BayEOS Frame Protocol Specification

Revision 1.4.0

BayCEER - University of Bayreuth

Inhalt

1 Introduction

BayEOS Frame is a lightweight protocol to send sensor data over low band width communication channels. It can be used to transport sampling data as numeric values over XBee or serial lines from a data producer to a data receiver. The protocol implements basic routing capabilities and allows flexible time stamping.

2 Frame Types

The basic concept of a frame is really simple. A frame is just an array of bytes. The first byte is used to define the frame type and the remainder builds the payload. In general there is no upper size limit of a frame.

Frame Type	Payload

Byte	Name	Length	Value
0	Frame Type	1 Byte	Type identifier
1..N	Payload	N-1 Bytes	According to frame type.

2.1 Data Frame

A data frame can be used to send numeric channel values to a receiver. The frame values are treated as an ordered list of channel values. The index of each value is used as a channel index (starting at 1). Each data frame can only include one type of values.

Frame Type	Value Type	Value 1	Value 2	…

Byte	Name	Length	Value
0	Frame Type	1 Byte	0x1
1	Value Type	1 Byte	0x21: Float32 0x22: Int32 0x23: Int16 0x24: UInt8
2	Value 1, Value 2, .. Value n	n-2, composed of n * value length Bytes	Value length according to value type encoded in LE Order: Float32: 4 Byte Int32:4 Byte Int16:2 Byte Unit8: 1 Byte

2.2 Data Frame with Channel Offset

This type of frame can be used if you would like to split a large frame in several smaller frames. The channel offset defines the original channel position.

Frame Type	Value Type	Channel Offset	Value 1	Value 2	…

Byte	Name	Length	Value
0	Frame Type	1 Byte	0x1
1	Value Type	1 Byte	0x1: Float32 0x2: Int32 0x3:Int16 0x4:UInt8
2	Offset	1 Byte	Channel offset value, Base 0
3	Value 1, Value 2, .. Value n	N-3, composed of n * value length Bytes	Value length according to value type as LE: Float32: 4 Byte Int32: 4 Byte Int16: 2 Byte Unit8: 1 Byte
…			

2.3 Data Frame with Channel Index

This type is used when we would like to transport only a few channel values. The data is sent by pairs of channel index and channel value.

Frame Type	Value Type	Channel Index 1	Value 1	Channel Index 2	Value 2	…	

Byte	Name	Length	Value
0	Frame Type	1 Byte	0x1
1	Value Type	1 Byte	0x41: Float32 0x42: Int32 0x43: Int16 0x44: UInt8
2	Channel Index 1	1 Byte	Channel Index, Base 1
3	Value 1	Value length	Value according to value type.
…			

2.4 Data Frame with Channel Label

Channel values are stored as key value pairs using strings as labels.

Frame Type	Value Type	Label Length	Label	Value 1	Label Length	Label	Value 2	…

Byte	Name	Length	Value
0	Frame Type	1 Byte	0x1
1	Value Type	1 Byte	0x61: Float32 0x62: Int32 0x63: Int16 0x64: UInt8
2	Label Length	1 Byte	Length of next channel label in bytes
3	Channel Label	Label Length	Label as String
3 + Label Length	Value 1	Value length	Value according to value type.
…			

2.5 Command and Response

Command frames are used to send instructions from a controller to the receiver. A typical use case would be to control actors like relays or to trigger a sampling. A command can return a command response frame.

Frame Type	Command Type	Arguments

Byte	Name	Length	Value
0	Frame	1 Byte	Command: 0x2 Response: 0x3
1	Command Type	1 Byte	
2 .. N	Value	N-2 Byte	Argument or response value

2.6 Message

A frame message can be used to transport strings to the receiver. There is a universal message type and an error type.

Frame Type	Payload

Byte	Name	Length	Value
0	Frame Type	1 Byte	Message: 0x4 Error Message: 0x5
1..N	Payload	N-1 Bytes	Message String

2.7 Action and ActionResponse

Frame type which has a similar background as the Command and Response frames. The key is used to separate different actions. The receiver returns an ActionResponse Frame as an answer.

ActionFrame

Frame Type	Action Key	Payload

Byte	Name	Length	Value
0	Frame Type	1 Byte	0x12
1	Action key	1 Byte	
2 .. N	Payload	N-2 Byte	

ActionResponse

Frame which is sent as an answer to an Action frame. A status byte defines whether the action was successful.

Frame Type	Action Key	Status	Payload

Byte	Name	Length	Value
0	Frame Type	1 Byte	0x13
1	Action key	1 Byte	
2	Status	1 Byte	0x0: success 0x1: failed
3 .. N	Payload	N-3 Byte	

3 Wrapped Frames

Several additional Frame types exist to attach additional values to a frame. All types can be nested. The payload is reduced by the header of each wrapped frame.

3.1 Routed Frame

Routed frames are useful in situations where the frame is transported via several hubs. The origin of the original sender (MY_ID and PAN_ID) is wrapped around the original frame.

Frame Type	MY ID	PAN ID	Original Frame

Byte	Name	Length	Value
0	Frame Type	1 Byte	0x6
1..2	MY ID	2 Byte	Int16 [1]
3…4	PAN ID	2 Byte	Int16 [1]
5 ..N	Original Frame	N-5 Bytes	

3.2 Delayed Frame

Delayed frames are used in situations when the frame can't be sent and must be stored to be transmitted later. The delay value is attached to the frame and the receiver can calculate the absolute sampling time by subtract the delay from the reception time.

Frame Type	Delay	Original Frame

Byte	Name	Length	Value
0	Frame Type	1 Byte	0x7
1..4	Delay	4 Byte	Milliseconds as UInt32
5 ..N	Original Frame	N-5 Bytes	

3.3 Routed RSSI Frame

An additional routed frame type including the Remote Signal Strength Indicator (RSSI) value.

Frame Type	MY ID	PAN ID	RSSI	Original Frame

Byte	Name	Length	Value
0	Frame Type	1 Byte	0x8
1..2	MY ID	2 Byte	Int16[1]
3…4	PAN ID	2 Byte	Int16 [1]
5	RSSI	1 Byte	Remote Signal Strength Indicator. Value is negative but coded without sign as Uint8.
6 ..N	Original Frame	N-6 Bytes	

3.4 Timestamp Frame

Timestamp frame are used when the sampling time is created by the frame producer and not by the receiver. A real time clock is necessary to produce the timestamp value.

Frame Type	Timestamp	Original Frame

Byte	Name	Length	Value
0	Frame Type	1 Byte	0x9
1…4	Timestamp	4 Byte	UInt32 as seconds since 2000-01-01 00:00 GMT
5 ..N	Original Frame	N-5 Bytes	

3.5 Binary Frame

This frame type is used to transport arbitrary binary data in a frame. A typical use case would be to transport image data. To overcome a frame size limit, large binary data can be split in several frames by means of a position value.

Frame Type	Position	Payload

Byte	Name	Length	Value
0	Frame Type	1 Byte	0xa
1..4	Position	4 Byte	Position as UInt32
5..N	Payload	N-5 Bytes	

3.6 Origin Frame

This frame type includes the sender address a.k.a. origin as a String.

Frame Type	Origin length	Origin	Original Frame

Byte	Name	Length	Value
0	Frame Type	1 Byte	0xb
1	Origin length	1 Byte	
2..(1+[Origin length])	Origin	Origin Length	Origin as String
(2+[Origin length])..N	Original Frame	N - (2 + [Origin Length])	Original Frame

3.7 Millisecond Timestamp Frame

This frame type stores timestamps with a millisecond resolution.

Frame Type	Timestamp	Original Frame

Byte	Name	Length	Value
0	Frame Type	1 Byte	0xc
1…8	Timestamp	8 Byte	Milliseconds since January 1, 1970, 00:00:00 GMT as a 64Bit long value.
9 ..N	Original Frame	N-9 Bytes	

3.8 Routed Origin Frame

A routed frame with sender information encoded as a string.

Frame Type	Origin Length	Origin	Original Frame

Byte	Name	Length	Value
0	Frame Type	1 Byte	0xd
1	Origin Length	1 Byte	
2	Origin	Origin Length	Origin as String
3	Original Frame	n - (2 + [Origin Length])	Original Frame

3.9 Checksum Frame

A frame that guarantees frame data consistence by means of a 2 Byte Checksum.

Frame Type	Original Frame	Checksum

Byte	Name	Length	Value
0	Frame Type	1 Byte	0xf
1	Original Frame	n -3 Byte	Original Frame
n - 2	Checksum	2 Byte	Checksum as Unit16

Checksum calculation

a) Sum up all bytes of the original frame

b) Subtract the result from 0xffff

3.10 Delayed Second Frame

Another frame type which similar usage scenarios like the Delayed Frame. The delay is expressed as seconds.

Frame Type	Delay	Original Frame

Byte	Name	Length	Value
0	Frame Type	1 Byte	0x10
1..4	Delay	4 Byte	Seconds as UInt32
5 ..N	Original Frame	N-5 Bytes	

4 Sample Frames

The following frames are provided to test an implementation of the protocol:

Frame Type	Array	Value
Data frame with channel offset and Float32 values.	{0x1, 0x1,0x0,0x0,0x80,0xa9,0x41,0x0,0x80, 0xa9,0x41}	value[1]: 21.1875F value[2]:21.1875F
Multiple routed data frame with channel offset and Unit8 value.	{0x6, 0xff,0x00,0xff,0x00,0x6,0xff,0x00,0xff,0 x00,0x1 , 0x4,0,256}	value[1]: 256
Delayed (1 ms) routed frame with channel offset and Uint8 value.	{0x7, 0x01,0x00,0x00,0x00, 0x6,0xff,0x00,0xff,0x00,0x1, 0x4,0,256}	Delay : 1 ms Value[1]: 256

5 References

[1] XBee 802.15.4 (also known as Series 1 hardware) – The initial point-to-point (PTP), point-to-multipoint (PTM) radio running the IEEE 802.15.4 protocol

6 History

Version	Date	Description	Author
1.0.0	25.01.2013	Initial release	Oliver Archner
1.1.0	16.03.2015	Added millisecond frame type	Oliver Archner
1.2.0	08.06.2015	Added binary and origin frame type	Oliver Archner
1.3.0	28.10.2016	Added checksum, routed origin and labeled frame	Oliver Archner
1.4.0	05.09.2024	Added action, delayed second and RF24 frame type	Oliver Archner