**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |
| --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) |
| Title | **Kookmin D3 Text Revision** |
| Date Submitted | [July 2017] |
| Source | Trang Nguyen, and Yeong Min Jang (Kookmin University) |
| Re: | **Revise from the original document: 17-0019-02** |
| Abstract | D3 PHY 4, 5, 6 specifications are revised. |
| Purpose | D3 comments and resolutions |
| 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. |

# PHY IV specifications

## **13.3 S2-PSK**

### 13.3.1 Reference Architecture

A reference architecture to implement S2-PSK is shown in Figure ABC.



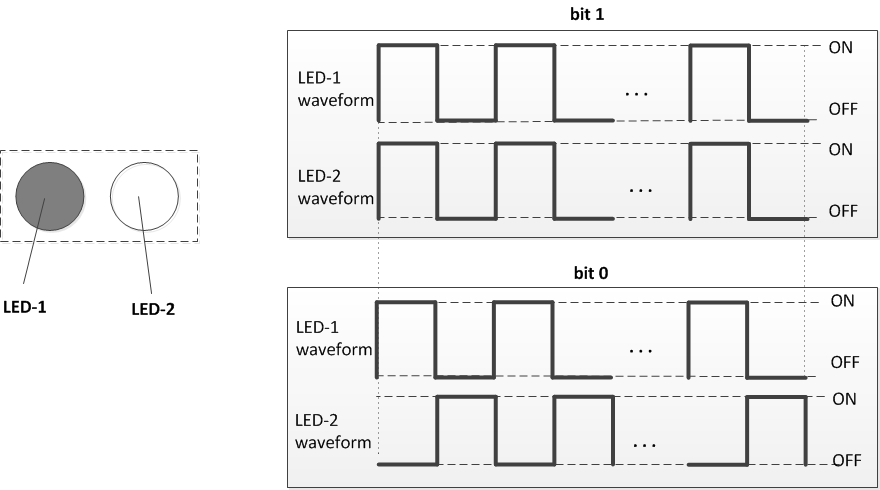
**Figure ABC. S2-PSK block diagram**

### 13.3.1 S2-PSK Bit-to-symbol Mapping

The signal to each LED shall have square waveform which ON and OFF amplitudes are configurable for dimming. A single bit is mapped to a pair of two waveforms to drive a pair of LEDs that have the same phase or inverse phases if the input bit is one or zero respectively (see Figure A1).

The optical clock rate equivalent to bit rate is configured via the PHY PIB attribute *phyOccOpticalClockRate*, and being chosen to be no greater than a half of the camera frame rate (e.g. 10 Hz) to ensure the Nyquist sampling rate.

The configuration of modulation rate is performed over the PHY PIB attribute *phyS2pskModulationRate* that ensures being non-flicker. Besides ensuring the non-flicker requirement, there is no restriction to the choice of the modulation rate. However, the modulation rate that is always a multiple of the optical clock rate to fill up the bit interval. The constant value of the modulation rate is recommended at 200Hz. Reserved values of the PHY PIB attribute *phyS2pskModulationRate* is to extend the modulation rate for later use.

****

**Figure A1 –S2-PSK bit-to-symbol mapping**

### 13.3.2 S2-PSK Line Coding

The RLL coder shall be implemented to protect the signal from the error caused by the rotation of camera and the error caused by the time deviation between a pair of light sources on the rolling image. The configuration of RLL coder is implemented over the PHY PIB attribute *phyOccRLLCode*.

Once the RLL coder is applied, the PPDU shall utilize RLL coding at code rate 1/2 as follows.

|  |  |  |
| --- | --- | --- |
| **Duration** | **the first bit time** | **the second bit time** |
| Data bit | 0 | 1 |
| RLL coding | 0 0 | 0 1 |

### 13.3.3 S2-PSK Error Correction

The configuration of error correction for S2-PSK is implemented via the PHY PIB attribute *phyOccFec.*

In addition, a majority bit voting shall be optionally applied if the camera frame rate (e.g. 30fps) is higher than the optical clock rate (e.g. 10Hz), and being considered as a type of temporal error correction.

## ***13.5 Hybrid Spatial-Phase Shift Keying***

### 13.5.1 Reference Architecture

A reference architecture to implement HS-PSK is shown in Figure ABC. Two data streams are modulated individually in which a low rate data stream is modulated by S2-PSK and a high rate data stream is modulated by DS8-PSK. The outputs of S2-PSK modulator control the dimming levels of the outputs of DS8-PSK.



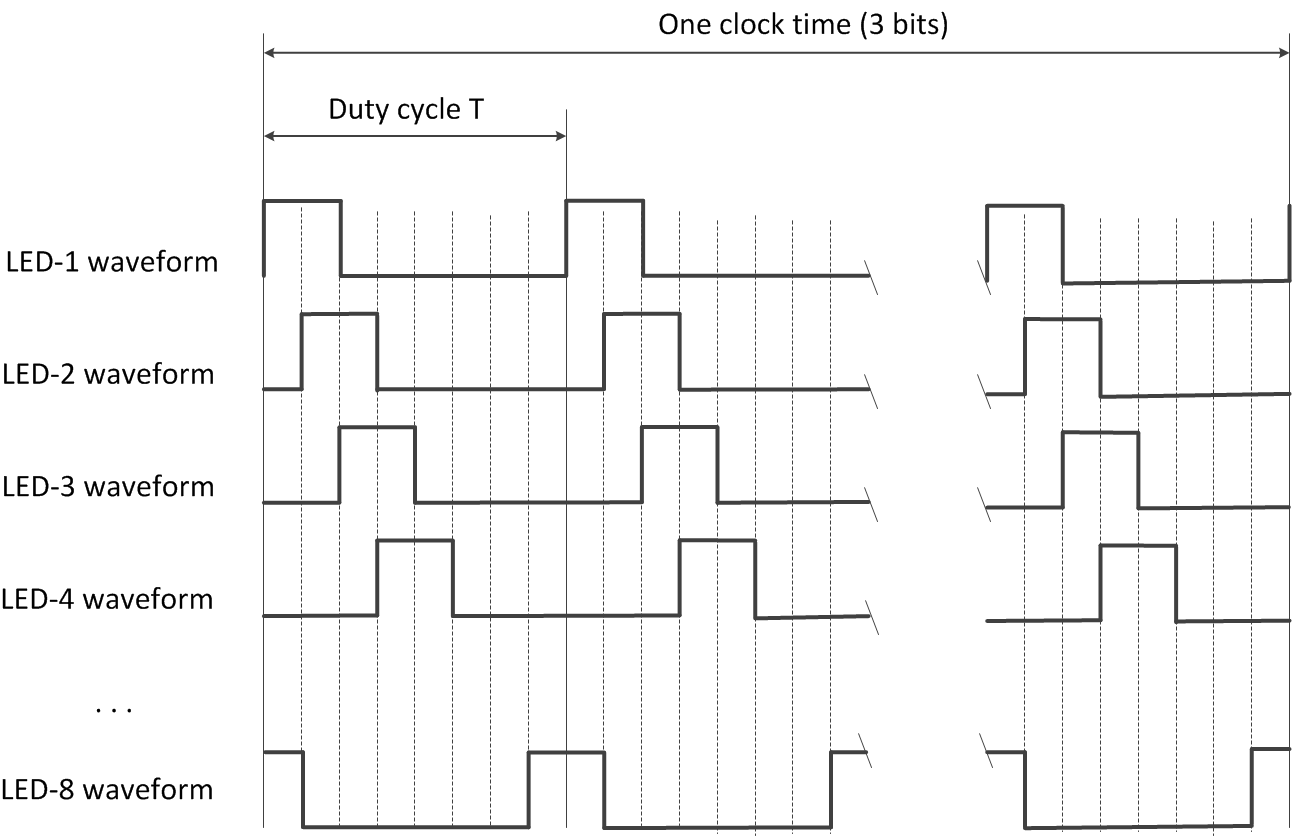
**Figure ABC. HS-PSK block diagram**

### 13.5.2 DS8-PSK modulator

#### 13.5.2.1 DS8-PSK Bits-to-symbol mapping

This subsection describes the DS8-PSK mapping from three-bits into a pair of two-sets of waveforms to drive a pair of light sources. Herein, each light source shall consist of a number of LEDs (e.g. eight LEDs), that is configurable over the PHY PIB attribute *phyHSpskNoLightSources*.

All waveforms to drive LEDs shall have rectangular forms at the same modulation rate (i.e. frequency) but different phases. In a group of waveforms to drive LEDs within a light source, the (i+1)th waveform is delayed 1/8 duty cycle compared to the (i)th waveform as an example of eight LEDs per light source in Fig. A7. The modulation rate of waveform signals that equals 1/(duty cycle) is configured via the PHY PIB attribute *phyHSpskModulationRate*.



**Figure A7. Waveforms to drive a group of eight LEDs within a light source (example of 25% dimming)**

A pair of two light sources, each light source is a group of eight-LEDs, is used to transmit 3 bits per optical clock by controlling the shifting value (called ***S\_Phase\_Shift****)* of the phases of all waveforms between two groups. This is implemented by maintaining the phases of the waveforms of the first group (at 0; T/8; 2T/8; 3T/8; …; 7T/8 respectively) while shifting all the phases of the waveforms of the second group by (i ×T/8) compared to that of the first group, where *i* is an integer depending upon 3-bits input. The mapping from 3 bits to the value of ***S\_Phase\_Shift*** (represented by the value of i) is shown in Table A8.

**Table A8. Mapping table from bits to S\_Phase\_Shift**

|  |  |
| --- | --- |
| 3-bits  **Input** | S\_Phase\_Shift / (T/8)  **Output** |
| 000 | 0 |
| 001 | 1 |
| 010 | 2 |
| 011 | 3 |
| 100 | 4 |
| 101 | 5 |
| 110 | 6 |
| 111 | 7 |

Finally, the optical clock rate to control the frequency at which a block of 3-bit is clocked out shall be configured over the PHY PIB attribute *phyOccOpticalClockRate.*

#### 13.5.2.2 DS8-PSK Forward Error Correction

The configuration of error correction for DS8-PSK shall be implemented over the PHY PIB attribute *phyOccFec.* If DS8-PSK modulation is selected and *phyOccFec = 1,* RS (15, 11) shall be used for DS8-PSK error correction.

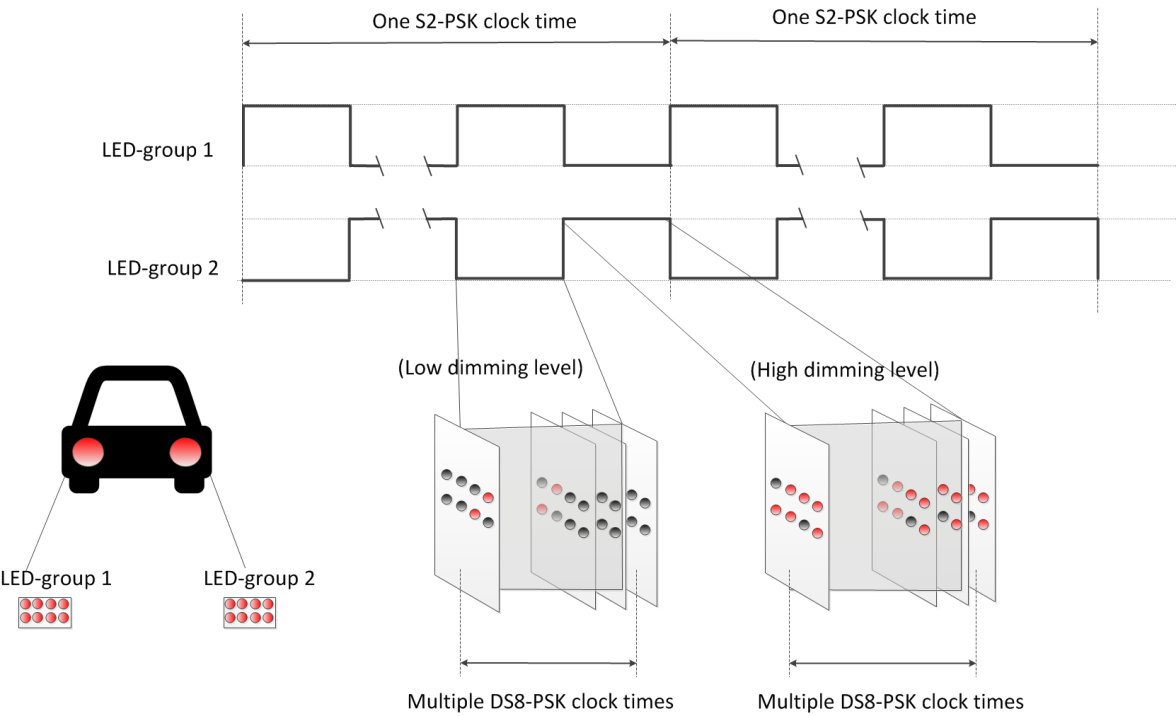
Also, a majority bit voting shall be optionally applied if the camera frame rate is higher than the optical clock rate and being considered as a type of temporal error correction. This is performed by configuring the optical clock rate over the PHY PIB attribute *phyOccOpticalClockRate* that is independent from the configuration of the modulation rate over the PHY PIB attribute *phyHSpskModulationRate*.

Also, the decoding under bad-sampling condition is considered as an error correction with the code rate equals 1. In the DS8-PSK decoder, a bad-sampling that happens when the camera captures the transition time of a single LED among LEDs of the group presents an unclear state (x\_state) among a set of eight states. The demodulation process with x\_state is descried in Annex <J.2>.

### 13.5.3 HS-PSK Waveform

The DS8-PSK (a specific case of DSM-PSK) modulator shall maps 3 bits into a pair of two sets of waveforms driving a pair of light sources at a high optical clock rate (such as 10kHz) as described in Section “13.5.1.1 DS8-PSK bit-to-Symbol mapping”. Meanwhile it periodically changes the dimming level from a selected low dimming level to a selected high dimming level. The change of dimming level during DS8-PSK encoding that generates an AM signal at a low frequency (such as 200Hz or 125Hz) shall be controlled by the S2-PSK modulator. As a result, the S2-PSK clock interval is a multiple of the DS8-PSK clock interval.

Figure A11 illustrates the output waveform of HS-PSK.



**Figure A11 –HS-PSK waveform to modulate vehicular light sources**

### 13.5.4 Forward Error Correction

The configuration of error correction for HS-PSK, including FEC for S2-PSK and FEC for DS8-PSK, shall be implemented over the PHY PIB attribute *phyOccFec.* The suggested configuration is given as follows.

No error correction is needed for S2-PSK. However, a majority bit voting shall be optionally applied if the camera frame rate (e.g. higher than 30fps) is higher than the optical clock rate (e.g. 10Hz), and being considered as a type of temporal error correction.

For DS8-PSK in HS-PSK, RS (15, 11) shall be implemented. Also, a majority bit voting shall be optionally applied if the camera frame rate is higher than the optical clock rate (which is configurable over *phyOccOpticalClockRate*) and being considered as a type of temporal error correction. Also, the decoding under the bad-sampling condition is considered as an error correction with the code rate equals 1 (see the DS8-PSK decoder in Annex J.2).

# PHY V Specifications

## **14.2 CM-FSK Modulation**

### 14.2.1 Reference Architecture

The CM-FSK modulation scheme is applied to a system as shown in **figure B1.** Ab bit(s) shall be inserted into a block of data bits before mapping from bits into frequency (and phase, if used). Finally, the IFFT converts the mapped value into the waveform to drive LED.

A camera that has the sampling rate satisfying the Nyquist sampling is used to receive the modulated light. A demodulation guideline using rolling shutter camera is introduced in Annex J.4.



**Figure B1– Reference Architecture for CM-FSK system**

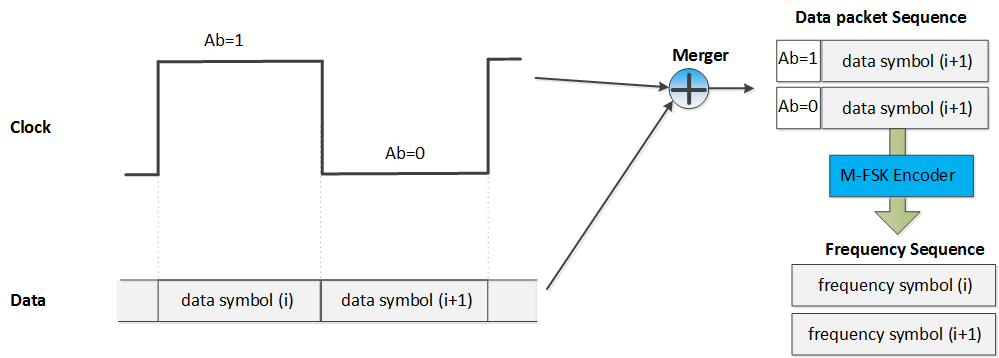
### 14.2.2 Asynchronous bit(s) insertion

Before being feed into the encoder, a number of clock information bits (typically one bit), called Ab, shall be inserted at the beginning of a data symbol to form a packet of bits to support asynchronous communication (see 14.2.3). Also, the number of Ab bits may be increased being greater than one to support the detection of missing symbols during reception time in the receiver side. An example of the detection of missing symbols with two Ab bits is described in section “**14.3.2 Missing packet detection**.”

The configuration of Ab is implemented via the PHY PIB attribute *phyCmfskAb*.

### 14.2.3 CM-FSK Encoder

After inserting Ab, the packet of bits (including data bits and Ab) is feed into the CM-FSK encoder to map from bits into a frequency. Figure B2 illustrates an example of CM-FSK encoding a sequence of data packets into a sequence of frequency symbols.

****

**Figure B2– An example of encoding**

#### **14.2.3.1 Encoder configuration**

The number of frequencies used to map data shall be configured over the PHY PIB attribute *phyCmfskNoFrequency*. 32 or 64 frequencies are suggested to use for selected devices. Also, the number of frequencies is extendable with reserved value of *phyCmfskNoFrequency*.

The frequency separation is fixed during a selected mode, and configurable via the PHY PIB attribute *phyCmfskFrequencySeparation.*

By default, all frequency symbols shall not modulate their phases. However, the number of phases is configurable over the PHY PIB attribute *phyCmfskNoPhase.* If *phyCmfskNoPhase*=1*,* 2-PSK shall be additionally used in conjunction with M-FSK for a specific scenario, utilizing the bandwidth and coverage efficiency.

Finally, the optical clock rate that controls the rate frequency symbols are clocked out is recommended being lower than the selected camera frame rate. The configuration of the optical clock rate shall be implemented via the PHY PIB attribute *phyOccOpticalClockRate*

#### **14.2.3.2 32-FSK bits-to-frequency mapping**

32-FSK is a specific case of CM-FSK encoder. It shall map a packet of bits, including one asynchronous bit and four data bits as shown in Figure B3(a), into a frequency among selected 32 frequencies. The bits-to-symbol mapping table is shown in figure B3(b).

**(b) C32-FSK encoding table**

|  |  |
| --- | --- |
| Packet of bits  **Input** | Frequency **Output** |
| Preamble 1 | f0 |
| 00000 | f1 |
| 00001 | f2 |
| . . . | . . . |
| 11110 | f31 |
| 11111 | f32 |
| Preamble 2 | f33 |

**(a) Packet structure**

|  |  |
| --- | --- |
| b0 | b1 – b4 |
| Ab | Data bits |

**Figure B3–32-FSK encoding**

Beside 32 frequencies (f1 – f32) are selected to encode bits, two additional frequencies are selected as preamble symbols (fSF and f’SF). The relationship between data frequencies and preamble frequencies is as follow:

* Data frequencies: fi = fSF + i.∆f (i=1; 2;…; 32)
* Preambles: f’SF = fSF + 33.∆f

where ∆f is the selected frequency separation value.

The selection of all frequencies shall be configured over the first frequency preamble (fSF) which shall be implemented over the PHY PIB attribute *phyCmfskPreamble1* and the frequency separation which shall be implemented over the PHY PIB attribute *phyCmfskFrequencySeparation.*

#### **14.2.3.3 64-FSK bits-to-frequency mapping**

64-FSK is also a specific case of CM-FSK encoder. It shall map a packet of bits, including one asynchronous bit and five data bits as shown in Figure B4(a), into a frequency among selected 64 frequencies. The bits-to-symbol mapping table is shown in figure B4(b).

**(b) 64-FSK encoding table**

|  |  |
| --- | --- |
| Packet of bits  Input | Frequency Output |
| Preamble 1 | **fo** |
| 00000 | f1 |
| 00001 | f2 |
| … | |
| 11110 | f31 |
| 11111 | f32 |
| Preamble 2 | f33 |
| 010000 | f34 |
| 010001 | f35 |
| … | |
| 111110 | f64 |
| 111111 | f65 |

**(a) Symbol structure**

|  |  |
| --- | --- |
| b0 | b1 – b5 |
| Ab | Data bits |

**Figure B4–64-FSK encoding**

The 64-FSK frequency band is a twice extension of the 32-FSK frequency band. The first 32 data frequencies and two preamble frequencies are the same values as addressed in the 32-FSK modulation. Also, the other 32 frequencies are additionally allocated on the right side of the 32-FSK modulation band to achieve a higher capacity of data per frequency symbol.

#### **14.2.3.4 Hybrid Frequency-Phase Shift Keying**

The hybrid frequency and phase mode is also an extended case of CM-FSK encoder. The M-FSK encoding is achieved by implementing the 32-FSK encoding or the 64-FSK encoding. When *phyCmfskNoPhase =1,* 2-PSK modulation is additionally used in conjunction with M-FSK encoder on the hybrid modulation to tackle the bandwidth efficiency and spatial redundancy.

In an example of using a pair of light sources with *phyCmfskNoPhase*=1, both light sources shall utilize the same bits-to-frequency mapping, thus two light sources carry the same frequency at any time. On the other hand, the relationship of phases of two waveforms shall be modulated by a single Ab bit. If Ab= “0”, two signals at the same phase are output. If Ab= “1”, two inverse phases shall be applied. Figure B5 shows the reference architecture for the hybrid modulator.



**Figure B5– Reference architecture for modulator of using a pair of LEDs**

#### 14.2.**3.1** **Outer FEC**

The payload carried by frequency symbols (sub-packets of data bits and Ab) shall not be coded. However, PHR and PSDU may be protected by RS(15,11) as an outer FEC. The generation of RS(15,11) is described in section “**10.2 Outer forward error correction encoder**”. This option of outer FEC is implemented by configuration of the PHY PIB attribute *phyOccFEC*.

## **14.3 C-OOK**

### 14.3.1 Reference Architecture

A reference architecture to implement C-OOK is shown in Figure ABC.



**Figure ABC. C-OOK block diagram**

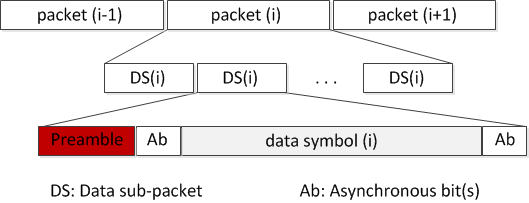
### 14.3.1 C-OOK Encoder

#### 14.3.1.1 Encoder configuration

C-OOK mode is selected if the PHY PIB attribute *phyOCCMscID* = 3.

A sub-packet of data shall be modulated using OOK modulation. The optical clock rate at which OOK symbols are clocked out is configurable over PHY PIB attribute *phyOccOpticalClockRate.* Two options of the optical clock rate, 2.2 kHz and 4.4 kHz, are suggested to use.

The data packet structure is as shown in figure B6. A packet is the multiple times repetition of a data subpacket to avoid missing data in between the gap time of adjacent images. The times of repetition depends on the selected mode, and is configurable over the PHY PIB attributes *phyCookPacketRate* and *phyCookSubPacketRate*. For example, if *phyCookPacketRate =1* specifying 10 packet/sec and *phyCookSubPacketRate =0* specifying 60 sub-packet/sec, every data sub-packet shall be repeated 6 times.



**Figure B6– Data packet structure**

A Data Sub-packet (DS) shall consist of two subfields, a preamble symbol, and a payload section. The configuration of the preamble is performed over the PHY PIB attribute *phyCookPreambleSymbol*. The preamble shall be configured being suitable for the selected RLL coding used in payload subfield. Manchester coded payload shall require a short preamble while 4B6B coded payload shall require a longer preamble. Table B7 shows the suitable preamble for selected RLL code.

T**able B7: Data Sub-packet (DS) format**

|  |  |  |  |
| --- | --- | --- | --- |
| **Preamble** | **DS Payload** | | |
| **Front Ab** | **Data** | **Rear Ab** |
| 011100 | Manchester coding | | |
| 0011111000 | 4B6B coding | | |

#### 14.3.1.2 RLL coding

RLL coding shall be applied in payload subfield to maintain the average brightness at 50%. The configuration of RLL code shall be implemented over the PHY PIB attribute *phyOccRLLCode*. Manchester code and 4B6B code are suggested for C-OOK mode.

#### 14.3.1.3 Ab insertion

The payload of DS (see table B7) shall consist of three subparts: a front Ab, data, and a rear Ab. The front Ab and the rear Ab shall carry the same information which consists of a single asynchronous bit or more. The configuration of the number of asynchronous bits for the front Ab and the rear Ab subparts shall be implemented over the PHY PIB attribute *phyCookAb*.

The use of a single Ab to support Asynchronous Decoder shall be described in section 14.3.2. A pair of Ab bits to support the detection of missing packets shall be described in section 14.3.3.

#### 14.3.1.4 Forward Error Correction

The DS payload, including data bits and Ab bits, may be coded by inner FEC to protect the payload from error. Hamming (8,4) or (15,11) code may be used as an inner FEC.

Additionally, outer FEC may also be used to protect PHR and PSDU. In case of outer FEC is enable, RS(15,11) shall be implemented. The generation of RS(15,11) is described in section “**10.2 Outer forward error correction encoder**”. Also, if outer FEC is applied, the output of FEC shall be feed into a block interleaver with the interleaver height n=15. The implementation description of the block interleaver is described in Section “**10.3 Interleaving and puncturing block”**

Both inner FEC and outer FEC shall be configured via the PHY PIB attribute *phyOccFEC*.

Finally, a receiver may implement repeating the data reception to vote data and correct possible error. This is considered as a temporal error correction.

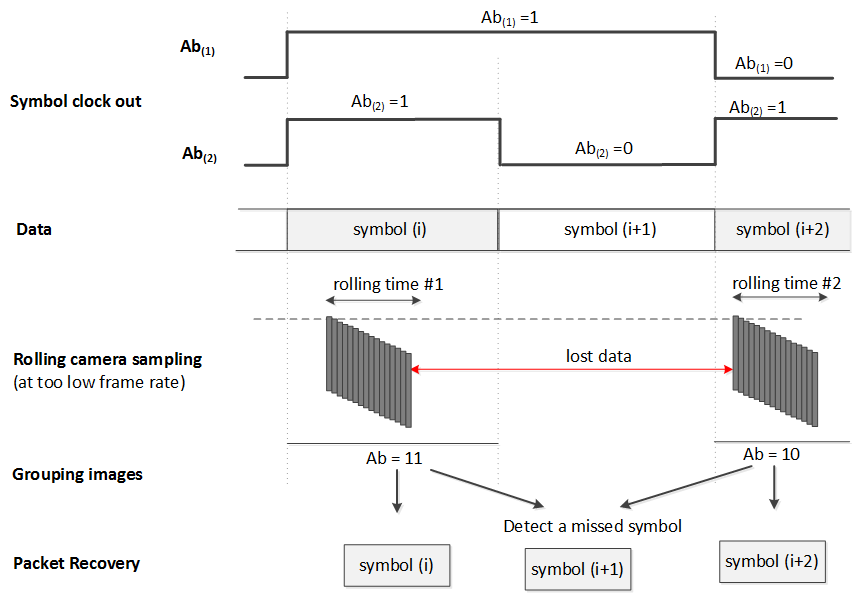
### 14.3.2 Missing packet detection on frame rate drop

The decoding procedure in sub-clause 15.3.2 is performed under the assumption of the receiver frame rate greater than the transmitting packet rate. In some circumstance, the frame rate may drop to less than the packet rate, causing to an entire packet is missed. The detection of the missed packet is proposed herein for a later process.

The core idea comes from the usage of asynchronous bits inserted into the payload of every sub-frame. Two bits (Ab1Ab2) are inserted at the forward and the backward of the body payload as shown in Figure B11. Those two bits together bring the clock information of the sub-packet and being modulated as shown in Figure B12.

|  |  |  |  |
| --- | --- | --- | --- |
| **Preamble** | **DS payload** | | |
| **Ab (front)** | **data bits** | **Ab (rear)** |
|  | 2 bits (Ab1Ab2) | Variable | 2 bits  (Ab1Ab2) |

**Figure B11– Data Sub-Packet Structure**

****

**Figure B12–An example of the detection of a missed symbol by using two Ab inserted**

Ab1 and Ab2 are square signals. Ab1 changes from zero/one into one/zero every time of single data packet, while Ab2 changes every time of two data packets.

The combination of two Ab, Ab1 and Ab2, generates four different values, 00 01 10 and 11. Therefore, the usage of those two Ab enables the detection of 2 missed packets continuously. It means the detection of missed packets is successful for any frame rate drop to no less than 1/3 of the packet rate. For example, a packet rate at 10Hz with 2 Ab allows the frame rate drops to 3.3fps while all the missed packets are detectable.

### 14.3.3 Packet Structure Specification Modes

Tables B13 and B14 below suggest some parameters for C-OOK modes.

**Table B13 – Suggested Parameters for C-OOK modes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Mode 1** | **Mode 2** | **Mode 3** | **Mode 4** |
| Optical clock rate | 2.2 kHz | 2.2 kHz | 4.4 kHz | 4.4 kHz |
| Sub-Packet rate | 100 DS/s | 60 DS/s | 60 DS/s | 60 DS/s |
| RLL code | Manchester | 4B6B | Manchester | 4B6B |
| Uncoded bit rate | 80 bps | 180 bps | 330 bps | 400 bps |

**Table B14– SubPacket Structure on Suggested C-OOK modes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Mode 1** | **Mode 2** | **Mode 3** | **Mode 4** |
| DS clocks | 22B | 37B | 74B | 74B |
| Preamble | 6B | 10B | 6B | 10B |
| DS payload  (Ab, data bits, Ab) | (16B)  8 bits | 18 bits  (27B) | 33 bits  (66B, 2B unused) | 40 bits  (60B, 4B unused) |
| Inner FEC | Hamming (8,4) | Hamming (15,11)  3 bits unused | Hamming (15,11)  3 bits unused | None |
| Bit rate | 40bps | 110bps | 220bps | 400bps |

# PHY VI Specifications

## **15.1 A-QL Mode**

### 15.1.1 Reference Architecture

Figure 221 shows the A-QL system diagram with light sources of three colors (band i, j, k). The valid selection of these three bands is described in table 132. An example of color band combination (110, 010, 000) for band i, j, k respectively is called red, green, and blue.



Figure 221: A-QL system diagram

The data sequence including PHR, PSDU, and Ab bits shall be feed into the channel encoder. This A-QL channel encoder controls the intensity of RGB channels (Pi, Pj, and Pk) based on bits input. However, unlike CSK system in PHY III, the intensity modulation of each color channel in A-QL system is independent from the others, and there is no requirement of color combination such as Pi +Pj +Pk =1. Thus, A-QL modulation operates with flickering, and no effort is made to generate a desired color within the triangle IJK on the xy color coordinates.

**15.1.1.1 2D color code design**

Two-dimensional design of A-QL code for sequential transmission is shown inFigure Cx2.

Four cells (each cell is 2x2) at the four corners of the code called reference cells are to support the receiver in identifying the starting corner of the code. The intensity of these reference cells do not change over time (see Section **15.1.2.1 Encoder configuration**). In addition, a corner has the number of reference cells that is configurable over the PHY PIB *phyAqlNoCells*.

The remaining cells change their intensity to transmit data, called data cells.



**Figure. (a) Design example of 16x16-cell A-QL code, and (b) Allocation of bits onto A-QL code**

### 15.1.2 Channel Encoder

#### **15.1.2.1 Encoder configuration**

A block of coded bits shall be converted into the matrix of intensity values (binary values) to drive the data cells via three color channels, red, green, and blue independently. The optical clock rate at which the intensity of data cells is updated is configurable over PHY PIB attribute *phyOccOpticalClockRate.*

In a MxN-cell Tx, (MxN -16) data cells shall be used to transmit 3x(MxN -16) binaries each time through three color bands if the number of reference cells is *phyAqlNoCells* =16. The forming process of a matrix of binaries to drive all MxN cells is described via an example of 16x16 Tx as follows.

* The intensity values to drive 16-reference cells are constant over time:
* The intensity values of data cells are mapped from bits as described in section **“15.1.2.2 bit-to-intensity mapping”**

#### **15.1.2.3 Asynchronous bits (Ab) insertion**

To support Rx dealing with its frame rate variation, Ab shall be generated and inserted into every block of data bits, and then, error correction is applied and coded bits are mapped into intensity. The number of Ab bits generated for each block of data is configurable via PHY PIB *phyOccNoAb*. The A-QL decoding method with Ab support is described in Annex J.5.

When *phyOccNoAb* =1, a single bit (zero or one) shall be generated for each block of data bit as described in table A1 and then added at the beginning of the block of data.

Table A1**: Generation of single Ab for each block of data bits (***phyOccNoAb* =1)

|  |  |  |
| --- | --- | --- |
| **Optical clock index** | 2i | 2i+1 |
| **Block of data** | block(2i) | block(2i+1) |
| **Ab** | 1 | 0 |

When *phyOccNoAb* = 4, Ab that has 7 binaries (0101010, 1010101, 1111000, or 0000111) shall be generated for each block of data (as described in table A2) according to the optical clock index of the block of data, and then added at the beginning of the block of data.

Table A2: **Generation of Ab for each block of data bits (***phyOccNoAb* = 4)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Optical clock index** | 4i | 4i+1 | 4i+2 | 4i+3 |
| **Block of data** | block(4i) | block(4i+1) | block(4i+2) | block(4i+3) |
| **Ab** | 0101010 | 1010101 | 1111000 | 0000111 |

**15.1.2.3 Error correction**

The PHR, PSDU, and Ab bits are coded to protect from noisy channel as described as follows.

**Table 1: Error correction for A-QL**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Outer code** | **Interleaver (optional)** | **Inner code (optional)** |
| PHR, PSDU, Ab bits | RS(15,7) | n=15 | CC (1/3) |

RS(15,7) shall be applied for A-QL system. The generation of RS(15,7) is described in section **“10.2 Outer forward error correction encoder”.**

In addition, the inner code based on a rate-1/3 mother convolutional code of constraint length seven (K=7) shall be optionally used for each block of data. The generation of inner FEC is described in section **“10.4 Inner forward error correction encoder”.**

The selection of error correction is configurable by the value of PHY PIB attribute *phyOccFec*.

When both RS(15,7) and CC(1/3) are applied, a block interleaver with the interleaver height n=15 shall be implemented between the inner convolutional code and the outer RS code. The description of the block interleaver is described in Section “**10.3 Interleaving and puncturing block”**

### 15.1.3 Bit-to-intensity mapping

The intensity of band i, j, k is modulated independently based on the value of 3 bits input (b0b1b2). Notably, there is no requirement of color combination while modulating the intensity, thus Pi +Pj +Pk ≠1.

The mapping from coded bits to intensity (Pi, Pj, Pk) is given in table A3.

Table A3: Bits to (Pi, Pj, Pk) mapping

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Input** | Intensity Output | | |
| b0b1b2 | Pi | Pj | Pk |
| 000 | 0 | 0 | 0 |
| 001 | 1 | 0 | 0 |
| 010 | 0 | 1 | 0 |
| 011 | 1 | 1 | 0 |
| 100 | 0 | 0 | 1 |
| 101 | 1 | 0 | 1 |
| 110 | 0 | 1 | 1 |
| 111 | 1 | 1 | 1 |

### 15.1.4 Color calibration at the receiver

A channel estimation sequence shall be added as an extended subfield after the PHR subfields to support a receiver dealing with multi-color imbalance or multi-color interference. The block diagram with color calibration is described in Figure B1.



Figure B1. A-QL system with color calibration

Three Walsh codes W(1,4), W(2,4), and W(3,4) are used for band R, G, B respectively for channel estimation before data communication. Each bit of the Walsh code are transmitted twice. Accurate channel estimation can also be obtained as described in Section “12.9 CSK calibration at the receiver”.



Figure: A block with the preamble and Walsh code for color calibration

## **15.6 Hidden A-QL (HA-QL)**

### 15.6.1 Reference Architecture

The reference architecture for HA-QL is specified as shown in figure x7. Data bits are feed into FEC Encoder. The coded sequence is splitted; and then goes to RLL encoder to generate pairs of images according to the data input. The channel encoder will allocate data to fit the screen size.



**Figure x1. Reference Architecture**

The number of cells on the screen to be modulated is configurable by the value of PHY PIB attribute *phyHAqlNoCell*.

### 15.6.2 Channel encoder

#### 15.6.2.1 RLL encoder

The HA-QL encoder shall utilize 1/2-rate line coding (see table x1). From a block of data, the RLL encoder utilizes the 1/2-rate line code to generate a pair of two images, a reference image, and a data image. In within both reference image and data image, the reference cells at the corners of each image shall not carry any data. These reference cells are modulated by the intensity matrix, R2×2 for the reference image and R’2×2 for the data image, to support the determination of the starting corner of Tx (see table x2).

**Table x1. RLL encoding table**

|  |  |  |
| --- | --- | --- |
| **Binary Input** | **RLL Output** | **Cell(ij) intensity description** |
| “0” | 0 0 | the state of the cell does not change between two frames |
| “1” | 0 1 | the state of the cell changes between two frames. |

Finally, the output of RLL encoder and channel encoder is as shown in table x2.

**Table x2– HA-QL RLL and channel encoding**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Data block time (2i) | | Data block time (2i+1) | |
|  | Reference image | Data emdeded image | Reference image | Data emdeded image |
| cells Tx | invisible 1.png | C:\Users\Trang\Desktop\Warsaw\New folder\invisible\invisible 2.png | invisible 1.png | C:\Users\Trang\Desktop\Warsaw\New folder\invisible\invisible 2.png |
| reference-cells |  |  |  |  |
| PSDU | Data block (2i) | | Data block (2i+1) | |
| Ab to insert | bit 1 | | bit 0 | |

where R2×2 is the intensity matrix to modulate four reference-cells for supporting the identification of the starting corner of the code.

To support a receiver dealing with its frame rate variation, Ab shall be inserted into the beginning of every block of data. The number of Ab is configurable via the PHY PIB atrribute *phyHAqlAb*. After that, a block of data along with Ab shall also be protected from error by using inner FEC (see 15.6.2.2 Error correction)

The decoding guideline is described in Annex J-5. A-QL decoding method.

#### 15.6.2.2 Error correction

RS(15,7) shall be implemented as an inner FEC to protect data from error.

|  |  |  |
| --- | --- | --- |
|  | Error correction |  |
| **PHR, PSDU, Ab** | RS(15,7) |  |

The value of PHY PIB attribute *phyOccFec* shall determine the error correction used if any change is applied.

### 15.6.3 Bits-to-Intensity mapping

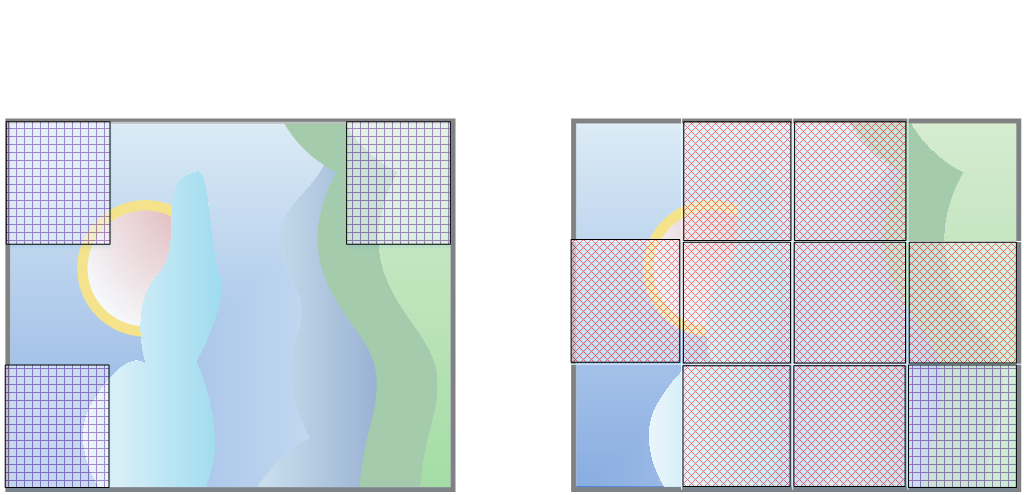
The channel encoder converts a block of data bits into a pair of matrixes, in which the first matrix contains all zeros and the second matrix contains the data bits. The value of each element in a matrix (element # ij) shall be mapped into the intensity (Pij) to control the corresponding data cell (cell ij) of the HA-QL layer that is to be added in front of the original image.

bit 0: Pg(ij) = 0

bit 1: Pg(ij) = 1

where Pg(ij) = 0 and Pg(ij) = 1 indicates that the intensity of data cell (ij) is unmodulated and modulated respectively in compared to that intensity value the original image. The change should less than 0.008 on Cb channel of YCbCr color space to be imperceptible to human eyes. The configuration of the level of intensity modulation is along with the PHY PIB attribute *phyHAqlIntensity*.

Consequently, a block of data is carried by a pair of images including a reference image with unmodulated intensity of all data cells and a data embedded image with the modulated intensity of data cells as illustrated in figure x2.



Reference image (no data)

Data embedded image

**Figure x2. Illustration of HA-QL channel encoder output**

The HA-QL encoder divides the Screen transmitter into nxm cells as shown in figure x2 to modulate the intensity of individual cells.

Four reference cells at corners are modulated by the clock information (2x2 matrix via reference-cells), thus only data bits (nxm – 4) bits are carried by data cells.

The optical clock rate at which the cells on Tx are updated is configurable over PHY PIB attribute *phyOccOpticalClockRate.*

# Annex J:

# PHY 4,5,6 OCC-Waveforms Decoding Guide

### J-2. DS8-PSK Decoding Method

At a sampling time, a camera captures two groups of light sources; and from an image, each group forms a set of eight ON/OFF states. The decoding procedure is illustrated as in Figure x.

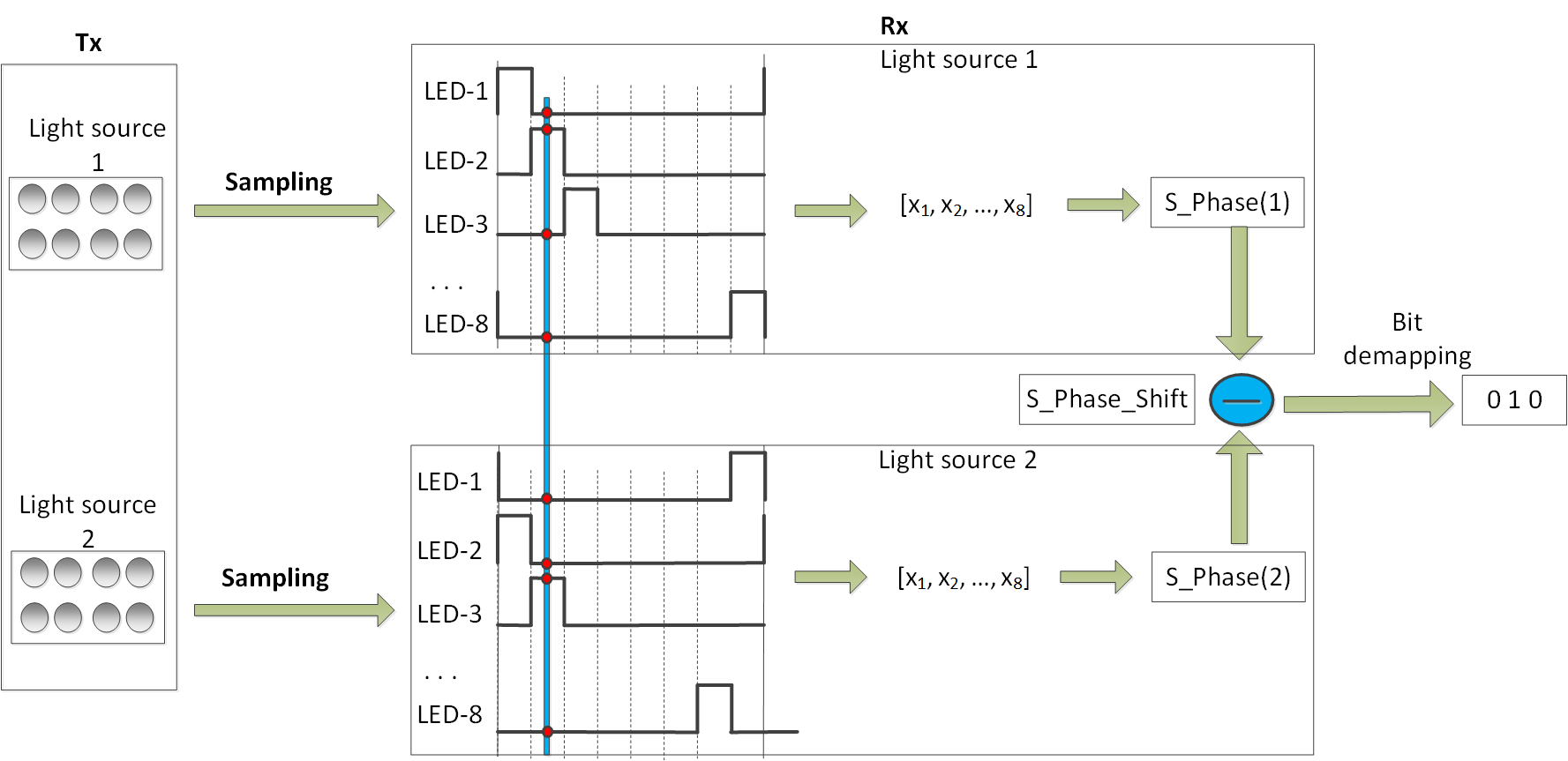


Figure. Decoding procedure

The captured set of a group shall be represented by **S\_Phase** as shown in Table 166.

**Table 166: S\_Phase representing the captured set of states of a light source under dimming**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **8-states Input** | | | | | | | **S\_Phase Output** |
| **Dimming 1/8** | **Dimming 2/8** | **Dimming 3/8** | **Dimming 4/8** | **Dimming 5/8** | **Dimming 6/8** | **Dimming 7/8** |
| 1000 0000 | 1000 0001 | 1000 0011 | 1000 0111 | 1000 1111 | 1001 1111 | 1011 1111 | 1 |
| 0100 0000 | 1100 0000 | 1100 0001 | 1100 0011 | 1100 0111 | 1100 1111 | 1101 1111 | 2 |
| 0010 0000 | 0110 0000 | 1110 0000 | 1110 0001 | 1110 0011 | 1110 0111 | 1110 1111 | 3 |
| 0001 0000 | 0011 0000 | 0111 0000 | 1111 0000 | 1111 0001 | 1111 0011 | 1111 0111 | 4 |
| 0000 1000 | 0001 1000 | 0011 1000 | 0111 1000 | 1111 1000 | 1111 1001 | 1111 1011 | 5 |
| 0000 0100 | 0000 1100 | 0001 1100 | 0011 1100 | 0111 1100 | 1111 1100 | 1111 1101 | 6 |
| 0000 0010 | 0000 0110 | 0000 1110 | 0001 1110 | 0011 1110 | 0111 1110 | 1111 1110 | 7 |
| 0000 0001 | 0000 0011 | 0000 0111 | 0000 1111 | 0001 1111 | 0011 1111 | 0111 1111 | 8 |

Also, a sampling called bad-sampling (when the camera captures the transition time of a single LED among LEDs of the group) shall generate a presence of an unclear state (x\_state) among a set of eight states. The determination of S\_Phase value under the presence of x\_state is as shown in Table 167.

**Table 167: S\_Phase representing the captured set of states of a light source (with x\_state) under dimming**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **8-states Input** | | | | | | | **S\_Phase Output** |
| **Dimming 1/8** | **Dimming 2/8** | **Dimming 3/8** | **Dimming 4/8** | **Dimming 5/8** | **Dimming 6/8** | **Dimming 7/8** |
| xx00 0000 | 1x00 000x | 1x00 00x1 | 1x00 0x11 | 1x00 x111 | 1x0x 1111 | 1xx1 1111 | 1 |
| 0xx0 0000 | x1x0 0000 | 11x0 000x | 11x0 00x1 | 11x0 0x11 | 11x0 x111 | 11xx 1111 | 2 |
| 00xx 0000 | 0x1x 0000 | x11x 0000 | 111x 000x | 111x 00x1 | 111x 0x11 | 111x x111 | 3 |
| 000x x000 | 00x1 x000 | 0x11 x000 | x111 x000 | 1111 x00x | 1111 x0x1 | 1111 xx11 | 4 |
| 0000 xx00 | 000x 1x00 | 00x1 1x00 | 0x11 1x00 | x111 1x00 | 1111 1x0x | 1111 1xx1 | 5 |
| 0000 0xx0 | 0000 x1x0 | 000x 11x0 | 00x1 11x0 | 0x11 11x0 | x111 11x0 | 1111 11xx | 6 |
| 0000 00xx | 0000 0x1x | 0000 x11x | 000x 111x | 00x1 111x | 0x11 111x | x111 111x | 7 |
| x000 000x | x000 00x1 | x000 0x11 | x000 x111 | x00x 1111 | x0x1 1111 | xx11 1111 | 8 |

From an image, two group of light sources shall produce a pair of S\_Phase values (S\_Phase(1) representing the captured state of the group 1 and S\_Phase(2) representing the captured state of the group 2). And then, the value of ***S\_Phase\_Shift*** is calculated as following:

***S\_Phase\_Shift = S\_Phase(1) - S\_Phase(2)***

Finally, the de-mapping from S\_Phase\_Shift to 3 bits shall be done inversely as the mapping table (**Table 136)** showed.

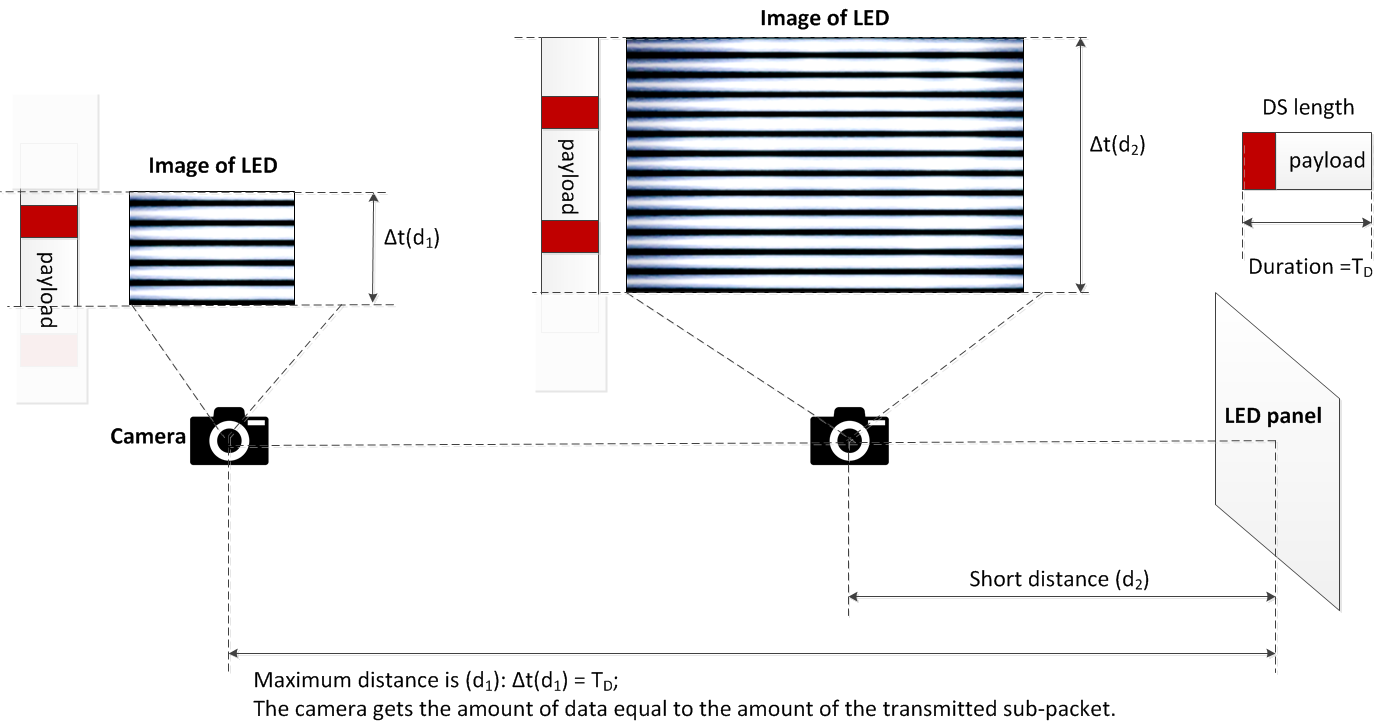
### J-3. HS-PSK Decoding Method

For a dual-camera receiver system, the hybrid signal can be demodulated as below:

* + A low frame rate camera (the frame rate should be greater than the S2-PSK optical clock rate) is to detect the S2-PSK signal. Either a global or a rolling shutter camera can be used.
  + A global shutter and high-speed camera (the frame rate should be greater than the D8M-PSK optical clock rate) is to decode the DS8-PSK signal.

### J-4. C-OOK Decoding Method

To demodulate the entire data sub-packet DS, the distance from a camera to the LED transmitter should be close enough. Figure 258 shows the relationship between the amount of data being captured by the camera and the distance from the camera to the LED transmitter.



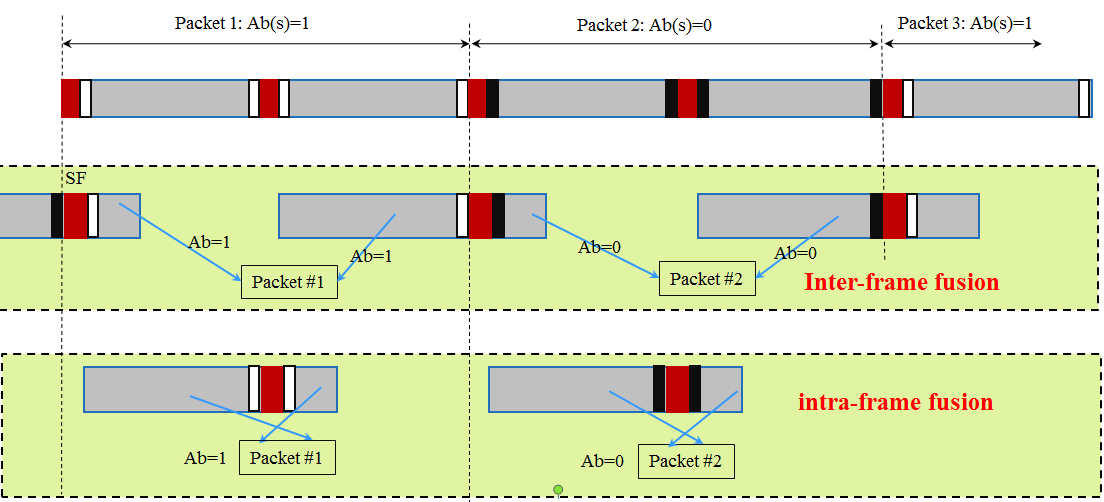
**Figure 258– Decoding scenario**

From the figure 258, the maximum distance achieved is the distance at which the camera gets the amount of data equal to the amount of the sub-packet.

**Decoding case 1: Fuse incomplete parts of a sub-packet into a complete one**

At this distance far, the distance d1 as shown in figure 258, the camera detects the preamble symbol and then demodulates the amount of data enough for a sub-packet; however, the uncertainty whether the forward part and the backward part counted from the position of the preamble belong to a sub-packet or not is problematic. The problem of a small amount of data also happens at a shorter distance when the transmitted subpacket is long.

Asynchronous bits representing the clock information of the packet are used for the asynchronous decoding algorithm in this case.



**Figure 259– Decoding algorithm at a far distance**

Figure 259 illustrates the decoding algorithm to recover a packet of data from the forward part and the backward part of an image when the size of LED is small in the captured image. By observing the values of an asynchronous bit before and an asynchronous bit after the preamble, two statements of fusing those two parts of a subpacket are addressed:

* Case 1- *Inter-frame data fusion*: Fusing two parts of a packet at two different images into a complete packet.

This type of data fusion is applied in case two Ab on an image are different.

* Case 2- *Intra-frame data fusion*: Recovering a complete packet from an image.

This type of data fusion is applied in case two Ab on an image are similar.

**Decoding case 2: Combination of Data Fusion and Majority Voting**

When the camera goes closer to the LED transmitter, the amount of data being captured per image is greater than that of a sub-packet. Therefore, the extra amount of data is used for correcting the possible error by applying a majority vote.

At distance d2 on figure 58, the amount of data equivalent to two sub-packets is captured. The majority voting is used in this case to correct the error throughout the entire sub-packet.

Figure 260 shows an experimental example of decoding under *Intra-frame data fusion.* The extra data after fusion a sub-packet is used for correcting the error by voting.

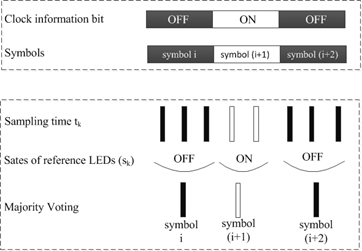
Assume that the camera frame rate may vary but be greater than the packet rate of transmission. Therefore, any extra data after fusion is useful for the error correction by grouping multiple images which belong to a sub-packet to vote. The voting is on the amount of data grouped from all of the forward parts and backward parts of images as well as extra data.



**Figure 260– An example of decoding employing intra-frame fusion along with error correction.**

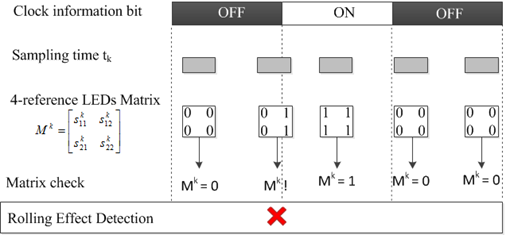
**J-5. A-QL Decoding Method**

A camera that has frame rate greater than the optical clock rate shall be used to receive data. Assume that there is more than one image sampled per block of data. The decoder shall group the adjacent blocks that have the same Ab bits as shown in Figure 261. After that, the majority voting shall be applied for each group of adjacent blocks to down-sample and mitigate error.



**Figure 261 –Asynchronous decoding**

In case a rolling shutter camera captures in between the transition time of two adjacent blocks of data, four Ab bits at four reference cells on the captured image shall not be the same. The image is detected as a rolling affected image as shown in Figure 262, and be discarded.



**Figure 262 – Detection and Removal of a Rolling Effected Image**

Also, to support a rotated camera decoding, the transmission of four rotation-indication bits via reference cells over the blue channel allows a receiver in identifying the starting corner of the code. The starting corner shall have the similar values with its two adjacent reference cells.

# Annex L:

# L-1. Generation of Hamming code

# L-2. Temporal Error Correction and Majority bit voting