Improved PHR coding of the MR-O-QPSK PHY

Michael Schmidt- ATMEL

July 6, 2010
Title: Improved PHR coding of the MR-O-QPSK PHY
Date Submitted: July 6, 2010
Source: Michael Schmidt - Atmel (email: michael.schmidt@atmel.com)
Re: Task Group 15.4g LB51 comment resolution
Abstract: Comment resolutions related to PHR coding of the MR-O-QPSK PHY
Notice: This document has been prepared to assist the IEEE P802.15. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.
Release: The contributor acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15.
Motivation

This document describes the proposed resolution of LB51 on some comments related to MR-O-QPSK PHY regarding the PHY header (PHR).

The following comments are addressed:

- SFD length: CID # 879
- PHR error detection: CID # 890, 891, 912
- PHR FEC: CID # 1470

The following comments are partially addressed:

- Pilot sequences: CID # 1475
Coding of PPDU scheme (780/915 MHz band)\(^1\)

<table>
<thead>
<tr>
<th>Parity</th>
<th>RateMode</th>
<th>Reserved</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 bits</td>
<td>2 bits</td>
<td>1 bit</td>
<td>11 bits</td>
</tr>
</tbody>
</table>

**Preamble** | **SFD** | **PHR** | **PSDU**

15.625 kbit/s: (no FEC) + (bit-diff enc) + (64,1)–DSSS

31.25 kbit/s: (rate 1/2 FEC) + (bit-diff enc) + (16,1)–DSSS

500 kbit/s: (rate 1/2 FEC)–only

\(^1\) as specified for the MR-O-QPSK PHY in draft P802.15.4g/d1, March 2010
Bits of Preamble, SFD and PHR field:
  - no FEC
  - bit-differential encoding (BDE)
  - \((N*4,1)\)-DSSS

Bits of the PSDU for the lowest data rate are:
  - FEC: rate 1/2 convolutional coding, \(K = 7\)
  - bit-differential encoding (BDE)
  - \((N,1)\)-DSSS

(SHR,PHR) and PSDU are approx. balanced with regard to the BER.

So why is there an issue?
Purpose of bit-differential encoding (BDE)

The diagram illustrates the process of bit-differential encoding (BDE) and its components. The flow starts with the (info-)bit经过FEC-ENC, followed by (code-)bit and then $z^{-1}$, leading to $(N,1)$-DSSS and O-QPSK. The output is then fed to the non-coherent demodulation process, which includes de-spread, de-mod, and Re{$x(k)x^*(k-1)$}. The diagram shows the flow of data through these components, illustrating the overall process of bit-differential encoding.
Purpose of bit-differential encoding (BDE)

- low chip SNR
- exploit non-coherent detection
- no phase control loop required
- Note: at low SNR, a phase control loop may cause considerable noise enhancement.
- non-coherent demodulation after de-spreading in order to improve multi-path robustness
Consider a binary \((32,8,d_{\text{min}} = 13)\) block code

- code rate is 1/4
- optimal\(^3\) minimum distance for a binary \((32,8)\) linear code
- Soft-decision ML-decoding at moderate complexity (256 codewords only)

\(^2\)This code can be obtained from a \((33,8,14)\)-code by puncturing position 33; see T. Helleseth & Ø. Ytrehus, How to find a \([33,8,14]\) code, Report in Informatics (preliminary version), Dept. of Informatics, Univ. of Bergen, Norway, Nov. 1989.

\(^3\)see Markus Grassel http://www.codetables.de/
Simulation, assuming perfect chip synchronization and phase

influence of bit-differential encoding (BDE)

- no-FEC + BDE + (64,1)-DSSS (non-coherent)
- no-FEC + no-BDE + (64,1)-DSSS (coherent)
- (32,8,13)-FEC + BDE + (16,1)-DSSS (non-coherent)
- (32,8,13)-FEC + no-BDE + (16,1)-DSSS (coherent)

$4^{\text{relevant for coherent detection}}$
There seems to be little performance difference between:

- no-FEC + BDE + (4*N,1)-DSSS

and

- (32,8,13)-FEC + BDE + (N,1)-DSSS

All coding gain is eaten up by enhanced noise due to reduced spreading length.

But there is a gain!
For real world applications, there is always a frequency offset

\[ \Omega \text{ [ rad/} T_{\text{chip}}\text{]} \]

which cannot be perfectly estimated.

A residual frequency offset \( r = \Omega - \hat{\Omega} \) causes phase drift

\[ \tilde{x}(k) = \exp(jrk)x(k) \]

Depending on \( r \), de-spreading will result in loss of processing gain, when de-correlating against the known chip sequence.

Clearly, a long DSSS sequence is prone to a residual error \( r \).
Influence of residual frequency offset $r = \Omega - \hat{\Omega}$

no-FEC + BDE + (64,1)-DSSS

![Graph showing the BER vs. 10 log10(SNR_chip) for different values of $r$.]
Influence of residual frequency offset $r = \Omega - \hat{\Omega}$

$(32,8,13)$-FEC + BDE + $(16,1)$-DSSS

BER

$10 \log_{10}(\text{SNR}_{\text{chip}}) \text{ [dB]}$

- $r = 0.000$
- $r = 0.005$
- $r = 0.010$
- $r = 0.015$
What about the SFD?

- FEC cannot be conveniently applied to the SFD.
- Will this make PHR FEC useless?
Difference between SFD and PHR field

- During PHR detection, a single info-bit error (or more) usually causes a packet error.
- During SFD detection, a single info-bit error (or more) must not lead to a packet error.
Let $A_D$ be the event of a correctly received word of length $M$

$$A_D : \| w - \hat{w} \| \leq D$$

where

- $\| . \|$ is the Hamming distance
- $w \in GF(2^M)$ is transmitted word
- $\hat{w} \in GF(2^M)$ is the detected word
- $D$ is the maximum number of allowed bit errors.

The complementary event $\bar{A}_D$ is called a word error.
Word Error Rate (WER) for $D = 0$

$M = 16 \quad D = 0; \text{ no-FEC } + \text{ BDE } + (64,1)-\text{DSSS}$
Word Error Rate (WER) for $D = 1$

$M = 16 \quad D = 1; \text{ no-FEC + BDE + (64,1)-DSSS}$

![Graph showing word error rate (WER) vs. 10 log_{10}(SNR_{chip}) [dB] for different values of $r$.]
Word Error Rate (WER) for $D = 2$

$M = 16$  $D = 2$; no-FEC + BDE + (64,1)-DSSS

- $r = 0.000$
- $r = 0.005$
- $r = 0.010$
- $r = 0.015$

$10 \log_{10}(\text{SNR}_{\text{chip}})$ [dB] vs word error rate (WER)
Word Error Rate (WER) for $D = 3$

$M = 16$, $D = 3$; no-FEC + BDE + (64,1)-DSSS

![Graph showing word error rate (WER) vs. 10 \log_{10}(\text{SNR}_{\text{chip}}) [\text{dB}]](image)
SFD candidates

- $M = 16$ bit rather than $M = 8$ bit SFD
- In order to allow bit errors during SFD search while preserving a low false alarm rate, the Hamming distance to zero-bit preamble sequence needs be optimized:

$$w^{opt} = \arg\max_{w \in GF(2^M)} \left\{ \min_{k \in 1, \ldots, M} \|w - (0, \ldots, 0, w_0, \ldots, w_{M-k-1})\| \right\}$$

For an SFD pair, two such SFD words need to be found with good Hamming distances to each other.
SFD candidates

- single SFD

\[(w_0, w_1, ..., w_{15})^{opt} = (1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0)\]

During search, distance to preamble is \(\geq 9\).

- good SFD pair

\[(w_0, w_1, ..., w_{15})^{opt} = (1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0)\]

\[(w_0, w_1, ..., w_{15})^{opt2} = (1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0)\]

During search, distance to preamble is \(\geq 8\) and \(\|w^{opt} - w^{opt2}\| = 9\).

\(^5\)SpreadingMode \(\in\{\text{DSSS, MDSSS}\}\) cannot be indicated by the SFD value.
Candidates for PHR coding and spreading (780/915 MHz band)

<table>
<thead>
<tr>
<th>Scheme</th>
<th># bits</th>
<th>FEC</th>
<th>Intrl.</th>
<th>CRC-8</th>
<th>DSSS</th>
<th># chips</th>
</tr>
</thead>
<tbody>
<tr>
<td>no-FEC-16-bit</td>
<td>16</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>(64,1)</td>
</tr>
<tr>
<td>no-FEC-24-bit</td>
<td>24</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>(64,1)</td>
</tr>
<tr>
<td>BC-16-bit</td>
<td>16</td>
<td>(32,8, (d_{min} = 13))$^7$</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>(16,1)</td>
</tr>
<tr>
<td>BC-24-bit</td>
<td>24</td>
<td>(32,8, (d_{min} = 13))</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>(16,1)</td>
</tr>
<tr>
<td>CC-32-bit</td>
<td>32</td>
<td>1/2 CC$^8$ ( K = 7 )</td>
<td>yes(8x8)</td>
<td>yes</td>
<td>no</td>
<td>(16,1)</td>
</tr>
<tr>
<td>CC-24-bit</td>
<td>24</td>
<td>1/2 CC ( K = 7 )</td>
<td>yes(10x6)</td>
<td>yes</td>
<td>no</td>
<td>(16,1)</td>
</tr>
<tr>
<td>BCH-16-bit</td>
<td>16</td>
<td>BCH$^9$ (63,16, ( t = 11 ))</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>(16,1)</td>
</tr>
</tbody>
</table>

$^6$as specified in draft P802.15.4g/d1, March 2010
$^7$applying ML-decoding with soft decisions on code bits
$^8$applying ML-Viterbi decoding with soft decisions on code bits
$^9$applying usual bounded minimum distance decoding with hard decisions on code bits
BC-16/-24-bit: Rate 1/4 (32,8,13) Block Coding

Ctrl Field (5 bits) | Length Field (11 bits) | HCS Field (8 bits) | PSDU
 oct-1 (8 bits) | oct-2 (8 bits) | oct-3 (8 bits)

(32,8,13)–ENC

code bits (32) | code bits (32) | code bits (32)

BDE + (16,1)–DSSS

dec–spread + non–coh. demodulation

(32,8,13)–ENC

Ctrl Field info available (early setup of PSDU baseband processor)

(32,8,13)–DEC

Length Field info available
**CC-32-bit: Rate 1/2 Convolutional Coding with $K = 7$**

<table>
<thead>
<tr>
<th>Ctrl Field (5 bits)</th>
<th>Length Field (11 bits)</th>
<th>HCS Field (8 bits)</th>
<th>0,0,0,0,0,0,0,0,0,0,0,0</th>
<th>PSDU</th>
</tr>
</thead>
</table>

rate 1/2 Convolutional Coding ($K = 7$)

Interleaving (8x8) of 64 code-bits

BDE + (16,1)–DSSS

de–spread + non–coh. demodulation + de–interleaving

---

**TX**

**RX**

Ctrl Field info available
(late setup of PSDU baseband processor)

Length Field info available
BCH-16-bit: BCH \((63,16, t = 11)\)

- Ctrl Field (5 bits)
- Length Field (11 bits)
- PSDU

BCH (63,16, \(t = 11\)) Encoding

BDE + (16,1)–DSSS

de–spread + non–coh. demodulation

RX

PHR–BCH–DEC

Ctrl Field info available
Length Field info available
(late setup of PSDU baseband processor)

TX
AWGN performance of PHR coding schemes
Burst Errors: $N_c$ consecutive chip values blanked out

Burst Performance of PHR coding schemes

- SFD: 16-bit $D = 2$
- PHR: no-FEC-16-bit
- PHR: BC-24-bit
- PHR: CC-32-bit
- PHR: CC-32-bit (no intl.)
- PHR: BCH-16-bit

Word Error Rate (WER) vs. $N_c$: burst length in # of chip samples ($\rightarrow$ better)
Issues related to BC-24-bit:

- Inserting a CRC-8 based HCS causes more overhead compared to CC-32 (i.e. 1536 chip samples versus 1024 chip samples).
Issues related to CC-32-bit and BCH-16-bit:

- In contrast to the (32,8,13) code (processing data octet by octet), it is not possible to access reliable information of the Control field before all chip samples belonging to the PHR field are received.
- Consequently, chip samples belonging to the PSDU part need to be buffered while decoding the PHR field.
- The buffer size depends on the processing delay of the PHR FEC decoder.
CC-24-bit: Rate 1/2 Convolutional Coding with $K = 7$

Ctrl Field (5 bits) | Length Field (11 bits) | HCS Field (8 bits) | PSDU

Ctrl Field (5 bits) | Length Field (11 bits) | HCS Field (8 bits) | 0,0,0,0,0,0

rate 1/2 Convolutional Coding ($K = 7$)

Interleaving (6x10) of 60 code-bits

BDE + (16,1)−DSSS

append PSDU chip sequence

PHR chip sequence (16 x 60 = 960 chips)

insert pilots into PSDU chip sequence

PHR chip sequence (16 x 60 = 960 chips)
Pilot symbols

PHR decoding (non-coherent)

initial phase estimation

channel update

phase control-loop (coherent only)

PSDU decoding (coherent or no-coherent)

channel update
Pilot symbols

- **first pilot:**
  - breathing space for PHR-FEC decoding
  - simplifies initial phase estimation
  - simplifies initial channel estimation

- **following pilots:**
  - supports channel tracking for long packets when combined with equalization (500 kbit/s PSDU data rate)
  - supports timing point tracking for long packets (all PSDU data rates)

\(^{10}\) used for coherent detection of \((N,4)\)-DSSS or \((N,8)\)-MDSSS during PSDU
Candidates for pilot symbols (780 / 915 MHz band)

<table>
<thead>
<tr>
<th>pilot length # chips</th>
<th>pilot duration [us]</th>
<th>inter pilot spacing # chips</th>
</tr>
</thead>
<tbody>
<tr>
<td>32x4 = 128</td>
<td>128</td>
<td>2048</td>
</tr>
<tr>
<td>16x4 = 64</td>
<td>64</td>
<td>1024</td>
</tr>
</tbody>
</table>

- Assuming a 16-MHz processing clock, this implies 1024 and 2048 processing cycles, respectively.
- This keeps FEC decoder complexity at a moderate level.
- Pilot sequence shall support simple correlation based channel estimation.
- Reasonable inter pilot spacing with regard to expected coherence time while keeping data rate loss due to overhead small (approx. 6%)
PHR Error Detection

- reduces sync on frames with incorrectly decoded Ctrl or Length field information
- may cause overhead
- let $E$ be the event a detected PHR field error
- let $\overline{E} \& \overline{A_0}$ be the event a non-detected PHR field error and the PHR Field was incorrectly received
- let FAR (False Alarm Rate) denote the rate of occurrence of the event $\overline{E} \& \overline{A_0}$
- clearly, $\text{FAR} = \text{WER}$ if no Error detection capability can be exploited
Candidates for PHR Error Detection

- single parity check bit PHR(0) over PHR(1:15)
- two parity check bits\(^{11}\) PHR(0:1) over PHR(2:8) and PHR (9:15), respectively
- CRC-8 based HCS\(^{12}\) with generator polynomial \(X^8 + X^2 + X + 1\)
- implicitly for BCH code (syndrome check)

\(^{11}\) as specified for the MR-O-QPSK PHY in draft P802.15.4g/d1, March 2010
\(^{12}\) similar to the OFDM PHY in draft P802.15.4g/d1, March 2010
Performance of Error Detection: no FEC-16/24-bit

Two parity check bits: poor
CRC-8: good
Performance of Error Detection: BC-16/24-bit

false alarm rate (FAR)

$10 \log_{10}(\text{SNR}_{\text{chip}}) [\text{dB}]$

Two parity check bits: poor
CRC-8: good
Performance of Error Detection: CC-24-bit

Two parity check bits: poor
CRC-8: good
Performance of Error Detection: BCH-16-bit

Two parity check bits: poor
Syndromes: nearly perfect\(^\text{13}\)

\(^{13}\)The BCH-(63,16,t = 11) code has a generator polynomial of degree 47.
<table>
<thead>
<tr>
<th>Scheme</th>
<th>advantage</th>
<th>disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>No-FEC-16-bit</td>
<td>▶ simple detection</td>
<td>▶ detection prone to frequency estimation and burst errors</td>
</tr>
<tr>
<td></td>
<td>▶ early access to Ctrl field (bit-by-bit)</td>
<td>▶ poor error detection (two parity bits)</td>
</tr>
<tr>
<td>No-FEC-24-bit</td>
<td>▶ simple detection</td>
<td>▶ detection prone to frequency estimation and burst errors</td>
</tr>
<tr>
<td></td>
<td>▶ early access to Ctrl field (bit-by-bit)</td>
<td>▶ error detection: CRC-8 causes more overhead</td>
</tr>
<tr>
<td></td>
<td>▶ good error detection (CRC-8)</td>
<td></td>
</tr>
<tr>
<td>Scheme</td>
<td>advantage</td>
<td>disadvantage</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>BC-16-bit</td>
<td>▶ robust detection</td>
<td>▶ poor error detection (two parity bits)</td>
</tr>
<tr>
<td></td>
<td>▶ early access to Ctrl field (octet by octet)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▶ simple decoder</td>
<td></td>
</tr>
<tr>
<td>BC-24-bit</td>
<td>▶ robust detection</td>
<td>▶ error detection: CRC-8 causes more overhead</td>
</tr>
<tr>
<td></td>
<td>▶ good error detection (CRC-8)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▶ early access to Ctrl field (octet-by-octet)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▶ simple decoder</td>
<td></td>
</tr>
<tr>
<td>Scheme</td>
<td>advantage</td>
<td>disadvantage</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>CC-24/32-bit</td>
<td>▶ very robust detection</td>
<td>▶ late access to Ctrl field</td>
</tr>
<tr>
<td></td>
<td>▶ complex hardware of PSDU FEC decoder can be reused</td>
<td>▶ PSDU chip buffering required (unless pilots are prepended)</td>
</tr>
<tr>
<td></td>
<td>▶ good error detection (CRC-8)</td>
<td></td>
</tr>
<tr>
<td>BCH-16-bit</td>
<td>▶ detection: sufficiently robust</td>
<td>▶ late access to Ctrl field</td>
</tr>
<tr>
<td></td>
<td>▶ error detection: nearly perfect (syndromes)</td>
<td>▶ additional complex hardware for FEC decoder required</td>
</tr>
<tr>
<td></td>
<td>▶ syndrome computation can already be performed during receive</td>
<td>▶ PSDU chip buffering required (unless pilots are prepended)</td>
</tr>
</tbody>
</table>
Recommendation

- **CC-24-bit (terminated convolutional coding)**
  - proven approach (similar to IEEE 802.11a/g, TG4g OFDM PHY)
  - 3 octets (24 info bits), while hiding 6 bit termination from the PPDU view
  - termination supports fast traceback
  - hardware for PSDU FEC can be reused

- **CRC-8 based HCS**
  - sufficiently robust
  - can be nicely integrated into CC-24-bit

- extend SFD to 16 bit: reduce preamble by one octet in order to sustain overall SHR length
Recommendation A: SpreadingMode distinguished by SFD

- SpreadingMode = DSSS
  - PHR: CC–24–bit
  - RateMode: 2 bits
  - Reserved: 3 bits
  - Length: 11 bits
  - HCS: 8 bits

- SpreadingMode = MDSSS
  - PHR: scheme to be discussed
  - RateMode: 2 bits
  - Reserved: 3 bits
  - Length: 11 bits
  - (HCS): 8 bits
Recommendation B: SpreadingMode indicated by PHR bit

PHR: CC–24–bit

<table>
<thead>
<tr>
<th>Preamble 56 bits</th>
<th>single SFD 16 bits</th>
<th>Mode 3 bits</th>
<th>Reserved 2 bits</th>
<th>Length 11 bits</th>
<th>HCS 8 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpreadingMode 1 bit</td>
<td>RateMode 2 bits</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

- PHR processing based on pure spreading without FEC is prone to frequency estimation errors and burst errors. **PHR FEC is highly recommended!**

- In order to benefit from PHR FEC, the uncoded SFD field should be extended 16 bits. This allows up to 3 SFD bit errors while keeping the false alarm rate low.

- Periodical insertion of pilots to the PSDU part is very useful for phase estimation, tracking and equalization. Decoding complexity of PHR FEC can be relaxed when prepending a first pilot sequences at the beginning of the PSDU part.

- PHR error detection based on 1 or 2 parity check bits is insufficient, especially in conjunction with FEC.