rate / 4</td></tr>
<tr><td>End device output max power</td><td colspan="2">14 dBm ERP</td><td>The legal limit for band 48 in EC Decision 2022/180/EU is 14 dBm. The end device must, with antenna, transmit as close as possible to this legal limit without exceeding it.</td></tr>
<tr><td>FEC encoder</td><td colspan="2"></td><td></td></tr>
<tr><td>Turbo code encoder</td><td colspan="2">Rate ½</td><td></td></tr>
<tr><td>Turbo code encoder</td><td colspan="2">Rate 1/3</td><td></td></tr>
</table>

Table 2 RF transmitter parameters.

Coherent reception is the optimal receiver architecture for achieving the best possible link budget on the long range channels.

The parameters in Table 2 reflect the necessity of having a stable oscillator frequency under all conditions, as coherent receivers need the phase information of the transmitter (the carrier phase) to recover the transmitted data at receiver side. A detailed explanation of the dynamic drift parameters is given in [1].

4.3 Media access techniques

4.3.1 Channel selection

All frames shall be transmitted using the four channels shown in 4.1.

Channel usage shall be uniformly distributed. As an example, channels could be selected in a simple cyclic sequence.

The receiving data collector shall be able to receive on all defined channels simultaneously.

4.3.2 Transmission intervals

Data can be transmitted as scheduled (metering data, etc.) or as events.

4.3.2.1 Scheduled transmissions

Scheduled data transmissions should have as long transmission intervals TI as the application allows for. A certain transmission loss shall be foreseen due to interference and noise conditions in the environment. Depending on the use case, redundant transmissions or use of data redundancy in the M-bus frame (e.g. by using M-bus compact profile) may be preferred.

The minimum recommended interval for transmission of scheduled messages is $TI = 600\ s$. The random timing described in section 4.3.3 shall be applied between two subsequent transmissions independent from the selected channel.

4.3.2.2 Event based transmissions

For event-based transmission, successive transmissions may be performed as fast as the power budget or power supply construction allows for (best effort). A transmission attempt may consist of up to 8 successive transmissions following the description in section 4.3.1.

For reporting events the sequence may (optionally) be repeated for robustness a number of times keeping the maximum constraints shown in Table 3.

Attempt #	Minimum interval between attempts	Comment
1	-	Initial transmission of the attempt Up to 8 successive transmissions
2	180 s	Up to 4 successive transmissions
3	180 s	1 transmission only
> 3	600 s	1 transmission only
Cease		≤ 3.600 s after first transmission of the original event

Table 3 Repetition scheme for transmission attempts concerning the same event.

If new events occur in the end device while executing a repetition scheme these new events should be transmitted as soon as possible.

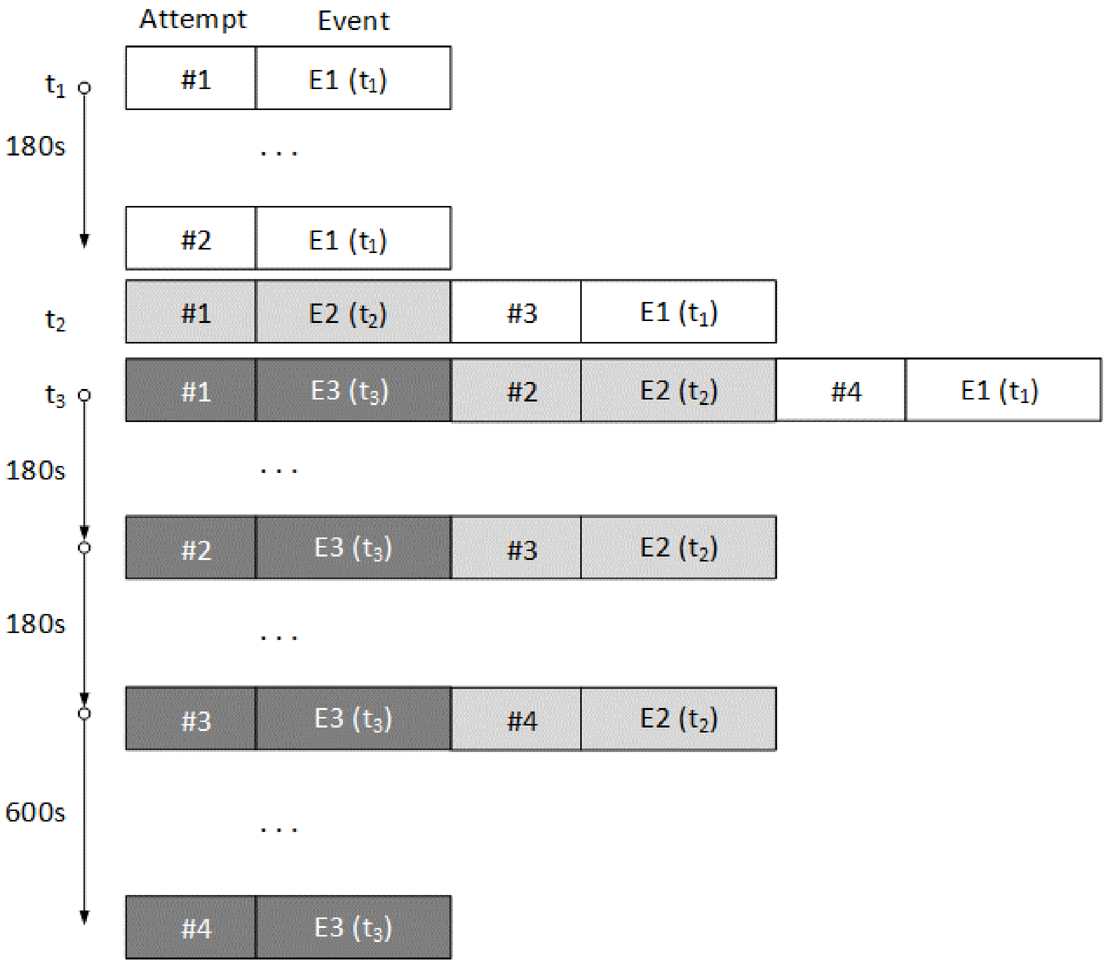

Figure 4 Illustration of three independent events (E1, E2 and E3) occurring at (t_1, t_2 and t_3).

This means the transmission interval is always controlled by the event which is allowed to be transmitted with the shortest interval between attempts (the newest or most actual event). To save the number of messages transmitted, events that are already in a repetition cycle/scheme should be piggybacked on the most recent event if possible, including time stamps. This may cause historical events to terminate their own repetition schedule sooner. Figure 4 shows an example of the frames transmitted for three independent events. Each attempt consists of a single transmission and is repeated 4 times.

See also the recommendation for priority bit usage in the MBAL frame ‘control’ field (6.2.1).

In general, the device overall duty cycle shall always be kept below the permitted duty cycle defined in section 3.1, which may force scheduled transmissions only, in case an excessive number of events should occur in the end device. As an example; if an end device implements the maximum allowed event repetition scheme from Table 3, each transmission having a duration of 200 ms (see also capacity considerations in [1]), then each new event (per hour) takes up a duty cycle of:

$$DC = \frac{0{,}2\ s\ \cdot 18}{3.600\ s} \cdot 100\% = 0{,}1\%$$

It is not recommended to plan for an average duty cycle higher than this. If a device is supposed to report many new events per hour, a more relaxed repetition scheme should be implemented.

4.3.2.3 Installation messages

If an end device has the capability to transmit install frames (for end device registration in a head-end system), the first transmission of this message shall only follow a manual event (e.g. push button). A repetition of the installation message is recommended as specified in Table 4.

Transmission #	Repetition interval	Comment
1	-	Following a manual event
2 – 6	30 s – 60 s	Recommended minimum number of repetitions for efficient commissioning is 6.
> 6	30 s – 60 s	Optional
Cease		≤ 3.600 s after first transmission

Table 4 Recommended repetition scheme for (optional) installation frames.

4.3.3 Randomization

After each scheduled transmission the following transmission is scheduled after the nominal interval but with a randomly selected jitter within ±8,33%.

5 Physical layer frame

5.1 Overview

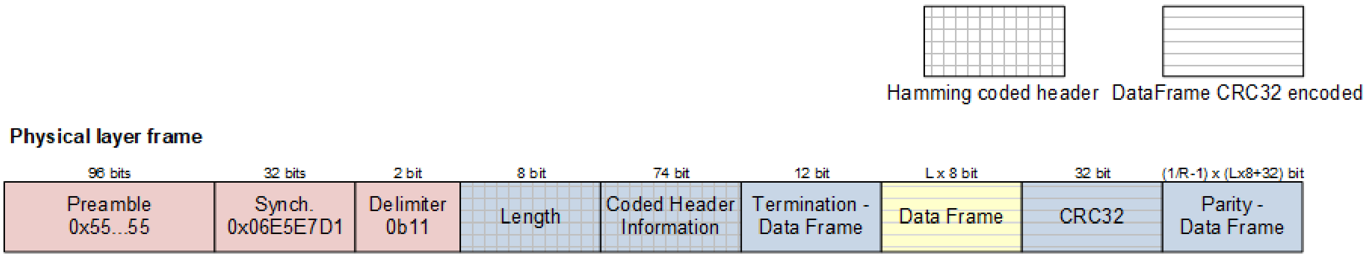

Figure 5 Physical layer frame structure.

The physical layer frame structure for OpenlinkIQ® is shown in Figure 5. The following sections explain the fields in detail.

5.2 Train Up

Before the header content, a train up containing a preamble and a synchronization word is necessary for the reception. A delimiter is inserted to assure that the resultant physical frame length is an integer number of bytes.

Preamble	Synch. Word	Delimiter
96 bit	32 bit	2 bit

The field definitions are described in the following sections.

5.2.1 Preamble

The preamble is 96 symbols, i.e. 0x555555555555555555555555.

5.2.2 Synch. Word

A synchronisation word of 32 bits is used with the coded header defined in section 5.3:

00000110 11100101 11100111 11010001 (06, E5, E7, D1)

5.2.3 Delimiter

The ‘delimiter’ field shall have the value 0b11.

5.3 Coded Header

The coded header (CH) precedes the embedded data frame. The CH contains the Hamming coded length field and Turbo coding rate for the embedded data frame.

Length field	Coded Header Information field
8 bit	74 bit

Table 5 Coded header structure.

5.3.1 Length field

The ‘Length’ field specifies the length L of the ‘Data frame’. The field is specified as follows:

BIT 7 ... BIT 0
Length of the Data frame (12 ... 251 bytes)

5.3.2 Coded Header Information field

The coded header is specified by the tables in annex 11.

5.3.3 Decoding the coded header

After receiving the bits for the ‘Length’ and ‘Coded Header Information’ fields these are compared with the rows of Table 13 and Table 14. If the received bit sequence does not appear in any of the tables, the table row with the shortest Hamming distance to the received is selected. The length of the OpenlinkIQ® embedded frame and Turbo code rate can be derived and is used in frame decoding.

5.4 OpenlinkIQ® embedded frame format

The OpenlinkIQ® embedded format is linked to the CH. Field lengths are determined by the data frame length L and Turbo coding rate $R = \{\frac{1}{2}, \frac{1}{3}\}$ information derived from CH.

5.4.1 Fields

The frame structure and field lengths are shown in Table 6.

Name	Termination – Data Frame	Data Frame	CRC32	Parity – Data Frame
No. of bits	12 bits	L bytes (Turbo coded)	32 bits (Turbo coded)	$(\frac{1}{R} - 1) \cdot (L \cdot 8 + 32)$ bits

Table 6 Structure of the OpenlinkIQ® embedded frame format.

5.4.1.1 Termination – Data Frame (12 bits)

The 12 termination bits from the encoding of the ’Data Frame‘ field and ’CRC32‘ field. The termination bits outputted from the encoder, after the block is encoded, are:

$$x_{10}^0, x_{11}^0, x_{12}^0, p_{10}^0, p_{11}^0, p_{12}^0, x_{10}^1, x_{11}^1, x_{12}^1, p_{10}^1, p_{11}^1, p_{12}^1$$

x represents y^0 and p represents y^1 as defined in section 7.2.

The bits must be transmitted in this specific order, that is, first the three feedback input bits (x_N^0) for terminating encoder #0, then the emitted parity bits (p_N^0) from encoder #0. These are then followed by the similar sequence, but for encoder #1.

5.4.1.2 Data Frame

This field contains the embedded data frame as defined in chapter 6.

5.4.1.3 CRC32

This field contains a CRC32 covering the ‘Length’-field and the ‘Data Frame’-fields.

The polynomial used for the CRC-32 is as follows:

$$x^{32} + x^{31} + x^{30} + x^{29} + x^{28} + x^{26} + x^{23} + x^{21} + x^{19} + x^{18} + x^{15} + x^{14} + x^{13} + x^{12} + x^{11} + x^9 + x^8 + x^4 + x + 1$$

(0x1f4acfb13)

Initial vector: 0x00000000

The result is not complemented.

5.4.1.4 Parity – Data Frame

The data to be encoded is the 'Data Frame‘ and 'CRC32‘ fields. The $(\frac{1}{R}-1)\cdot(L\cdot 8+32)$ parity bits are the output of the Turbo encoder as described in chapter 7.

5.5 Endianness and bit ordering

The physical layer frame structure is a bitwise protocol layer. This means that some fields have lengths not dividable by eight. The total frame is however always an integer number of bytes.

The transmission order of all fields in the physical layer frame structure (Figure 5) is left to right. Each individual field is transmitted with most significant bit first. Endianness and bit ordering of the 'Data Frame' field is described in 6.3.

6 Embedded data frame

6.1 Overview

The embedded data frame conveyed over the OpenlinkIQ® physical frame is specified by the content shown in Figure 6.

The MBAL frame supports unidirectional M-Bus data using the M-Bus adaptation layer defined in EN 13757-8 [5].

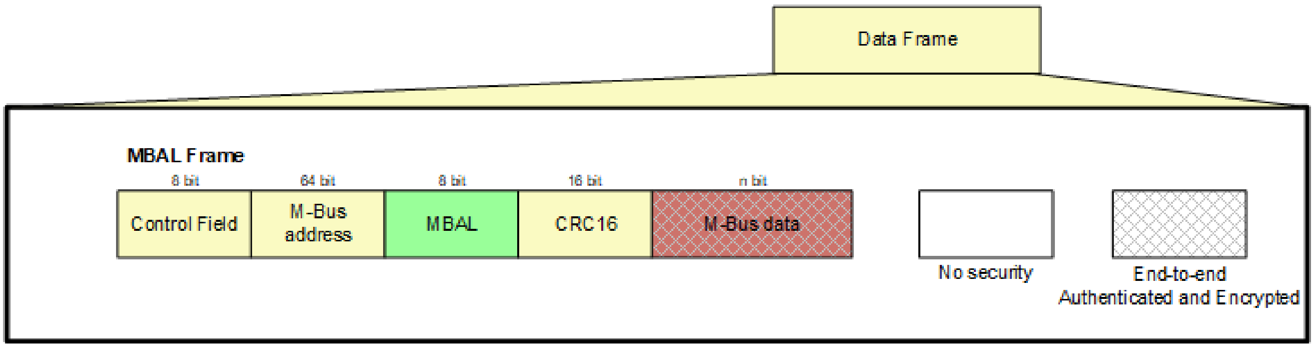

Figure 6 Embedded MBAL frame format.

6.2 MBAL Frame

The MBAL frame is used to embed M-Bus data in the uplink direction (from meter).

Control Field	M-Bus address	MBAL	CRC16	M-Bus data
8 bit	64 bit	8 bit	16 bit	n bit

Table 7 Structure of the MBAL frame.

In Table 7 n represents the number of bits in the 'M-Bus data' field, which is always an integer number of bytes.

6.2.1 Control field (8 bit)

This field is used to address variations in the interpretation or handling of the frame.

BIT0 (LSB): Priority

0_b: The frame contains data with no latency requirements (e.g. scheduled transmissions, 4.3.2.1).

1_b: The frame contains data that is prioritized, i.e. the data shall be transported as fast as possible through the system. Only frames containing alarms and other non-frequent data shall utilize this bit.

BIT1-BIT6: *Reserved* (always 0)

BIT7 (MSB): Control field extension exist in the frame.
(For future protocol extensions)

6.2.2 M-Bus address (64 bit)

The M-Bus address is divided into four fields as shown in Table 8.

Manufacturer ID	Identification number	Version	Device Type
16 bit	32 bit	8 bit	8 bit

Table 8 Sub-field structure of the M-Bus address (from [4]).

6.2.2.1 Manufacturer ID sub-field (16 bit)

The ‘Manufacturer ID’ sub-field contains a unique User/manufacturer ID of the sender. See also section 6.3.

BIT0 (LSB) – BIT14:	The 15 least significant bits of this field shall be formed from 3 uppercase letters as described in EN 13757-7 [4] section 7.5.2.
BIT15:	*Always 0$_b$*

6.2.2.2 Identification number sub-field (32 bit)

The 4 bytes in the ‘Identification number’ sub-field are coded as 8 BCD digits as specified in EN 13757-7 [4] section 7.5.1. See also section 6.3.

6.2.2.3 Version sub-field (8 bit)

The ‘Version’ sub-field is defined according to EN 13757-7 [4] section 7.5.3.

6.2.2.4 Device Type sub-field (8 bit)

The ‘Device Type’ sub-field shall be coded as specified in EN 13757-7 [4] section 7.5.4.

6.2.3 MBAL field (8 bit)

The ‘MBAL’ control field is specified in EN 13757-8 [5] section 7.2.1. The structure of the field is shown in Table 9.

Version	Access	Function code
2 bit	2 bit	4 bit

Table 9 Structure of the MBAL field (see [5]).

BIT0 (LSB) – BIT3: Uplink function codes as defined in EN 13757-8 [5]. Following function codes are valid in OpenlinkIQ® systems:

- 0100$_b$: SND-NR – Send unsolicited application data and expecting no response from HES
- 0110$_b$: SND-IR – Send manual initiated installation data with request to be registered by the HES

BIT4 – BIT5: The accessibility part shall be fixed to

- 00$_b$: No access or temporary no access

BIT6 – BIT7: The MBAL ‘Version’ shall be fixed to

- 00$_b$: Version 1 (initial version of MBAL)

6.2.4 CRC16

This field contains a CRC16 covering the ‘Control field’, ‘M-Bus address’ and the ’MBAL’ fields.

The polynomial is the standard CRC-16 specified in EN 13757-4 [3] section 12.5.7.

$$x^{16} + X^{13} + x^{12} + x^{11} + x^{10} + x^{8} + x^{6} + x^{5} + x^{2} + 1 \qquad \text{(0x13D65)}$$

Initial vector: 0x0000

The result is complemented.

6.2.5 M-Bus data

The M-bus data container is defined in EN 13757-3 [2]. All M-Bus layers above the MBAL are introduced by a CI-field and may be end-to-end encrypted as defined in [4] (see chapter 9).

6.3 Endianness and bit ordering

For the embedded data frame all fields have the size of an integer number of bytes.

The transmission order of the fields or sub-fields (as defined in relevant tables) is left to right. If a field or sub-field consists of multiple bytes, the lowest (rightmost) byte is transmitted first. The 'CRC16' field (6.2.4), shall be transmitted with the highest byte first.

Multi-byte fields within the 'M-Bus data' field shall be transmitted in the order specified in the implemented layers, i.e. according to EN 13757-3 [2] for the application layer and EN 13757-7 [4] for the transport layer.

All individual bytes are transmitted with the most significant bit first.

The transmission order for the first 96 bits MBAL frame, before the 'M-Bus data' field, is shown in Table 10. The example fields show a scheduled data frame transmitted by a heat meter.

Legend	**Control**	**M-Bus address**								**MBAL**	**CRC16**	
Sub-field	-	Manufacturer ID ("KAM")		Identification number (12344321)				Version (25)	Device type (12)	-	-	
Byte order	Byte	Byte (LSB)	Byte (MSB)	Byte (LSB)	Byte	Byte	Byte (MSB)	Byte	Byte	Byte	Byte (MSB)	Byte (LSB)
Example	01_h	$2D_h$	$2C_h$	21_h	43_h	34_h	12_h	19_h	$0C_h$	40_h	$B3_h$	$B5_h$
Example bit order	0000 0001	0010 1101	0010 1100	0010 0001	0100 0011	0011 0100	0001 0010	0001 1001	0000 1100	0100 0000	1011 0011	1011 0101

Table 10 Transmission order for the MBAL frame content.

7 Turbo Coding

Turbo codes are used in many applications such as UMTS, 3G, LTE, etc. The main features of a Turbo code are that:

- Achieves close to Shannon capacity.
- The encoder (running on the meter) is very simple.
- Because the encoder is not a block code, all different packet lengths are supported.

The name Turbo stems from the decoding step, where an iterative decoder can interchange between two decoding steps, achieving much greater performance than considering each decoding step separately. This feedback mechanism has some similarities to that of a turbo in a car, hence the name.

Generally, rate 1/3 coding or higher is applied, where higher rates are obtained by puncturing the parity produced by a rate 1/3 encoder; the same decoder is then used at the receiver side independently of the rate.

In the following, the definitions and notations defined by Liang Li [10] are used.

The applied Turbo encoder consists of three parts shown to the left in Figure 7.

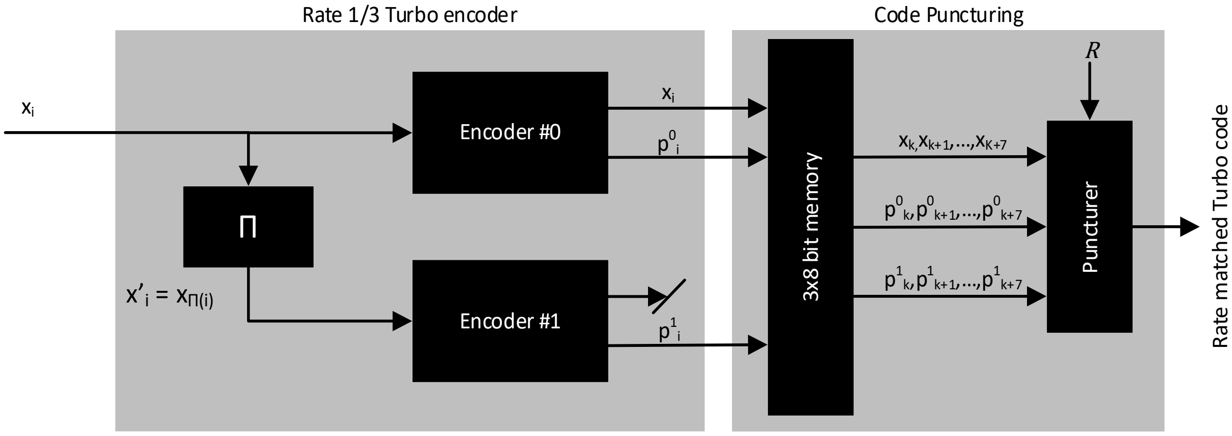

Figure 7 Schematic of the applied Turbo encoder.

The two encoders are recursive and systematic. The recursive element stems from the feedback in the transfer function as described more in detail in section 7.1 and the systematic element ensures that one of the two outputs from each of the encoders is equal to the input. For each input bit x_i, $i = 0,1,\ldots,N-1$, a parity bit p_i^0 is generated from the first encoder (this is the y_i^1 output of the RSC encoder). The original block of N bits is then "randomly" permuted through the interleaver, denoted $\Pi(i)$ (see section 7.3), and this permuted bit sequence defines the input to the second constituent encoder. From this encoder the systematic part is discarded (should not be transmitted), and the total set of bits output from the Turbo encoder is (x_i, p_i^0, p_i^1). This gives a total of three output bits per input bit; hence the rate is 1/3.

To increase the rate (effectively reducing the length of the transmitted message), a code puncturing device can be inserted after the Turbo encoder. The puncturing device holds 8 bits per output channel from the Turbo encoder and decides which of the 24 bits should be transmitted based on the desired rate. For example, if the target rate is $R = 1/2$, then the puncture block will remove bits such that the output from the rate matched Turbo code is:

$$x_k, x_{k+1}, \ldots, x_{k+7}, p_k^0, p_{k+2}^0, p_{k+4}^0, p_{k+6}^0, p_{k+1}^1, p_{k+3}^1, p_{k+5}^1, p_{k+7}^1.$$

This sequence contains 8 data bits per transmitted 16 bits; hence a rate 1/2 matched Turbo code. The punctured bits $p_{k+1}^0, p_{k+3}^0, p_{k+5}^0, p_{k+7}^0, p_k^1, p_{k+2}^1, p_{k+4}^1, p_{k+6}^1$ are not transmitted. The decoder on the receiver knows about the puncturing pattern and can compensate for the not transmitted bits. The puncturing codes for available rates are defined in section 7.4.

Termination is generated for each encoder in the same way as described in 7.2. Note that the feedback input is not the same for the two constituent encoders; hence both the feedback bits and the parity bits must be transmitted for all encoders (in total 12 bits).

7.1 Recursive Systematic Convolution Encoder

The constituent encoder is one of the two key elements in the Turbo encoder. The two encoders are identical recursive systematic code (RSC) encoders, with the only difference that they operate on different input bits (information bits and a permutation hereof). The transfer function for each of the two 8-state constituent encoders is:

$$G(D) = \left[1, \frac{1 + D + D^3}{1 + D^2 + D^3}\right] = \left[1, \frac{15_{octal}}{13_{octal}}\right].$$

The transfer function is illustrated in Figure 8.

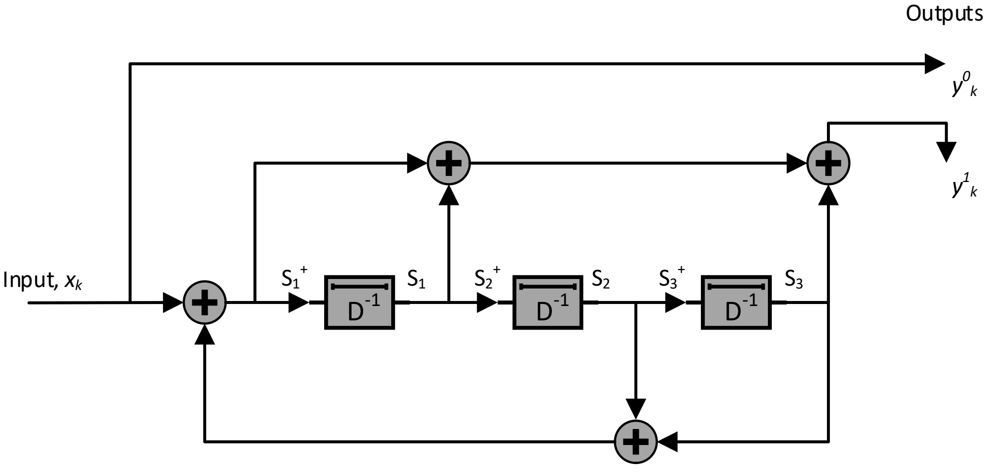

Figure 8 Illustration of the transfer function.

The initial value of the shift registers in the 8-state constituent encoders shall be all zeros when starting to encode the input bits. That is, when $k = 0$, then $s_1 = s_2 = s_3 = 0$. From Figure 8, we clearly see the following relations:

$$\begin{aligned} s_1^+ &= x \oplus s_2 \oplus s_3 \\ s_2^+ &= s_1 \\ s_3^+ &= s_2 \\ y^0 &= x \\ y^1 &= s_1^+ \oplus s_1 \oplus s_3 \end{aligned}$$

7.2 Encoder Termination (Tail-biting)

When the block of N input bits are encoded through a convolutional encoder, then depending on the bits in the encoded block, the internal state in the encoder may be any of 2^M possible states, where $M = K - 1$, K being the constraint length.

At the decoder, it is important to know the termination state of the encoder, otherwise decoding performance is severely degraded. To achieve a known state, a feedback loop is inserted after the N input bits are processed. This feedback ensures that the end state can be forced to the known state 0.

The termination bits, also called tail/tailing bits, are generated by feeding the feedback of the encoder as the input for the next M steps, resulting in the input of the first memory element to be zero M times in a row. This is shown for the encoder in Figure 8 ($K = 4$, $M = 3$) in Figure 9, where $l = N, N + 1, N + 2$.

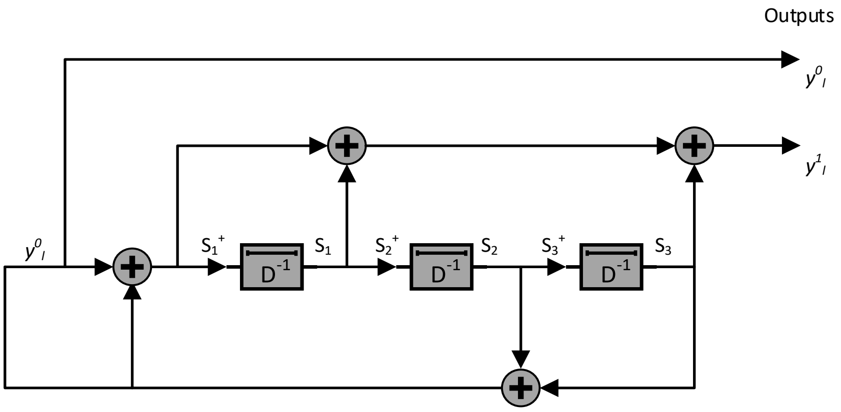

Figure 9 Encoder termination bit generator.

It should be clear from this example, that setting the input to be the feedback bits from the encoder, ensures that s_1^+ is forced to 0 as the output from the left most XOR when the two inputs are the equal. This value is then put through the delay blocks, ultimately ensuring that $s_1 = s_2 = s_3 = 0$. This sequence in total produces the following 6 bits: ${y^0}_N, {y^0}_{N+1}, {y^0}_{N+2}, {y^1}_N, {y^1}_{N+1}, {y^1}_{N+2}$.

In general, it can be said, that M termination steps are to be calculated, generating M termination bits per output (per G-component, including the systematic part).

The sequence is repeated (concatenated) for each of the required RSC encoders used in the Turbo encoder design to form the resultant termination bit sequence.

7.3 Interleaver

The interleaver performs a permutation of the input bits such that the second encoder sees a different block than the original message. Let the input bits to the turbo encoder be $x_0, x_1, \ldots, x_{N-1}$ then the respective output bits from the Turbo interleaver are $x_{\Pi(0)}, x_{\Pi(1)}, \ldots, x_{\Pi(N-1)}$. The relationship between the input index i and the permuted index $\Pi(i)$ is:

$$\Pi(i) = (f_1 i + f_2 i^2) \bmod N$$

The parameters f_1 and f_2 depends on the block length N. The value pairs are listed in Table 11.

N	f_1	f_2	N	f_1	f_2	N	f_1	f_2	N	f_1	f_2	N	f_1	f_2	N	f_1	f_2	N	f_1	f_2	N	f_1	f_2
128	7	16	**368**	11	46	**608**	37	152	**848**	27	106	**1088**	33	68	**1328**	41	166	**1568**	15	112	**1808**	57	226
136	121	102	**376**	23	94	**616**	25	154	**856**	35	214	**1096**	45	274	**1336**	41	334	**1576**	67	394	**1816**	77	454
144	5	12	**384**	35	120	**624**	19	78	**864**	17	48	**1104**	35	138	**1344**	41	84	**1584**	47	132	**1824**	37	228
152	17	114	**392**	25	98	**632**	27	158	**872**	37	218	**1112**	35	278	**1352**	43	338	**1592**	49	398	**1832**	75	458
160	9	20	**400**	7	40	**640**	19	40	**880**	27	110	**1120**	69	140	**1360**	43	170	**1600**	17	80	**1840**	57	230
168	5	42	**408**	25	102	**648**	11	36	**888**	115	222	**1128**	35	282	**1368**	77	114	**1608**	103	402	**1848**	323	462
176	109	132	**416**	25	104	**656**	21	82	**896**	27	56	**1136**	35	142	**1376**	29	172	**1616**	51	202	**1856**	57	232
184	11	46	**424**	27	106	**664**	27	166	**904**	37	226	**1144**	47	286	**1384**	217	346	**1624**	69	406	**1864**	59	466
192	23	48	**432**	7	24	**672**	41	84	**912**	37	228	**1152**	23	48	**1392**	43	174	**1632**	35	204	**1872**	17	156
200	3	20	**440**	27	110	**680**	29	170	**920**	39	230	**1160**	49	290	**1400**	13	70	**1640**	67	410	**1880**	77	470
208	25	52	**448**	13	28	**688**	29	172	**928**	57	116	**1168**	37	146	**1408**	21	44	**1648**	51	206	**1888**	29	118
216	5	18	**456**	173	342	**696**	83	174	**936**	53	78	**1176**	11	84	**1416**	35	354	**1656**	91	138	**1896**	47	474
224	13	28	**464**	15	58	**704**	43	88	**944**	29	118	**1184**	143	296	**1424**	45	178	**1664**	25	52	**1904**	59	238
232	15	58	**472**	57	118	**712**	29	178	**952**	39	238	**1192**	37	298	**1432**	135	358	**1672**	53	418	**1912**	59	478
240	7	30	**480**	29	60	**720**	11	30	**960**	41	240	**1200**	23	120	**1440**	29	60	**1680**	53	210	**1920**	29	60
248	91	186	**488**	31	122	**728**	31	182	**968**	41	242	**1208**	37	302	**1448**	227	362	**1688**	69	422	**1928**	61	482
256	15	32	**496**	15	62	**736**	45	92	**976**	31	122	**1216**	37	76	**1456**	45	182	**1696**	27	106	**1936**	61	242
264	17	66	**504**	13	42	**744**	23	186	**984**	31	246	**1224**	67	102	**1464**	37	366	**1704**	43	426	**1944**	35	108
272	11	68	**512**	15	32	**752**	23	94	**992**	61	124	**1232**	39	154	**1472**	45	368	**1712**	53	214	**1952**	31	122
280	17	70	**520**	21	130	**760**	31	190	**1000**	19	100	**1240**	53	310	**1480**	47	370	**1720**	53	430	**1960**	19	140
288	7	24	**528**	13	132	**768**	23	48	**1008**	13	42	**1248**	77	156	**1488**	47	186	**1728**	31	288	**1968**	61	246
296	39	222	**536**	23	134	**776**	33	194	**1016**	43	254	**1256**	51	314	**1496**	141	374	**1736**	55	434	**1976**	49	494
304	9	38	**544**	9	34	**784**	13	28	**1024**	31	64	**1264**	39	158	**1504**	23	94	**1744**	55	218	**1984**	15	62
312	115	78	**552**	17	138	**792**	17	66	**1032**	97	258	**1272**	119	318	**1512**	29	84	**1752**	107	438	**1992**	127	498
320	19	40	**560**	17	70	**800**	33	200	**1040**	33	130	**1280**	39	80	**1520**	47	190	**1760**	37	220	**2000**	19	100
328	125	246	**568**	23	142	**808**	33	202	**1048**	43	262	**1288**	41	322	**1528**	47	382	**1768**	75	442	**2008**	85	502
336	5	42	**576**	7	24	**816**	25	102	**1056**	43	264	**1296**	23	72	**1536**	47	96	**1776**	55	222	**2016**	41	84
344	21	86	**584**	25	146	**824**	35	206	**1064**	33	266	**1304**	53	326	**1544**	49	386	**1784**	55	446	**2024**	51	506
352	21	88	**592**	25	148	**832**	51	104	**1072**	33	134	**1312**	27	164	**1552**	49	194	**1792**	27	56	**2032**	63	254
360	11	30	**600**	11	60	**840**	79	210	**1080**	19	60	**1320**	41	330	**1560**	49	390	**1800**	17	90	**2040**	43	510

Table 11 Interleaver function f_1 and f_2 values.

7.4 Code Puncturing

Code puncturing is used to match the rate 1/3 output from the Turbo encoder to that of the transmitted packet. For each input byte in the information bits, there are two bytes of parity bits generated; one byte per encoder. The puncturing patterns in Table 12 decide which of the parity bits should be transmitted. A 1 indicates that the bit should be transmitted; a 0 indicates that the bit is punctured and should not be transmitted.

Rate	Puncturing pattern #0	Puncturing pattern #1
1/3	11111111	11111111
1/2	10101010	01010101

Table 12 Puncturing patterns per code rate.

Note that the termination bits generated from the two constituent encoders should not be punctured.

Rate 1/2 example, the following bits are transmitted:

$$x_k, x_{k+1}, \dots, x_{k+7}, p_k^0, p_{k+2}^0, p_{k+4}^0, p_{k+6}^0, p_{k+1}^1, p_{k+3}^1, p_{k+5}^1, p_{k+7}^1.$$

8 Annex: Turbo encoder test vectors

In this annex, test vectors for the Turbo encoding is included for enabling readers to verify implementations.

8.1 Turbo constituent encoder test vectors

Test vectors in this section can be used to validate the convolutional encoder defined in section 7.1.

8 byte test vector

Input: `0x5c068da56183db13`

Parity: `0x6f938994d3f05340`

Termination bits: `0b110010`

20 byte test vector

Input: `0xd7f9b6cc65dd8b2079b396f90a99ed963df69cee`

Parity: `0x90d58866a6ee8d3cecfc9aa7b575e22359f17392`

Termination bits: `0b000000`

50 byte test vector

Input:
`0xe1bbb58dea1906d3e4a0f8cb0fc45e7eb02fec3d168ab576b43c92323a5e0d2e56dfd286fd72ed768e826d520c3e891d5f80`

Parity:
`0xb61641a7bbdb04cbeccefa6356a230b5fe6a2892d169f8714093c10f5c899e454db196f619045be64014a93854296b176d4b`

Termination bits: `0b111001`

100 byte test vector

Input:
`0x19c74935ae4f06ac37aacc3e139b3a86522e561abd96cd94f9c13e75c61ec4d98080fa24c7af572236995ff6c6aefa85124a7fd29ca2cd80aea4ae3cb0c7a3dbc0eaa155153db9e8522bbf87e2d6f7371cf7d290acc83ae9dc6ebccff88e54c948249f84`

Parity:
`0x1019779d20962ac6c25514298bbd2ef6dda086ab1223f0c5d5d6e25d6a15a2c74b656e3ba0b64cf4c390fa3a8f0e1c11415b9a01ca91ac650e707c763545759840bb044f18053bb864d4a8850d06159e38152f9e9ad82eb978f713affa1c3c852ade0310`

Termination bits: `0b010110`

8.2 Full Turbo encoder test vectors

Test vectors in this section can be used to validate the full Turbo encoder functionality defined in chapter 7.

49 byte test vector – code rate 1/2

Input:
`0x01372c341234121b1606ba927a01002005193229bce64d651f1ded42687303b29af6a6805336084a0cc4b4b92371a3cab9`

Parity:
`0x09d35fe3fb1a8228ac6c8d1a7880f5d5cc2aa01dc1059f49e6dfabef60d3ca8e82ed27d78f3f5d499616517ecf777ad7f06cf78f19`

Termination bits: `0b000000101111`

99 byte test vector – code rate 1/2

Input:
`0x002d2c8963007134040458a3900f002c2545420100c9fe780118b7e8317a121840071035cd991de9c53c5dcc3105018782d72d1cdb39c5db1b7c2182057e1935d773afdaaa24f4fa1738e2bd8b13f3fc77a32b68f1d12e7366fec61d69d7e781c28865`

Parity:
`0x0485f9436a6ab176992226a20c65c65bae8f96d5f752b496764ddfa613f9ac7f87af20acade8358c844d76772a94f31f498d6d1553c7ddfbbd8d84747d24875a124350617e08775d239300250a1bda1abdaae6875912d911662df687cc12f9f267635d31f8136c`

Termination bits: `0b111001001011`

60 byte test vector – code rate 1/3

Input:
`0x002d2c02030405060004081c7a22abff2a1001ffeeddcce60d1f01dab0e2832a6518003ee7424ee865dfee2253c0d635eee66977f4204ba93fd3441c`

Parity:
`0x0035d1c85e90bf045cbeeaf37911eda3d58973a3590b3aefc2df5dc5479a49896df4b95bc0082e0131c359a83932747892b339bb4b129179cd2ee9dd5a0b64b5220332c933579b0b75811984920ee65e4892df45180228ce91c678a76a7abce1e7af645b2a03c3e2961f7a4f7d48b171d6c9ce35fd0f2c0db9e208e91f4d4b02`

Termination bits: `0b011101010110`

46 byte test vector – code rate 1/3

Input:
`0x002d2c02030405060004081c7a2a0000000dfd09e30a0301417c033453440d42661b0142fb1a4202446d1e29ab23`

Parity:
`0x0035d1c85e90bf045cbeeaf3791e00000009fc9950c770ca0a21c99c657eecedd6aa2fc36ff7085f0c4c8232b187a550bd002dc59da2ef841654770a2074f56359e930bc8e25b0c9a76b9a6babfec71acbb16e682f432dba1ea3db35fbd333d2fbee5f76`

Termination bits: `0b010110010110`

9 Annex: Precoded MSK modulation

This section will specify how MSK modulation is implemented.

MSK is a special case of frequency shift keying (FSK), where the modulation index $h = 0.5$. This is the smallest possible modulation index for FSK with orthogonal deviation frequencies, hence the name “minimum shift”. The modulation index for binary FSK is calculated using the formula:

$$h = \frac{2\Delta f}{f_s}$$

Where Δf is the deviation frequency and f_s is the signal rate (data rate). We can see from the formula that if $h = 0{,}5$ then $\Delta f = \frac{f_s}{4}$.

To obtain the desired performance in the coherent demodulator, the data bits must be differential precoded before transmission (hence the name, precoded MSK), such that:

$$c_k = d_{k-1} \oplus d_k$$

Where d_k is the kth data bit in the data stream $d_0, d_1, d_2, \ldots,$ and c_k is the kth precoded bit. The above means that the bit-wise XOR'ing of the data bits yields the precoded bits. The bit (d_{-1}) to be used for precoding the first data bit is the last bit of the synchronisation word that precedes the data bits (data bits includes the delimiter bits). The sequence is shown in Figure 10.

Note It is important to notice that it is the previous *data* bit and not the previous *encoded* bit that is applied in the operation.

Note The first data bit, d_0, is the most significant bit of the first transmitted data byte.

Note The last bit in the synchronization word, d_{-1}, is used in the differential precoding of the data. This means that the delimiter bits are precoded to *0b00*.

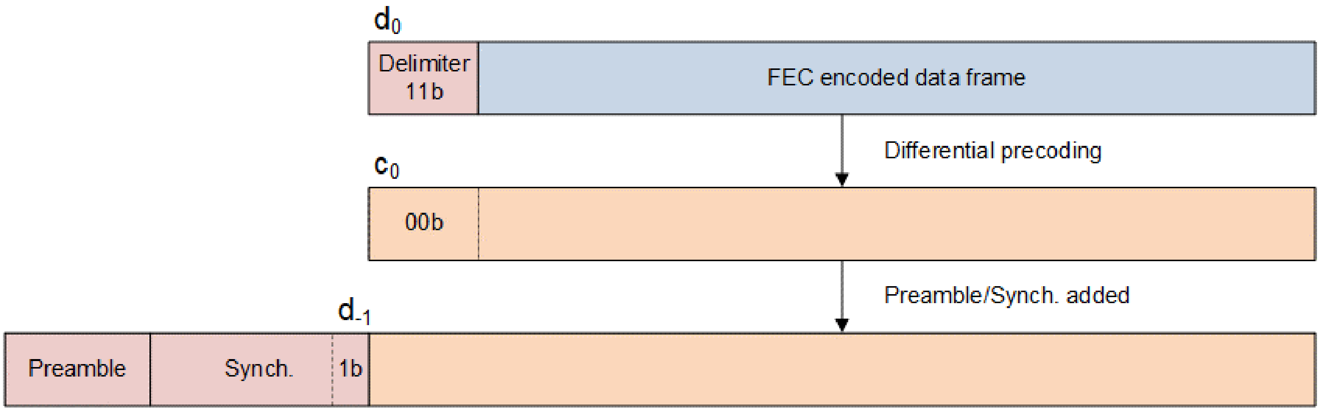

Figure 10 Illustration of the OpenlinkIQ® data to MSK modulation sequence.

10 Annex: Security model

In OpenlinkIQ® the security is implemented in the M-Bus data content. It is therefore fully up to the implementor of the embedded frame to assure encryption and authentication is done using the appropriate M-Bus layers as described in EN 13757-7 [4] when required.

For protecting the privacy of the consumer, all wireless communication containing consumption data shall be encrypted. This is mandatory by national law in most of Europe.

The OMS group [11] recommends that consumption data transmitted over a radio link at least should support security profile A or security profile B (security mode 5 and security mode 7 as defined in EN 13757-7 [4]).

10.1 Security generator inputs from the link layer

When encryption and/or authentication is used the end device ID is often used as an input for e.g. key derivation. For messages which use a short TPL-header the M-Bus address of the end device corresponds to the 'Manufacturer ID', 'Identification Number', 'Version' and 'Device Type' sub-fields of the MBAL frame.

11 Annex: Coded header information

This annex specified the 74 bits defining the ‘Coded Header Information’ field of the Coded Header.

The entry key for each table is the value of the ‘Length’ field. A table is defined for each Turbo coding rate to be used for encoding of the data frame.

An OpenlinkIQ® end device shall implement at least one of the coding rates (Turbo code rate 1/2 and/or 1/3).

11.1 Turbo Code rate 1/2

Length field (decimal)	CH Information
12	00001000110000000111000111001011011101100101101101000000001110011011110000
13	00001001001001010110111110010111011101100101101001001100101011111100100000
14	00001011000010110111001000000000011101011001000001001110010000000101100000
15	00001010111011100110110001011100011101011001000101000010110101100010110000
16	00011110001001010000011110010110011110010110110001001100101100100101100000
17	00011111110000000001100111001010011110010110110101000000001001000010110000
18	00011101111011100000010001011101011110101010011101000010110010111011110000
19	00011100000010110001101000000001011110101010011001001110010111011100100000
20	00011001101100100111111011100100011110010110101101000111011011111011110000
21	00011000010101110110000010111000011110010110101001001011111110011100100000
22	00011010011110010111110100101111011110101010000001001001000101100101100000
23	00011011100111000110001101110011011110101010000101000101100000000010110000
24	00010001000010110000100010111000011101100100001001001110011110010111010000
25	00010000111011100001011011100100011101100100001101000010111011110000000000
26	00010010110000000000101101110011011101011000100101000000000000001001000000
27	00010011001001010001010100101111011101011000100001001100100101101110010000
28	00010110100111000111000111001010011101100100010101000101101001001001000000
29	00010111011110010110111110010110011101100100010001001001001100101110010000
30	00010101010101110111001000000001011101011000111001001011110111010111010000
31	00010100101100100110110001011101011101011000111101000111010010110000000000
32	00111100110000000011101100101110010001011100101101000000000010111110010000
33	00111101001001010010010101110010010001011100101001001100100111011001000000
34	00111111000010110011100011100101010001100000000010011100111001000000000000
35	00111110111011100010011010111001010001100000000101000010111001000111010000
36	00111011010101110100001001011100010001011100110001001011110101100000000000
37	00111010101100100101110000000000010001011100110101000111010000000111010000
38	00111000100111000100000110010111010001100000011101000101101011111110010000
39	00111001011110010101111111001011010001100000011001001001001110011001000000
40	00110011111011100011010000000000010010101110010101000010110000000110010000
41	00110010000010110010101001011100010010101110010001001110010101101011110000
42	00110000001001010011011111001011010010010010111001001100101110010010110000

43	0011000111000000001010011001011101001001001011110100000000101111010110000 0
44	00110100011110010100110101110010010010101110001001001001000111010010110000
45	00110101100111000101001100101110010010101110001101000101100010110101100000
46	00110111101100100100111010111001010010010010100101000111011001001100100000
47	00110110010101110101000011100101010010010010100001001011111100101011110000
48	00100010100111000011101100101111010001011101010101000101100101101100100000
49	00100011011111001001001010111001101000101110101000100100100000000101111 0000
50	00100001010101110011100011100100010001100001111001001011111011110010110000
51	00100000010110010001001101011100001000110000111110100011101111001010110000 0
52	00100101000010110100001001011101010001011101001001001110010010110010110000
53	00100100111011100101110000000001010001011101001101000010110111010101100000
54	00100110110000000100000110010110010001100001100101000000001100101100100000
55	00100111001001010101111111001010010001100001100001001100101001001011110000
56	00101101101100100011010000000000101001010111110110100011101011101111001000 0
57	00101100010101110010101001011101010010101111101001001011110010111001000000
58	00101110011110010011011111001010010010010011000001001001001001000000000000
59	00101111100111000010100110010110010010010011000101000101101100100111010000
60	00101010001001010100110101110011010010101111110001001100100000000000000000
61	00101011110000000101001100101111010010101111110101000000000101100111010000
62	00101001111011100100111010111000010010010011011101000010111110011110010000
63	00101000000010110101000011100100010010010011011001001110011011111001000000
64	01111001000010110000011110010000011110010000101101001110010101101110010000
65	01111000111011100001100111001100011110010000101001000010110000001001000000
66	01111010110000000000010001011011011110101100000001000000001011110000000000
67	01111011001001010001101000000111011110101100000101001100101110010111010000
68	01111110100111000111111011100010011110010000110001000101100010110000000000
69	01111111011110010110000010111110011110010000110101001001000111010111010000
70	01111010101011101111101001010010111101011000111010010111111001011100100000
71	01111100101100100110001101110101011110101100011001000111011001001001000000
72	01110110001001010000100010111110011101100010010101001100100111011100100000
73	01110111110000000001011011100010011101100010010001000000000001011101111000 0
74	01110101111011100000101101110101011101011110111001000010111001000010110000
75	01110100000010110001010100101001011101011110111101001110011100100101100000
76	01110001101100100111000111001100011101100010001001000111010000000001011000 0
77	01110000010101110110111110010000011101100010001101001011110101100101100000
78	01110010011110010111001000000111011101011110100101001001001110011100100000
79	01110011100111000110110001011011011101011110100001000101101011111011110000
80	01100111010101110000011110010001011110010001010101001011110010111100100000
81	01100110101100100001100111001101011110010001010001000111010111011011110000
82	01100100100111000000010001011010011110101101111001000101101100100010110000
83	01100101011110010001101000000110011110101101111101001001001001000101100000
84	01100000110000000111111011100011011110010001001001000000000101100010110000

85	01100001001001010110000010111111011110010001001101001100100000000101100000
86	011000110000101101111110100101000011110101101100101001110011011111100100000
87	011000101110111001100011011101000111101011011000010000101111110011011110000
88	011010000111100100001000101111110111011000111011010010010000000001110010000
89	01101001100111000001011011100011011101100011101001000101100101101001000000
90	01101011101100100000101101110100011101011111000001000111011110010000000000
91	011010100101011100010101001010000111010111110001010010111111011110111010000
92	0110111111110111001110001110011010111011000111100010000101101110100000000000
93	011011100000101101101111110010001011101100011110101001110010010110111010000
94	01101100001001010111001000000110011101011111011101001100101001001110010000
95	011011011100000001101100010110100111010111110110010000000001100101001000000
96	01000101101100100011101100101001010001011011001001000111011100100111010000
97	01000100010101110010010101110101010001011011001101001011111001000000000000
98	01000110011110010011100011100010010001100111100101001001000010111001000000
99	0100011110011100001001101011111001000110011100001000101100111011110010000
100	01000001000100101010000100101101101000101011010101001100101011111001000000
101	01000001111000000001011100000001110100010110110100010000000001110011110010000
102	01000001111011100100000110010000010001100111111001000010110101100111010000
103	01000000000010110101111111001100010001100111111101001110010000000000000000
104	01001010100111000011010000000111010010101001110001000101101110010101100000
105	01001011011110010010101001011011010010101001110101001001001011110010110000
106	010010010101011100110111111001100010010010101011101001011110000001011110000
107	01001000101100100010100110010000010010010101011001000111010101101100100000
108	01001101000010110100110101110101010010101001101101001110011001001011110000
109	01001100111011100101001100101001010010101001101001000010111100101100100000
110	01001110110000000100111010111110010010010101000001000000000111010101100000
111	01001111001001010101000011100010010010010101000101001100100010110010110000
112	01011011111011100011101100101000010001011010110001000010111011110101100000
113	01011010000010110010010101110100010001011010110101001110011110010010110000
114	01011000001001010011100011100011010001100110011101001100100101101011110000
115	0101100111000000000100110101111110100011001100110010000000000000001100100000
116	01011100011110010100001001011010010001011010101101001001001100101011110000
117	01011101100111000101110000000110010001011010101001000101101001001100100000
118	010111111011001001000001100100010100011001100000001000111010010110101100000
119	01011110010101110101111111001101010001100110000101001011110111010010110000
120	010101001100000000110100000000110010010101000001001000000001001000111010000
121	01010101001001010010101001011010010010101000001101001100101100100000000000
122	01010111000010110011011111001101010010010100100101001110010111011001000000
123	010101101110111000101001100100010100100101001000010000101100101111110010000
124	01010011010101110100110101110100010010101000010101001011111110011001000000
125	01010010101100100101001100101000010010101000010001000111011011111110010000
126	01010000100111000100111010111111010010010100111001000101100000000111010000

127	01010001011110010101000011100011010010010100111101001001000101100000000000
128	11110010100111001111010101110010100010111001011101000101100111011011110000
129	11110011011110011110101100101110100010111001011001001001000010111100100000
130	11110001010101111111101101011100110010000101110001001011111001000101100000
131	11110000010110010111010001110010110001000010111010100011101110010001011000
132	11110101000010111000110000000000010001011100100000100111001000000010110000
133	11110100111011101001001001011100100010111001000101000001011010110001011000
134	11110110110000000100011111100101110001000010110110100000001110011011110000
135	11110111001001011001000110010111100010000101101001001100101011111100100000
136	11111101101100101111101001011100100001001011100101000111010101101001000000
137	11111100010101111110010000000000100001001011100001001011110000001110010000
138	11111110011110011111100110010111100001110111001001001001001011110111010000
139	11111111100111001110011111001011100001110111001101000101101110010000000000
140	11111010001001011000001100101110100001001011111001001100100010110111010000
141	11111011110000001001110101110010100001001011111101000000000111010000000000
142	11111001111011101000000011100101100001110111010101000010111100101001000000
143	11111000000010111001111010111001100001110111010001001110011001001110010000
144	11101100110000001111010101110011100010111000100101000000000000001001000000
145	11101101001001011110101100101111000101110000100001001100100101101110010000
146	11101111000010111111011010111000100010000100001001001110011110010111010000
147	11101110111011101110100011100100100010000100001101000010111011110000000000
148	11101011010101111000110000000001100010111000111001001011110111010111010000
149	11101010101100101001001001011101100010111000111101000111010010110000000000
150	11101000100111001000111111001010100010000100010101000101101001001001000000
151	11101001011110011001000110010110100010000100010001001001001100101110010000
152	11100011111011101111101001011101100001001010011101000010110010111011110000
153	11100010000010111110010000000001100001001010011001001110010111011100100000
154	11100000001001011111100110010110100001110110110001001100101100100101100000
155	11100001110000001110011111001010100001110110110101000000001001000010110000
156	11100100011110011000001100101111100001001010000001001001000101100101100000
157	11100101100111001001110101110011100001001010000101000101100000000010110000
158	11100111101100101000000011100100100001110110101101000111011011111011110000
159	11100110010101111001111010111000100001110110101001001011111110011100100000
160	11001110001001011100100111001011101101110010111001001100101110010010110000
161	11001111110000001101011110010111101101110010111101000000001011110101100000
162	11001101111011101100101000000000010110100111001010100001011000001100100000
163	11001100000010111101010001011100101101001110010001001110010101101011110000
164	11001001101100101011000010111001101101110010100101000111011001001100100000
165	11001000010101111010111011100101101101110010100001001011111100101011110000
166	11001010011110011011001101110010101101001110001001001001000111010010110000
167	11001011100111001010110100101110101101001110001101000101100010110101100000
168	11000001000001011110001101110010110111000000000000100111001110010000000000

169	1100000011101110110110001011100110111000000000001010000101110010001110100000
170	1100001011000000110001010010111010111011110010110100000000000101111100100000
171	1100001100100101110110110111001010111011110010100100110010011101100100000 0
172	110001101001110010111111100101111011100000000111010001011010111111100100000
173	1100011101111001101000011100101110111000000000110010010010011100110010000000
174	110001010101011110111100010111001011101111001100010010111101011000000000000
175	110001001011001010100010000000001011101111001101010001110100000000111010000
176	110100000111100111001001110010101011011100110000010010010010010000000000000
177	110100011001110011010111100101101011011100110001010001011011001001110100000
178	110100111011001011001010000000011011010011110110100011101011101111001000 0
179	110100100101011111010100010111011011010011110100100101111001011100100000 0
180	110101111110111010110000101110001011011100110111010000101111100111100100000
181	110101100000101110101110111001001011011100110110010011100110111110010000000
182	110101000010010110110011011100111011010011111100010011001000000000000000000
183	110101011100000010101101001011111011010011111101010000000000101100111010000
184	110111110101011111000110111001001011100000011110010010111110111100101100000
185	110111101011001011011000101110001011100000011111010001110111100101011000000
186	110111001001110011000101001011111011101111010101010001011001011011001000000
187	110111010111100111011011011100111011101111010100010010010000000001011110000
188	110110001100000010111111100101101011100000011001010000000001100101100100000
189	110110010010010110100001110010101011100000011000010011001010010010111100000
190	110110110000101110111100010111011011101111010010010011100100101100101100000
191	110110101110111010100010000000011011101111010011010000101101110101011000000
192	100010111110111011110101011101011000101111101110010000101110010000101100000
193	100010100000101111101011001010011000101111101111010011100111001001011000000
194	100010000001001011111011010111110100010000010010101001100100111011100100000
195	100010011100000011101000111000101000100000100100010000000000101110111100000
196	100011000111100110001100000001111000101111101001010010010011100111001000000
197	100011011001110010010010010110111000101111101000010001011010111110111100000
198	100011111011001010001111110011001000100000100010010001110100000000101100000
199	100011100101011110010001100100001000100000100011010010111101011001011000000
200	100001001100000011111010010110111000010011000000010000000010111100000000000
201	100001010010010111100100000001111000010011000001010011001011100101110100000
202	100001110000101111111001100100001000011100001011010011100101011011100100000
203	100001101110111011100111110011001000011100001010010000101100000010010000000
204	100000110101011110000011001010011000010011000111010010111111001011100100000
205	100000101011001010011101011101011000010011000110010001110110010010010000000
206	100000001001110010000000111000101000011100001100010001011000101100000000000
207	100000010111100110011110101111101000011100001101010010010001110101110100000
208	100101011011001011110101011101001000101111110000010001110111100100000000000
209	100101000101011111101011001010001000101111110001010010111110111101110100000
210	100101100111100111110110101111111000100000111011010010010000000011100100000

211	10010111100111001110100011100011100010000011101001000101100101101001000000
212	10010010001001011000110000000110100010111111011101001100101001001110010000
213	10010011110000000100100100101101010001011111101100100000000110010100100000
214	10010001111011101000111111001101100010000011110001000010110111010000000000
215	10010000000010111001000110010001100010000011110101001110010010110111010000
216	10011010100111001111101001011010100001001101111001000101101100100010110000
217	10011011011110011110010000000110100001001101111101001001001001000101100000
218	10011001010101111111100110010001100001110001010101001011110010111100100000
219	10011000101100101110011111001101100001110001010001000111010111011011110000
220	10011101000010111000001100101000100001001101100101001110011011111100100000
221	10011100111011101001110101110100100001001101100001000010111110011011110000
222	10011110110000001000000011100011100001110001001001000000000101100010110000
223	10011111001001011001111010111111100001110001001101001100100000000101100000
224	10110111010101111100100111001100101101110101011101001011110000001011110000
225	10110110101100101101011110010000101101110101011001000111010101101100100000
226	10110100100111001100101000000111101101001001110001000101101110010101100000
227	10110101011110011101010001011011101101001001110101001001001011110010110000
228	10110000110000000101100001011111010110111010100000100000000011101010101100000
229	10110001001001011010111011100010101101110101000101001100100010110010110000
230	10110011000010111011001101110101101101001001101101001110011001001011110000
231	10110010111011101010110100101001101101001001101001000010111100101100100000
232	10111000011110011100011011100010101110000111100101001001000010111001000000
233	10111001100111001101100010111110101110000111100001000101100111011110010000
234	10111011101100101100010100101001101110111011001001000111011100100111010000
235	10111010010101111110110110111010110111011101100110100101111100100000000000
236	10111111111011101011111110010000101110000111111001000010110101100111010000
237	10111110000010111010000111001100101110000111111101001110010000000000000000
238	10111100001001011011110001011011101110111011010101001100101011111001000000
239	10111101110000000101000100000011110111011101101000100000000111001110010000
240	10101001000010111100100111001101101101110100100101001110010111011001000000
241	10101000111011101101011110010001101101110100100001000010110010111110010000
242	10101010110000001100101000000110101101001000001001000000000100100011101000 0
243	10101011001001011101010001011010101101001000001101001100101100100000000000
244	10101110100111001011000010111111101101110100111001000101100000000111010000
245	10101111011110011010111011100011101101110100111101001001000101100000000000
246	10101101010101111011001101110100101101001000010101001011111110011001000000
247	10101100101100101010110100101000101101001000010001000111011011111110010000
248	10100110001001011100011011100011101110000110011101001100100101101011110000
249	10100111110000001101100010111111101110000110011001000000000000000011001000000
250	10100101111011101100010100101000101110111010110001000010111011110101100000

251	1010010000001011110110110111010010111011101011010100111001111001001011 0000

Table 13 Coded header preceding data frame encoded with Turbo code rate 1/2.

11.2 Turbo Code rate 1/3

Length field (decimal)	CH Information
12	0000100011000001011100011101010101110111101111001000111011100100111010001
13	0000100100100100011011111000100101110111101111110100101111100100000000001
14	0000101100001010011100100001111001110100011101010100100100001011100100000 1
15	000010101110111101101100010000100111010001110100010001011001110111100100 01
16	0001111000100100000000111100010000111100010001001010010111111100110010000 01
17	0001111111000001000110011101010001111000100010000100011101101111111001000 1
18	00011101111011110000010001000011011110110100001001000101100000000111010001
19	0001110000001010000110100001111101111011010000110100100100010110000000000 1
20	00011001101100110111111011111010011110001000111001000000000100100011101000 1
21	0001100001010110011000001010011001111000100011110100110010110010000000000 1
22	0001101001111000011111010011000101111011010001010100111001011101100100000 1
23	0001101110011101011000110110110101111011010001000100001011001011111001000 1
24	0001000100001010000010001010011001110111101001110100100100110010101111000 1
25	0001000011101111000101101111101001110111101001100100010110100100110010000 1
26	0001001011000001000010110110110101110100011011000100011101001011010110000 1
27	0001001100100100000101010011000101110100011011010100101111011101001011000 1
28	0001011010011101011100011101010001110111101000000100001011101111010110000 1
29	0001011101111000011011111000100001110111101000010100111001111001001011000 1
30	0001010101010110011100100001111101110100011010110100110010010110101111000 1
31	0001010010110011011011000100001101110100011010100100000000000000011001000 01
32	0011110011000001001110110011000001000100001011100100011101000000000101100 01
33	0011110100100100001001010110110001000100001011110100101111010110010110000 1
34	0011111100001010001110001111101101000111111001010100100100111001110010000 1
35	0011111011101111001001101010011101000111111001000100010110101111110111100 01
36	0011101101010110010000100100001001000100001010010100110010011101110010000 1
37	0011101010110011010111000001111001000100001010000100000000000101110111100 01
38	0011100010011101010000011000100101000111111000100100001011100100001011000 1
39	0011100101111000010111111101010101000111111000110100111001110010010110000 1
40	0011001111101111001101000001111001001011000000000100010110001011000000000 1
41	0011001000001010001010100100001001001011000000001010010010001110101110100 01
42	0011000000100100001101111101010101001000110010110100101111110010111001000 1
43	0011000111000001001010011000100101001000110010100100011101100100100100000 1
44	0011010001111000010011010110110001001011000000111010011100101011011100100 01
45	0011010110011101010100110011000001001011000000110010000101100000010010000 01
46	0011011110110011010011101010011101001000110011000100000000101111000000000 1

47	001101100101011001010000111110110100100011001101010011001011100101111010001
48	001000101001110100111011001100010100010000110000010000101101110100000000001
49	001000110111110000010010101101101010001000011000101001110010010110111010001
50	00100001010101100011100011111010010001111111101101001100101001001110010001
51	001000000101100110010011010100110010001111111101001000000001100101001000001
52	00100101000010100100001001000011010001000011011101001001000000001110010001
53	00100100111011110101110000011111010001000011011001000101100101101001000001
54	001001101100000010100000110001000010001111111110001000111011110010000000001
55	0010011100100100010111111101010001000111111111010100101111101110111010001
56	00101101101100110011010000011111010010110001111001000000000101100010110001
57	001011000101011000101010010000110100101100011111010011001000000000101100001
58	00101110011110000011011111010100010010001101010101001110011011111100100001
59	00101111100111010010100110001000010010001101010001000010111110011011110001
60	00101010001001000100110101101101010010110001100101001011110010111100100001
61	00101011110000010101001100110001010010110001100001000111010111011011110001
62	00101001111011110100111010100110010010001101001001000101101100100010110001
63	00101000000010100101000011111010010010001101001101001001001001000101100001
64	011110010000101000000011110001110011110001110111001001001000111010010110001
65	01111000111011110001100111010010011110001110111101000101100010110101100001
66	01111010110000010000010001000101011110110010010101000111011001001100100001
67	01111011001001000001101000011001011110110010010001001011111100101011110001
68	011111101001110101111110111111000111100011101001010000101100000001100100001
69	011111110111100001100000101000000011110001110100001001110010101101011110001
70	01111101010101100111110100110111011110110010001001001100101110010010110001
71	011111001011001101100011011010110111101100100011010000000001011110101100001
72	011101100010010000000100010100000011101111100000001001011110101100000000001
73	01110111110000010001011011111100011101111100000101000111010000000111010001
74	011101011110111100001011011010110111010000000101101000101101011111110010001
75	0111010000000101000010101001101110111010000000101001001001001110011001000001
76	011100011011001101110001110100100111011111000111010000000000010111110010001
77	01110000010101100110111110001110011101111100011001001100100111011001000001
78	011100100111100001110010000110010111010000000110001001110011100100000000001
79	011100111001110101101100010001010111010000000110101000010111001000111010001
80	0110011101010110000000111100011110111100011110000010011001000000000000000001
81	01100110101100110001100111010011011110001111000101000000000101100111010001
82	011001001001110100000010001000100011110110011101101000010111110011110010001
83	011001010111100000001101000011000011110110011101001001110011011111001000001
84	01100000110000010111111011111101011110001111011101000111010111011110010001
85	01100001001001000110000010100001011110001111011001001011110010111001000001
86	011000110000101001111101001101100111101100111100010010010010010000000000001
87	01100010111011110110001101101010011110110011110101000101101100100111010001
88	0110100001111000000001000101000010111011111101111001001110010010110010110001

89	01101001100111010001011011111101011101111101111101000010110111010101100001
90	0110101110110011000010110110101001110100000101010100000000110010110010000 1
91	01101010010101100001010100110110011101000001010001001100101001001011110001
92	01101111111011110111000111010011011101111101100101000101100101101100100001
93	01101110000010100110111110001111011101111101100001001001000000001011110001
94	01101100001001000111001000011000011101000001001001001011111011110010110001
95	01101101110000010110110001000100011101000001001101000111011110010101100001
96	01000101101100110011101100110111010001000101011101000000001110011011110001
97	01000100010101100010010101101011010001000101011001001100101011111100100001
98	01000110011110000011100011111000100011110011100010011100100000000101100001
99	01000111100111010010011010100000001000111100111010100001011010110001011000 1
100	01000010001001000100001001000101010001000101000001001011111001000101100001
101	01000011110000010101110000011001010001000101000101000111011100100010110001
102	01000001111011110100000110001110010001111001101101000101100111011011110001
103	01000000000010100101111111010010010001111001101001001001000010111100100001
104	01001010100111010011010000011001010010110111100101000010111100101001000001
105	01001011011110000010101001000101010010110111100001001110011001001110010001
106	01001001010101100011011111010010010010001011001001001100100010110111010001
107	01001000101100110010100110001110010010001011001101000000000111010000000001
108	01001101000010100100110101101011010010110111111100100100100101111011101000 1
109	01001100111011110101001100110111010010110111111110100010110111001000000000 1
110	01001110110000010100111010100000001001000101101010100011101010110100100000 1
111	01001111001001000101000011111000100100010110100010010111100000001110010001
112	01011011111011110011101100110110010001000100100101000101101001001001000001
113	01011010000010100010010101101010010001000100100001001001001100101110010001
114	01011000001001000011100011111101010001111000001001001011110111010111010001
115	01011001110000010010011010100001010001111000001101000111010010110000000001
116	01011100011110000100001001000100010001000100111001001110011110010111010001
117	01011101100111010101110000011000010001000100111101000010111011110000000001
118	01011111101100110100000110001111010001111000010101000000000000000100100000 1
119	01011110010101100101111111010011010001111000010001001100100101101110010001
120	01010100110000010011010000011000010010110110011101000111011011111011110001
121	01010101001001000010101001000100010010110110011001001011111110011100100001
122	01010111000010100011011111010011010010001010110001001001000101100101100001
123	01010110111011110010100110001111010010001010110101000101100000000010110001
124	01010011010101100100110101101010010010110110000001001100101100100101100001
125	01010010101100110101001100110110010010110110000101000000001001000010110001
126	01010000100111010100111010100001010010001010101101000010110010111011110001
127	01010001011110000101000011111101010010001010101001001110010111011100100001
128	11110010100111011111010101101100100010100111001001000010110101100111010001
129	11110011011110001110101100110000100010100111001101001110010000000000000001
130	11110001010101101111011010100111100010011011100101001100101011111001000001

131	1111000010110011111010001111101110001001101110000100000000011100111100 10001
132	111101010000101010001100000011110100010100111010101001001000010111001000001
133	11110100111011111001001001000010100010100111010001000101100111011110010001
134	1111011011000000110001111110101011000100110111111001000111011100100111010001
135	111101110010010010010001100010011000100110111111101001011111001000000000001
136	1111110110110011111110100100001010000101010101110001000000000111010101100001
137	1111110001010110111001000000111101000001010101110101001100100010110010110001
138	11111110011110001111100110001001100001101001011101001110011001001011110001
139	11111111100111011110011111010101100001101001011001000010111100101100100001
140	11111010001001001000001100110000100001010101101101001011110000001011110001
141	11111011110000011001110101101100100001010101101001000111010101101100100001
142	111110011110111110000000111110111000011010010000001000101101110010101100001
143	11111000000010101001111010100111100001101001000101001001001011110010110001
144	11101100110000011111010101101101100010100110110001000111010010110101100001
145	11101101001001001110101100110001100010100110110101001011110111010010110001
146	11101111000010101111011010100110100010011010011101001001001100101011110001
147	11101110111011111110100011111010100010011010011001000101101001001100100001
148	11101011010101101000110000011111100010100110101101001100100101101011110001
149	111010101011001110010010010000111000101001101010010000000000000001100100001
150	111010001001110110001111110101001000100110100000001000010111011110101100001
151	111010010111100010010001100010001000100110100000101001110011110010010110001
152	111000111110111111111010010000111000010101000010010001011000000000111010001
153	1110001000000101011100100000011111100001010100001101001001000101100000000001
154	111000000000100100111100110001000100001101000100101001011111110011001000001
155	111000001110000011110011111010100100001101000100001000111011011111110010001
156	111001000111100010000001100110001100001010100010101001110010111011001000001
157	11100101100111011001110101101101100001010100010001000010110010111110010001
158	111001111011001110000000011111010100001101000111001000000001001000111010001
159	11100110010101101001111010100110100001101000111101001100101100100000000001
160	11001110001001001100100111010101101101101100101101001011111100101110010001
161	110011111100000011101011110001001101101101100101001000111011001001001000001
162	11001101111011111100101000011110101101010000000001000101100010110000000001
163	110011000000010101101010001000010101101010000000101001001000111010111010001
164	1100100110110011101100000101001111011011011001100010000000001011110000000001
165	1100100001010110101010111011111011101101101100110101001100101110010111010001
166	11001010011110001011001101101100101101010000011101001110010101101110010001
167	1100101110011101101010110100110000010110101000001100100000101100000001001000001
168	110000010000010101100011011111011101110011110010101001001001110011100100001
169	110000000111011111110110001010011110111001111001000100010110101111101111 0001
170	11000001011000000111000101001100001011101000101110010001110100000000010110001
171	110000110010010011011011011001011101000101111010010111101011001011 00001
172	11000110100111011011111110001001101110011110001001000010111001000010110001

173	11000111011110001010000111010101101110011110001101001110011100100101100001
174	11000101010101101011110001000010101110100010100101001100100111011100100001
175	11000100101100111010001000011110101110100010100001000000000010111011110001
176	11010000011110001100100111010100101101101101010101001110011011111100100001
177	11010001100111011101011110001000101101101101010001000010111110011011110001
178	11010011101100111100101000011111101101010001111001000000000101100010110001
179	11010010010101101101010001000011101101010001111101001100100000000101100001
180	11010111111011111011000010100110101101101101001001000101101100100010110001
181	11010110000010101010111011111010101101101101001101001001001001000101100001
182	11010100001001001011001101101101101101010001100101001011110010111100100001
183	11010101110000011010110100110001101101010001100001000111010111011011110001
184	11011111010101101100011011111010101110011111101101001100101001001110010001
185	11011110101100111101100010100110101110011111101001000000001100101001000001
186	11011100100111011100010100110001101110100011000001000010110111010000000001
187	11011101011110001101101101101101101110100011000101001110010010110111010001
188	11011000110000011011111110001000101110011111110001000111011110010000000001
189	11011001001001001010000111010100101110011111110101001011111011110111010001
190	11011011000010101011110001000011101110100011011101001001000000001110010001
191	11011010111011111010001000011111101110100011011001000101100101101001000001
192	10001011111011111111010101101011100010100000101101000101101011111110010001
193	10001010000010101110101100110111100010100000101001001001001110011001000001
194	10001000001001001111011010100000100010011100000000100101111010110000000001
195	10001001110000011110100011111100100010011100000101000111010000000111010001
196	10001100011110001000110000011001100010100000011000100111001110010000000001
197	10001101100111011001001001000101100010100000110101000010111001000111010001
198	10001111101100111000111111010010100010011100011101000000000010111110010001
199	10001110010101101001000110001110100010011100011001001100100111011001000001
200	10000100110000011111101001000101100001010010010101000111011001001100100001
201	10000101001001001110010000011001100001010010010001001011111100101011110001
202	10000111000001010111100110001110100001101110111001001001000111010010110001
203	10000110111011111110011111010010100001101110111101000101100010110101100001
204	10000011010101101000001100110111100001010010001001001100101110010010110001
205	10000010101100111001110101101011100001010010001101000000001011110101100001
206	10000000100111011000000011111100100001101110100101000010110000001100100001
207	10000001011110001001111010100000100001101110100001001110010101101011110001
208	10010101101100111111010101101010100010100000101010100000001100101100100001
209	10010100010101101110101100110110100010100001010001001100101001001011110001
210	10010110011110001111011010100001100010011101111001001110010010110010110001
211	10010111100111011110100011111101100010011011111101000010110111010101100001
212	10010010001001001000110000011000100010100001001001001011111011110010110001
213	10010011110000001100100100100010010001010001001101000111011110010101100001
214	10010001111011111000111111010011100010011101100101000101100101101100100001

215	10010000000010101001000110001111100010011101100001001001000000001011110001
216	10011010100111011111101001000100100001010011101101000010111110011110010001
217	10011011011110001110010000001100010000101001110100100111001101111001000001
218	10011001010101101111100110001111100001101111000001001100100000000000000001
219	10011000101100111110011111010011100001101111000101000000000101100111010001
220	10011101000010101000000110011011010000101001111000100100100100000000000001
221	10011100111011111001110101101010100001010011110101000101101100100111010001
222	10011110110000011000000011111101100001101111011101000111010111011110010001
223	10011111001001001001111010100001100001101111011001001011110010111001000001
224	10110111010101101100100111010010101101101011001001001100100010110111010001
225	10110110101100111101011110001110101101101011001101000000000111010000000001
226	10110100100111011100101000011001101101010111100101000010111100101001000001
227	10110101011110001101010001000101101101010111100001001110011001001110010001
228	10110000110000011011000010100000101101101011010101000111010101101001000001
229	10110001001001001010111011111100101101101011010001001011110000001110010001
230	10110011000010101011001101101011101101010111111001001001001011110111010001
231	10110010111011111010110100110111101101010111111101000101101110010000000001
232	10111000011110001100011011111100101110011001110001001110010000000101100001
233	10111001100111011101100010100000101110011001110101000010110101100010110001
234	10111011101100111100010100110111101110100101011101000000001110011011110001
235	10111010010101101101101101101011101110100101011001001100101011111100100001
236	10111111111011111011111110001110101110011001101101000101100111011011110001
237	10111110000010101010000111010010101110011001101001001001000010111100100001
238	10111100001001001011110001000101101110100101000001001011111001000101100001
239	10111101110000011010001000011001101110100101000101000111011100100010110001
240	10101001000010101100100111010011101101101010110001001001000101100101100001
241	10101000111011111101011110001111101101101010110101000101100000000010110001
242	10101010110000011100101000011000101101010110011101000111011011111011110001
243	10101011001001001101010001000100101101010110011001001011111110011100100001
244	10101110100111011011000010100001101101101010101101000010110010111011110001
245	10101111011110001010111011111101101101101010101001001110010111011100100001
246	1010110101010110101011001101101010101101010110000001001100101100100101100001
247	10101100101100111010110100110110101101010110000101000000001001000010110001
248	10100110001001001100011011111101101110011000001001001011110111010111010001
249	10100111110000011101100010100001101110011000001101000111010010110000000001
250	10100101111011111100010100110110101110100100100101000101101001001001000001
251	10100100000010101101101101101010101110100100100001001001001100101110010001

Table 14 Coded header preceding data frame encoded with Turbo code rate 1/3.