**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |
| --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) |
| Title | **Kookmin D1 Text Revision updating to PHY PIBs** |
| Date Submitted | [January 2017] |
| Source | Trang Nguyen, and Yeong Min Jang (Kookmin University) |
| Re: |  |
| Abstract | D1 PHY 4, 5, 6 specifications are revised according to PHY PIB attributes. |
| Purpose | D1 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 A specifications

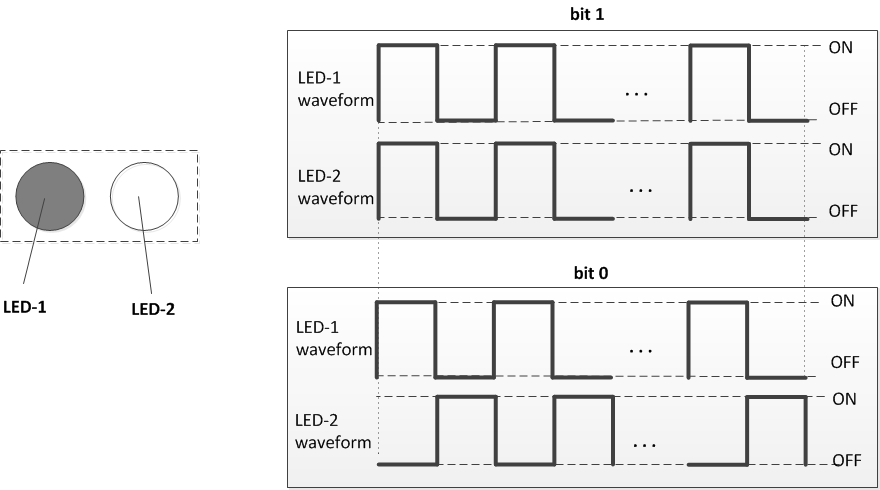
## **13.3 S2-PSK**

### 13.3.1 S2-PSK Encoder

**13.3.1.1 Bit-to-symbol mapping**

The signal to each LED is a square waveform. Two waveforms to drive a pair of LEDs shall have the same phase or inverse phases depending on a data bit input (see Figure A1). The optical clock rate equivalent to bit rate is configured over the PHY PIB attribute *phyOccOpticalClockRate* that is chosen to be no greater than the camera frame rate (e.g. 10 Hz) to ensure every bit is sampled at least once.

The configuration of modulation rate is performed over the PHY PIB attribute *phyS2pskModulationRate* that ensures being non-flicker. Usually, the constant value of the modulation rate is chosen at 200Hz for the indoor environment and 125Hz for the outdoor environment.

****

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

**13.3.1.4 Line Coding**

The RLL coder shall be optionally 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** | **one bit time** | **one bit time** |
| Data bit | 0 | 1 |
| RLL coding | 0 0 | 0 1 |

After RLL coding, the output sequence shall be feed into S2-PSK Encoder.

### 13.3.2 S2-PSK Error Correction

The configuration of error correction for S2-PSK shall be implemented over the PHY PIB attribute *phyOccFec.*

By default, no error correction is used for S2-PSK. However, a majority bit voting shall be 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.

### 14.3.2 S2-PSK dimming Support

S2-PSK dimming is achieved by amplitude modulation as described in the sub-clause 4.5.3.1.5 Low-Clock-Rate OOK amplitude dimming.

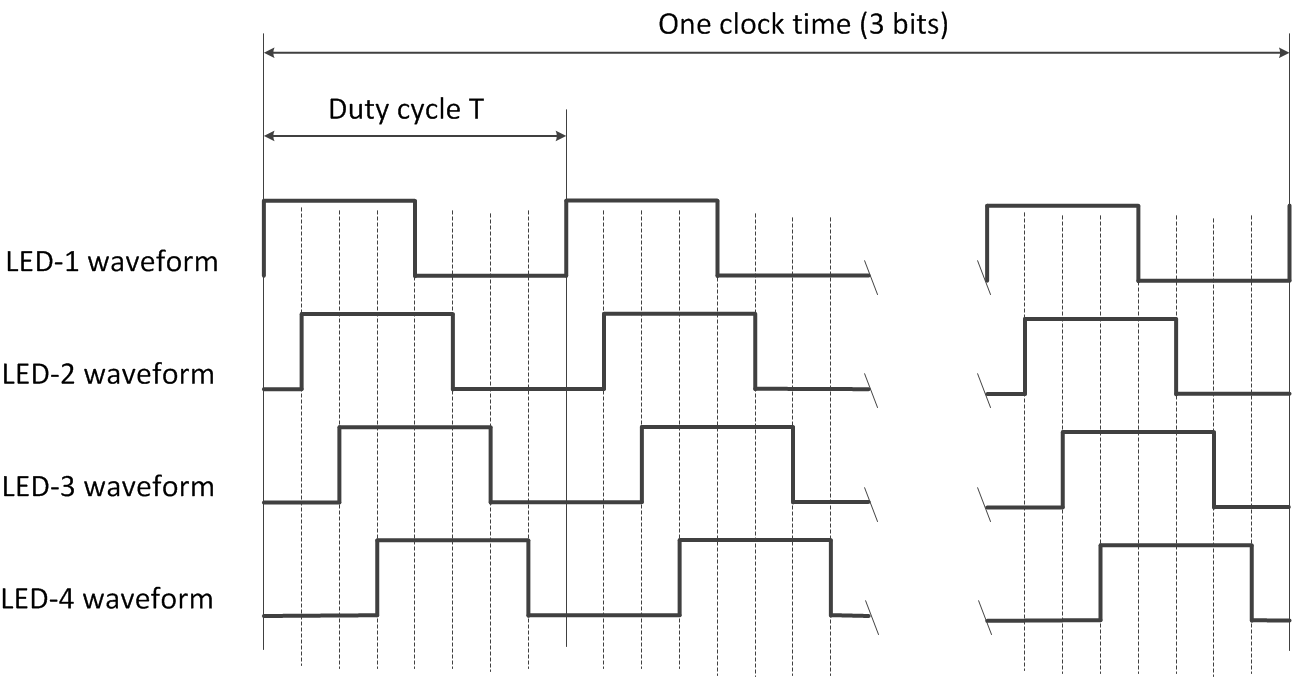
The configuration of dimming level for S2-PSK shall be implemented over the PHY PIB attribute *phyOccDim.*

## **13.4. S8-PSK**

### 13.4.1 S8-PSK Encoder

By default, four-LEDs are grouped into a light source. However, the number of LEDs per group is configurable over the PHY PIB attribute *phyS8pskNoLightSources*.

Four square waveforms that are at the same modulation rate (i.e. frequency) but different phases (the (i+1)th waveform is delayed 1/8 duty cycle compared to the ith waveform as shown in Fig. A2) are used to drive a group of LEDs. The modulation rate of waveform signals is configured over the PHY PIB attribute *phyS8pskModulationRate*.



**Figure A2. Waveforms to drive a group of four LEDs**

A pair of two light sources, each light source is a group of four-LEDs, is used to transmit 3 bits at once 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 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 A3.

**Table A3. Mapping table from bits to the value of *i*= S\_Phase\_Shift/**(T/8)

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

Grey coding is optionally used to map 3 bits into the value of *i*. To enable Grey coding for S8-PSK, the configuration shall be implemented over the PHY PIB attribute phyOccRLLcode=1.

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

### 13.4.2 S8-PSK Decoder

At a sampling time, a camera captures two groups of light sources; and from an image, each group forms a set of four ON/OFF states. 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 four states. For both cases, the captured set of a group shall be represented by **S\_Phase** as shown in Table A4.

**Table A4. S\_Phase Determination**

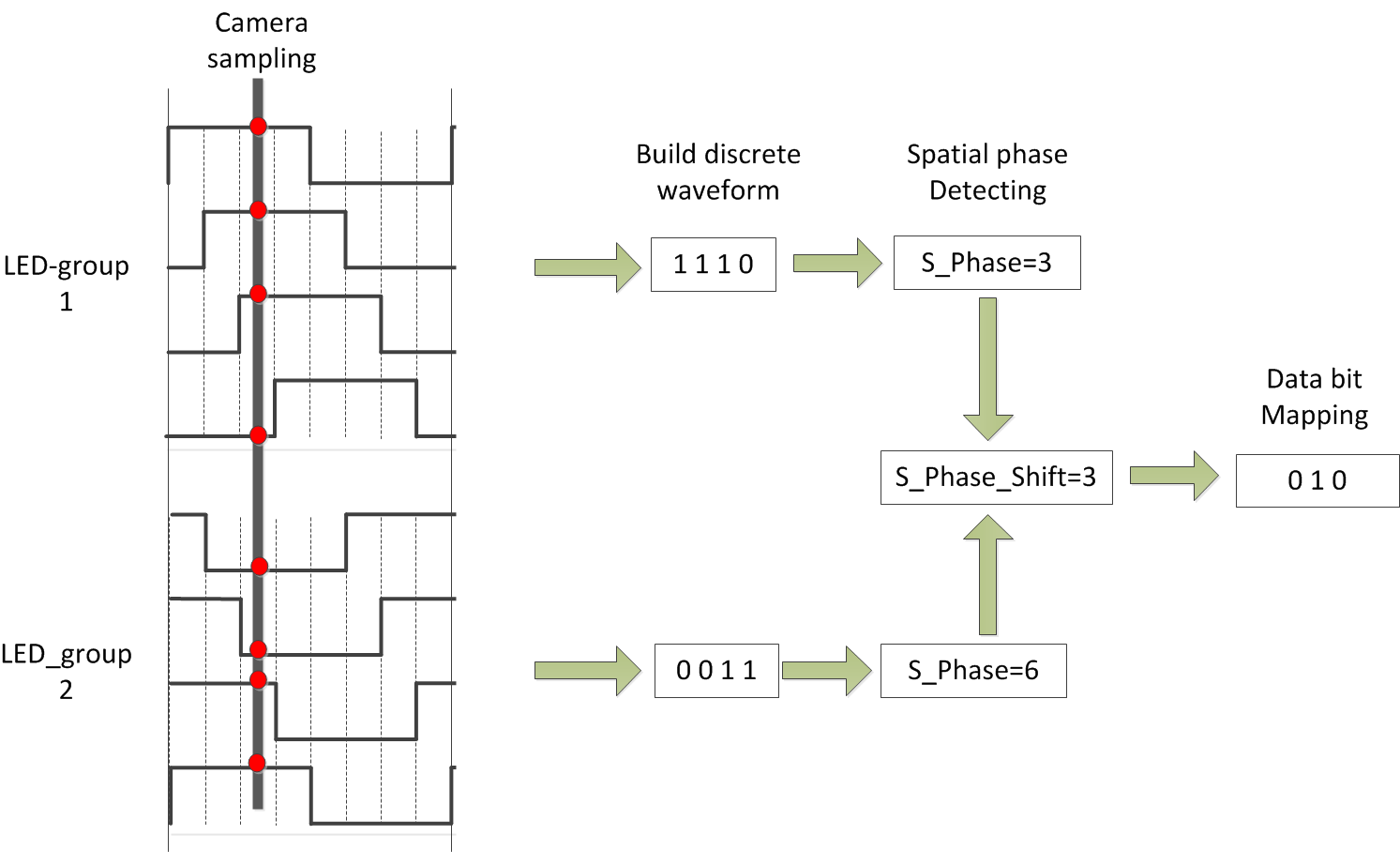
|  |  |  |
| --- | --- | --- |
| Set of 4-States  (w/o presence x\_state)  **Input** | Set of 4-States  (w/ presence of x\_state)  **Input** | S\_Phase  **Output** |
| 1000 | 1x00 | 1 |
| 1100 | 11x0 | 2 |
| 1110 | 111x | 3 |
| 1111 | x111 | 4 |
| 0111 | 0x11 | 5 |
| 0011 | 00x1 | 6 |
| 0001 | 000x | 7 |
| 0000 | x000 | 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)***

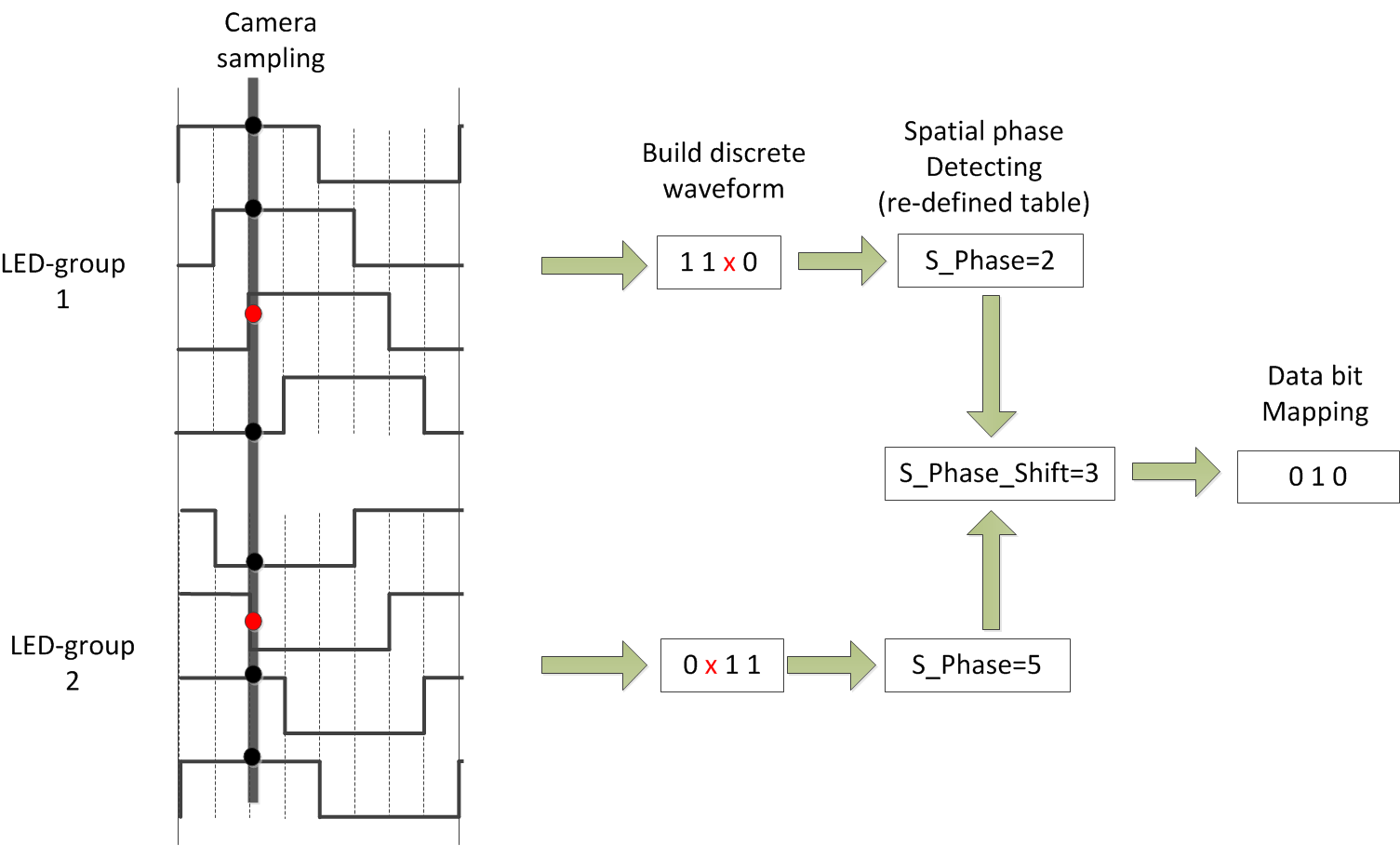
Finally, the de-mapping from S\_Phase\_Shift to 3 bits shall be done inversely as the mapping table (**Table A3)** showed.

**Example 1:** Decoding under none-presence of bad-sampling

****

**Figure A5 – An example of S8-PSK decoding**

**Example 2:** Decoding under presence of *bad-sampling*



**Figure A6. An example of S8-PSK decoding under the presence of bad-sampling**

### 13.4.3 S8-PSK Error Correction

The configuration of error correction for S8-PSK shall be implemented over the PHY PIB attribute *phyOccFec.*

By default, no error correction is used for S8-PSK. However, a majority bit voting shall be 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. Also, the decoding under the bad-sampling condition is considered as an error correction with the code rate equals 1.

### 13.4.4 S8-PSK Dimming

S8-PSK dimming is achieved by amplitude modulation as described in the sub-clause 4.5.3.1.5 Low-Clock-Rate OOK amplitude dimming.

The configuration of dimming level for S8-PSK shall be implemented over the PHY PIB attribute *phyOccDim.*

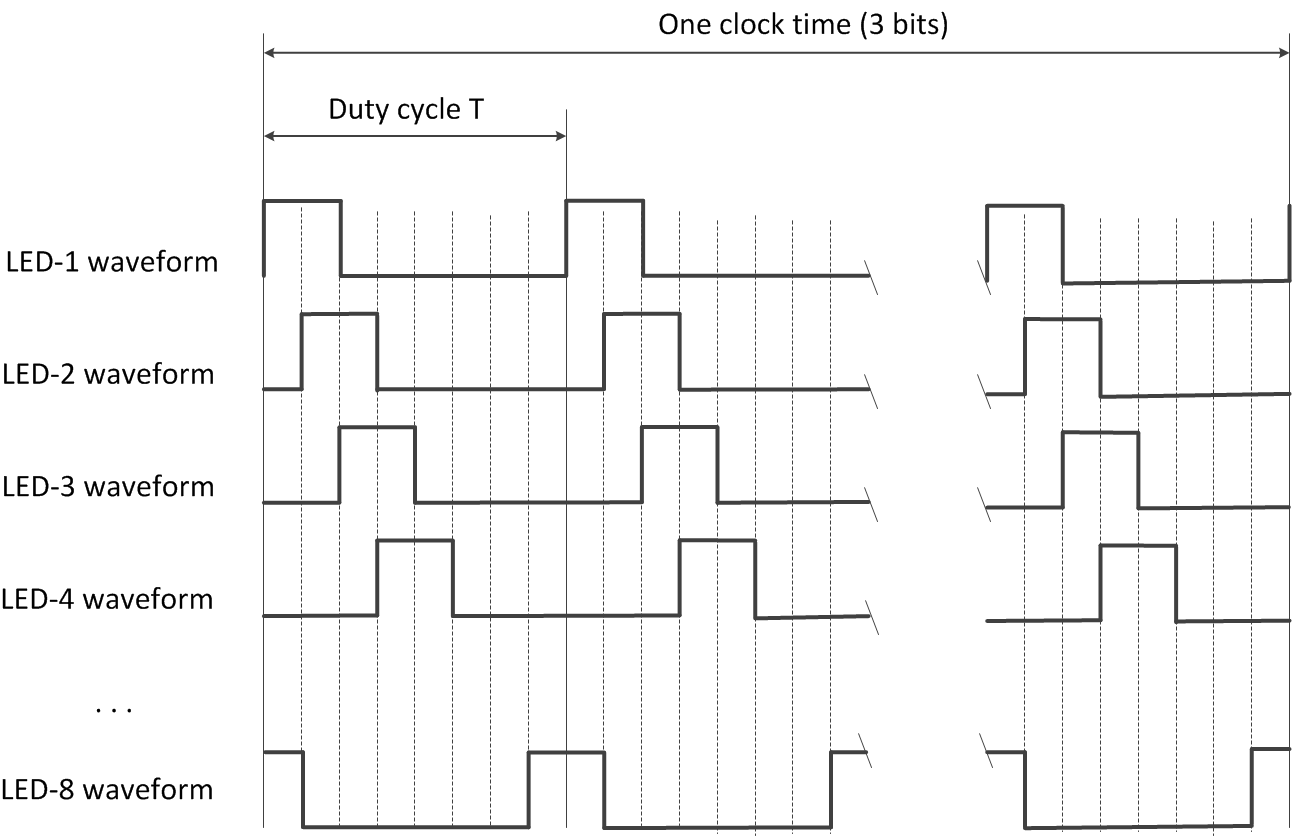
## ***13.5 Hybrid Spatial-Phase Shift Keying***

## **13.5.1 DS8-PSK**

### 13.5.1.1 DS8-PSK Encoder

By default, eight-LEDs are grouped into a light source. However, the number of LEDs per group is configurable over the PHY PIB attribute *phyHSpskNoLightSources*.

Eight square waveforms that are at the same modulation rate (i.e. frequency) but different phases (the (i+1)th waveform is delayed 1/8 duty cycle compared to the ith waveform as shown in Fig. A7) are used to drive a group of LEDs. The modulation rate of waveform signals is configured over the PHY PIB attribute *phyHSpskModulationRate*.



**Figure A7. Waveforms to drive a group of eight LEDs (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 at once 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 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 (i.e. the frequency at which a block of 3-bit is clocked out) shall be configured over the PHY PIB attribute *phyOccOpticalClockRate.*

### 13.5.1.2 DS8-PSK Decoder

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 captured set of a group shall be represented by **S\_Phase** as shown in Table A9.

**Table A9: 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 A10.

**Table A10: 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 A8)** showed.

### 13.5.1.3 DS8-PSK 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 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.

### 13.5.1.4 DS8-PSK Dimming

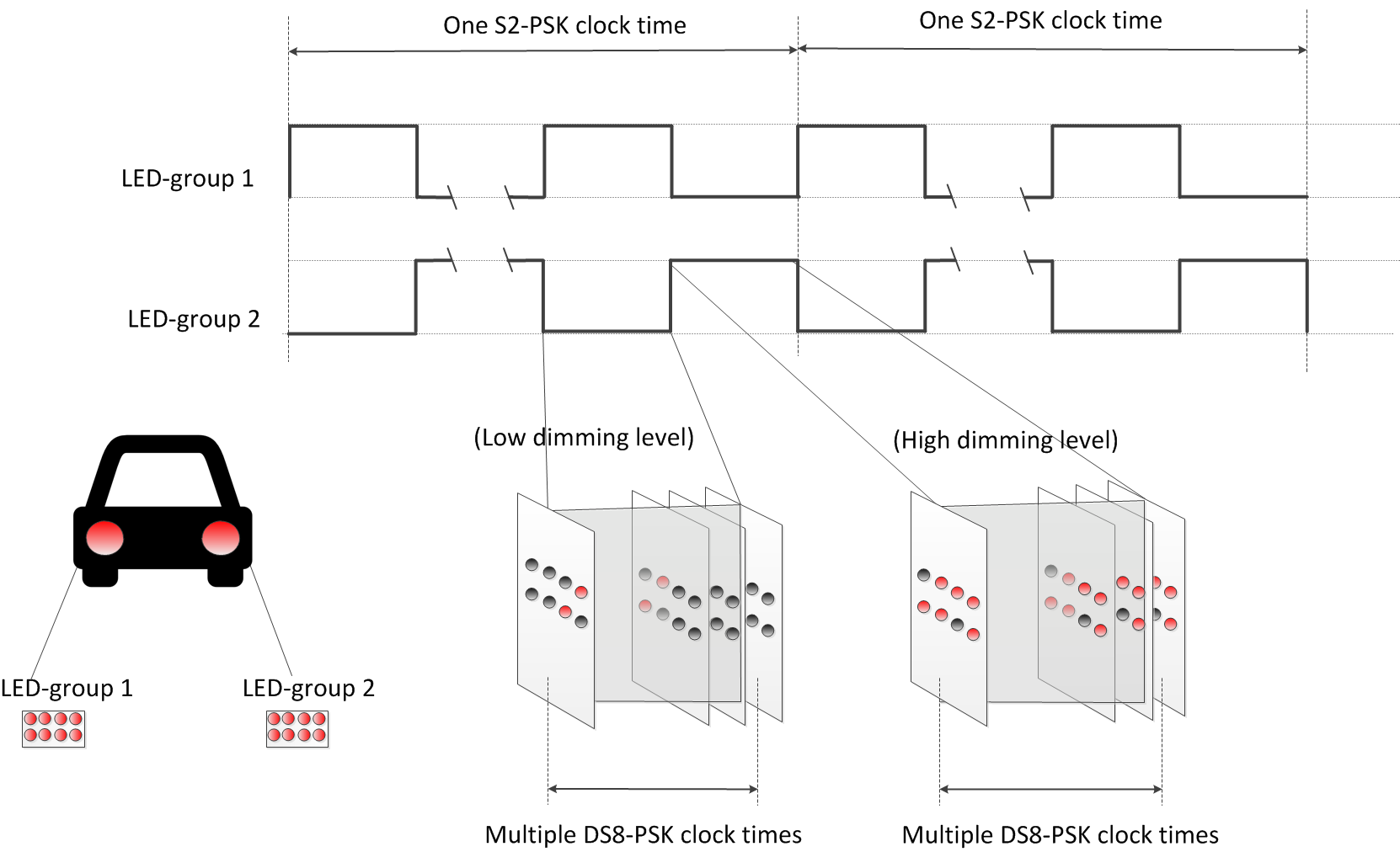
The dimming is performed by controlling the ON time of a duty cycle of all waveforms, not the amplitude. For DS8-PSK, dimming is supported in steps of 1/8 (12.5%). For other DSM-PSK, dimming is supported in steps of 1/M.

The configuration of dimming level is performed over the PHY PIB attribute *phyOccDim.*

## **13.5.2 HS-PSK**

### 13.5.2.1 HS-PSK Encoder

The DSM-PSK (e.g. DS8-PSK) encoder 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). 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 DSM-PSK encoding that generates an AM signal at a low frequency (such as 200Hz or 125Hz) shall be controlled by the S2-PSK encoder (see Figure A11).



**Figure A11 –HS-PSK for vehicular light sources**

### 13.5.2.2 HS-PSK Decoder

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.

### 13.5.2.3 HS-PSK Error Correction

The configuration of error correction for HS-PSK shall be implemented over the PHY PIB attribute *phyOccFec.*

By default, no error correction is used for S2-PSK. However, a majority bit voting shall be 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.

For DS8-PSK in HS-PSK, RS (15, 11) shall be optionally used. Also, a majority bit voting shall be 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.

### 13.5.2.4 HS-PSK Dimming Support

The selection of the low dimming level and the high dimming level for DS8-PSK shall output the desired dimming level as following:

Output dimming level = ½ (low dimmed level + high dimmed level)

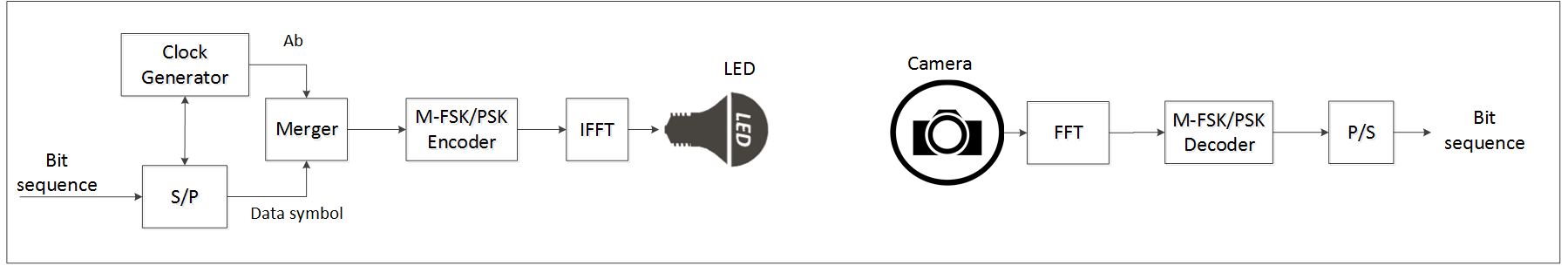
Thus, both low dimming level and high dimming level can control the desired output level of dimming. The configuration of desired dimming level is performed over either the low dimming level (via the PHY PIB attribute *phyHSpskLowDim)* or the high dimming level (via the PHY PIB attribute *phyHSpskHighDim)* or both.

# PHY B Specifications

## **14.2 CM-FSK Modulation**

### 15.2.1 Reference Architecture

The CM-FSK modulation scheme is applied to a system as shown in **figure B1.** A LED is to transmit light modulated with frequencies; and by default, a rolling shutter camera is to receive data.

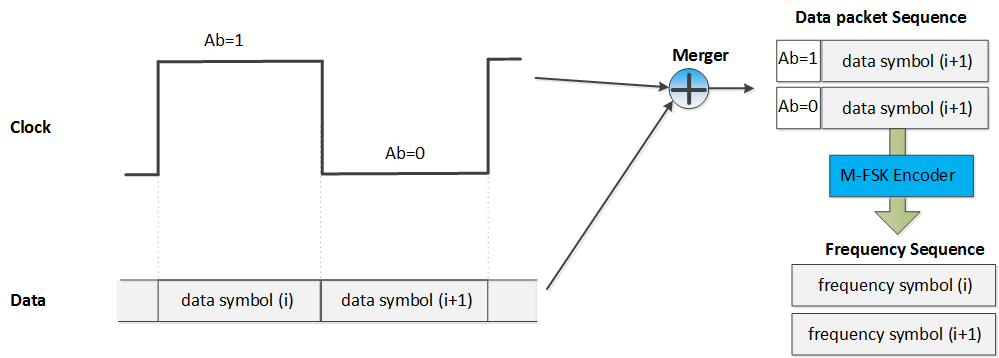
****

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

### 14.2.2 M-FSK/PSK Encoder

By default, a single bit representing clock information called Ab is 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. The configuration of Ab is implemented over the PHY PIB attribute *phyCmfskAb*.

The packet of bits is feed into the M-FSK/PSK encoder to map bits into a frequency symbol. Figure B2 illustrates an example of M-FSK/PSK encoding a sequence of data packets into a sequence of frequency symbols.

****

**Figure B2– An example of M-FSK encoding**

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 over 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.

### 14.2.3 CM-FSK asynchronous communication

Asynchronous communication aims to support a varying frame rate camera demodulating data. A single Ab shall be inserted each data symbol and a camera that has a frame rate greater than the optical clock rate can be used to receive modulated data. The optical clock rate shall control the rate at which frequency symbols are clocked out, being lower than the selected camera frame rate. The configuration of the optical clock rate shall be implemented over the PHY PIB attribute *phyOccOpticalClockRate.*

### 14.2.4 32-FSK Encoding

32-FSK is a specific case of M-FSK/PSK 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* (200Hz by default) and the frequency separation which shall be implemented over the PHY PIB attribute *phyCmfskFrequencySeparation.*

### 14.2.5 64-FSK Encoding

64-FSK is also a specific case of M-FSK/PSK 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**

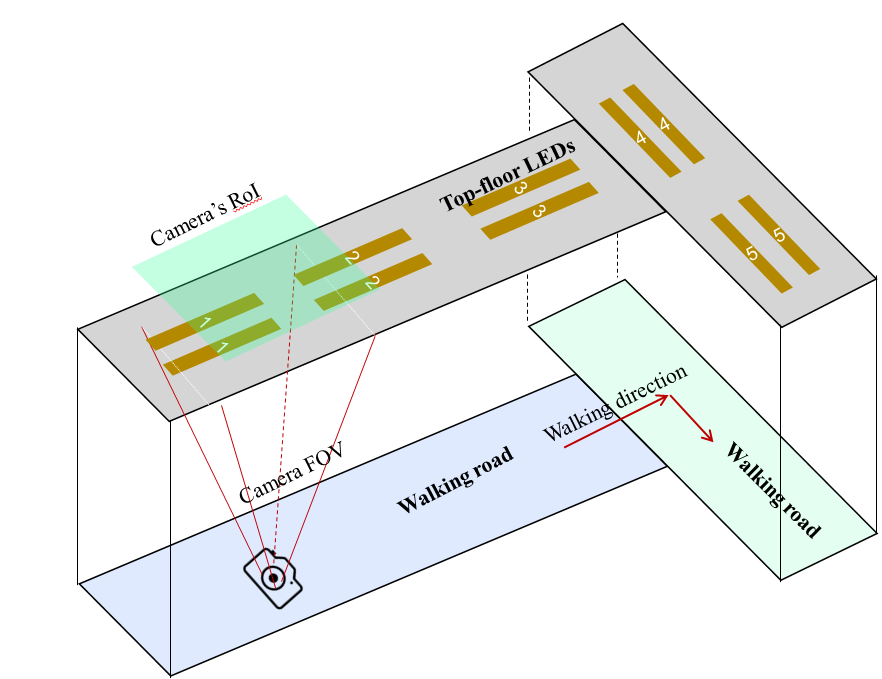
The 64-FSK frequency band is a twice extension of the 32-FSK frequency band. The first 32 data frequencies and two preamble frequencies are the same values as addressed in the 32-FSK modulation. Also, the other 32 frequencies are additionally allocated on the right side of the 32-FSK modulation band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

### 14.2.6 Hybrid Frequency-Phase Shift Keying

The hybrid frequency and phase mode is also a specific case of M-FSK/PSK encoder. The M-FSK encoding is achieved by allocating different frequencies on the selected band. The encoding of M-FSK shall follow 32-FSK encoding or 64-FSK encoding.

Additionally, when *phyCmfskNoPhase =1,* 2-PSK modulation is used on the hybrid modulation to tackle the bandwidth efficiency.

Figure B5 shows a use-case of hybrid M-FSK and 2-PSK. In this case, a pair of light sources utilizes the same M-FSK encoder, and both light sources shall transmit the same frequency at any time. The number of data bits shall be mapped into a frequency to drive both light sources. On the other hand, the relationship of phases of two waveforms shall be modulated by Ab bit. If Ab= “0”, two signals at the same phase are output. If Ab= “1”, two inverse phases shall be applied.



**Figure B5– Example of pairs of LEDs for hybrid M-FSK/2-PSK modulation**

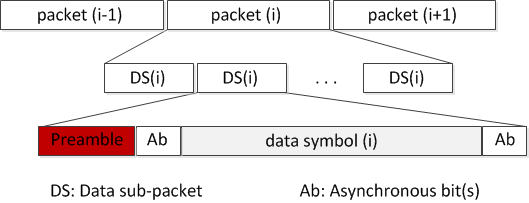
## **14.3 C-OOK**

### 14.3.1 C-OOK Encoder

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 | | |

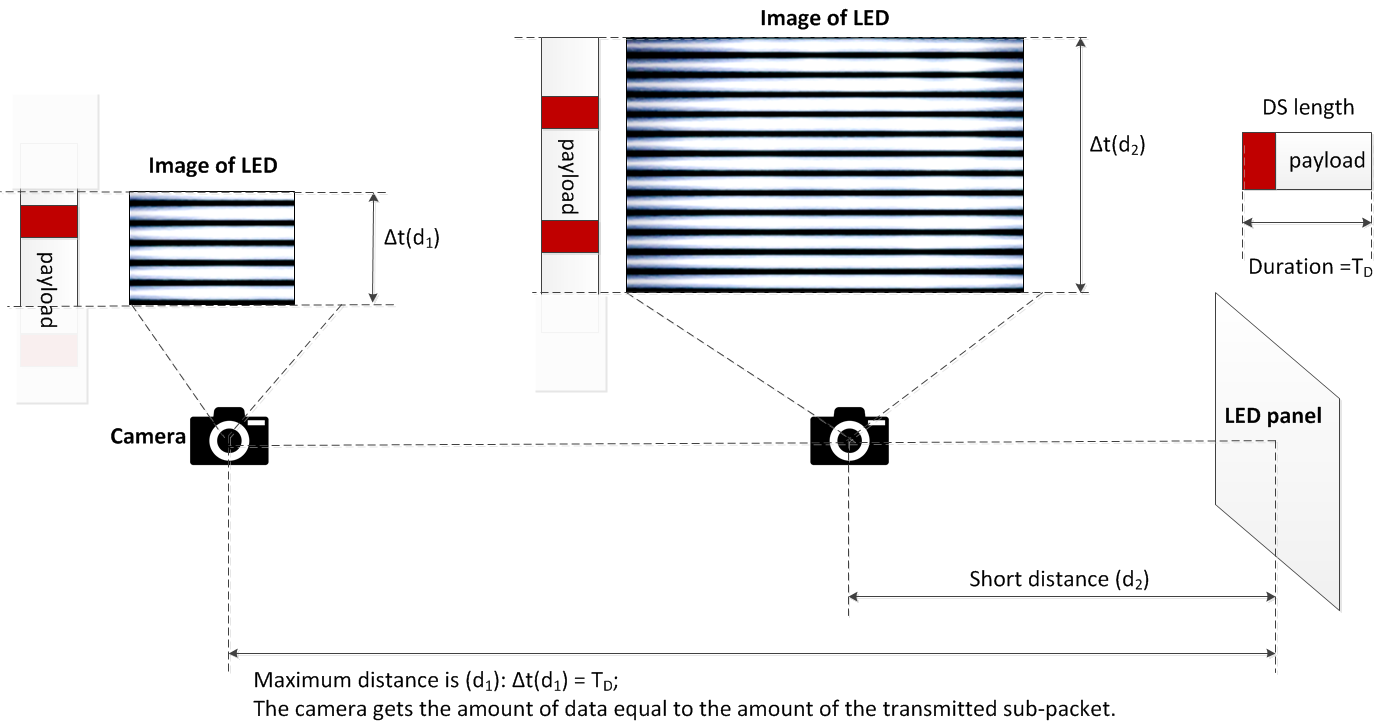
RLL coding shall be applied in payload subfield to maintain the average brightness at 50%. The configuration of RLL code shall be implemented over the PHY PIB attribute *phyOccRLLCode*. Manchester code and 4B6B code are suggested for C-OOK mode.

The payload of DS 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.

### 14.3.2 C-OOK Asynchronous Decoding

To demodulate the entire data sub-packet DS, the distance from a camera to the LED transmitter should be close enough. Figure B8 shows the relationship between the amount of data being captured by the camera and the distance from the camera to the LED transmitter.



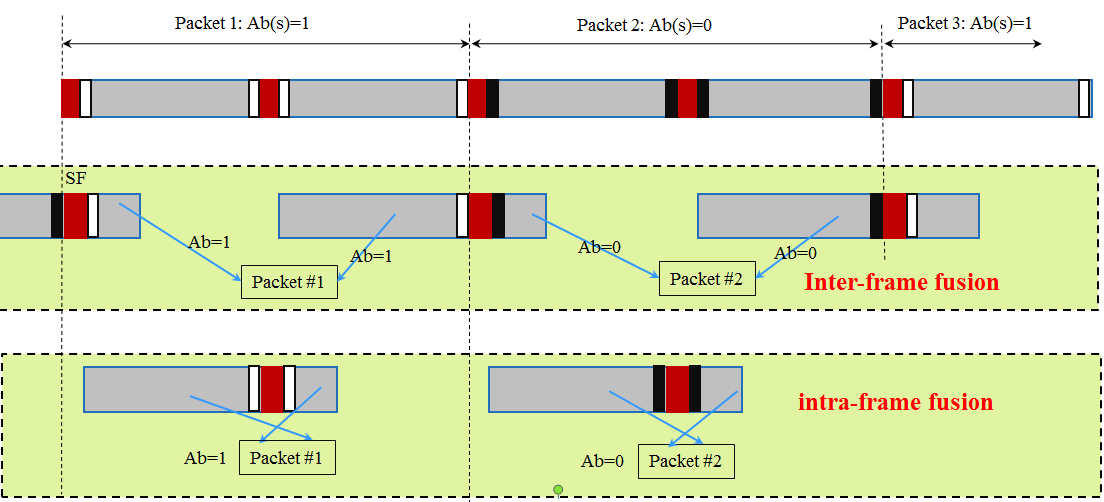
**Figure B8– Decoding scenario**

From the figure B8, 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 B8, 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 B9– Decoding algorithm at a far distance**

Figure B9 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 B8, 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 B10 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 B10– An example of decoding employing intra-frame fusion along with error correction.**

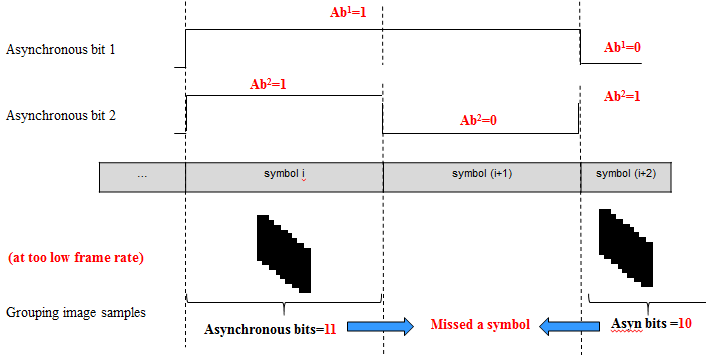
### 14.3.3 Missing packet detection on frame rate drop

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** | **Ab (front)** | **Body payload** | **Ab (rear)** |
|  | 2 bits (Ab1Ab2) | Variable | 2 bits  (Ab1Ab2) |

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



**Figure B12–Asynchronous bits transmission and a missed-symbol Detection**

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.

### 14.3.4 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 |
| Bit rate | 60 bps | 150 bps | 580 bps | 700 bps |

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

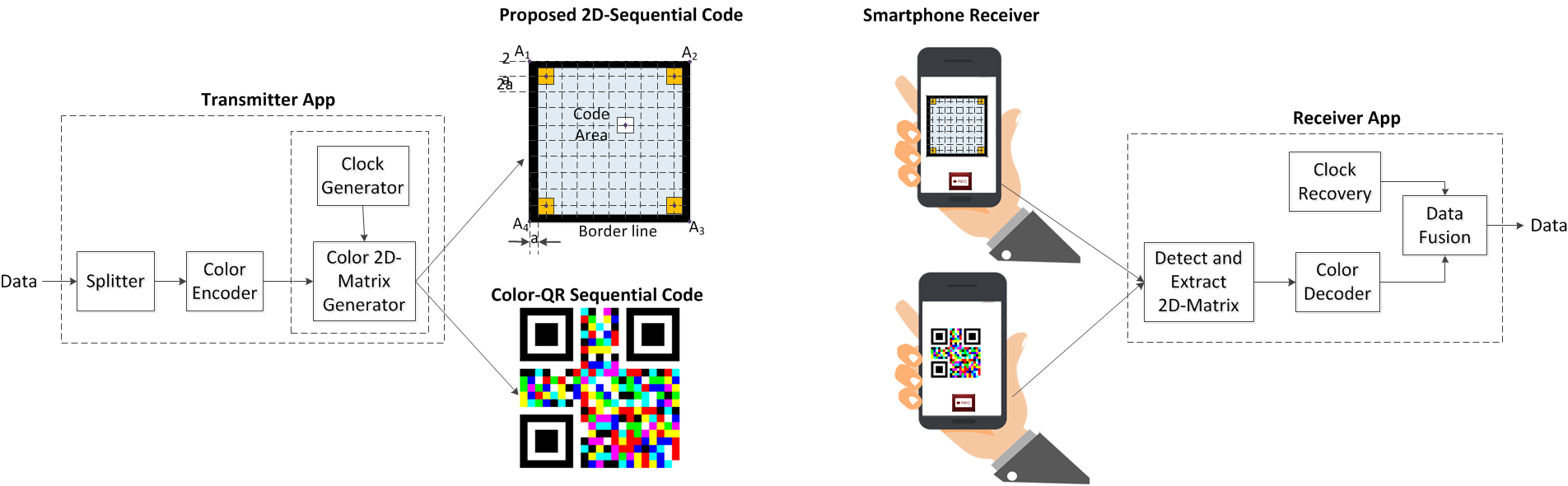
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Mode 1** | **Mode 2** | **Mode 3** | **Mode 4** |
| Preamble | 6B | 10B | 6B | **10B** |
| Front Ab | 2B | 2B | **4B** | **4B** |
| Payload (body) | 8 bits | 13 bits | 33 bits  (24B) | 41 bits  **(62B)** |
| Rear Ab | 2B | **2B** | **4B** | **4B** |

# PHY C Specifications

## **15.1 A-QL Mode**

### 15.1.1 Reference Architecture

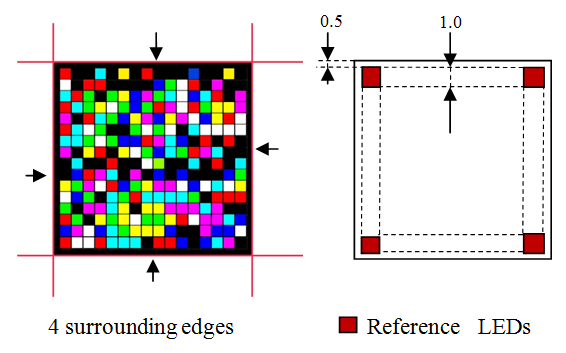
A reference architecture of A-QL is shown in Figure Cx1. The system considers two types of the 2D code transmitter for sequential transmission: (i) existing QR code; and (ii) a new 2D color code. The re-use of QR code interface (version 1 or 2) for sequential transmission requires a sequential communication protocol. Besides, the proposed 2D color code is to minimize the overhead in communications and speed-up the processing on a receiver.



**Figure Cx1 - Reference architecture for 2D-sequential color transmission system**

**15.1.1.1 2D color code design**

Two-dimensional design of color code for sequential transmission is shown inFigure Cx2. By default, four cells at four corners of the code called reference cells are to transmit reference signals (i.e. the transmission of Ab bits and the identification of the starting corner as explained in *subsection 15.1.2 A-QL Encoder*). The other cells are to transmit data, called data cells. The number of reference cells is configurable over the PHY PIB *phyAqlNoCells*.

****

**Figure Cx2 - An example design of 16x16 LEDs transmitter**

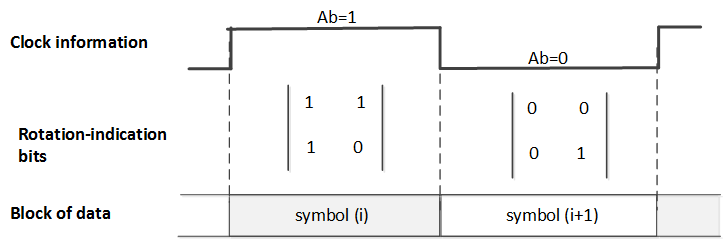
### 15.1.2 A-QL Encoder

A block of bits shall be modulated to the matrix of cells via three color channels, red, green, and blue. The optical clock rate at which data blocks are clocked out is configurable over PHY PIB attribute *phyOccOpticalClockRate.*

In a MxN cells transmitter, each time a block of data bits shall be transmitted over (MxN -4) data cells along with the transmission of four Ab-bits over four reference cells. Note that the number of reference cells is four by default. However it is configurable over PHY PIB *phyAqlNoCells*. The purpose of Ab bits is to support the decoding of a varying frame-rate receiver in performing asynchronous decoding as shown in subsection 15.1.3.

All reference cells shall carry the same Ab which is the clock information (see Figure Cx3) of the data block. The transmission of Ab bits is performed over the red channel.

On the other hand, reference cells shall also carry different bits to indicate the starting corner of the code as shown in Figure Cx3. The group of four bits shall be transmitted over the blue channel called the rotation indication bits. As named, the purpose of rotation-indication bits is to support the receiver identifying the starting corner of the code.



**Figure Cx3 – Clock information and rotating-indication bits for the block of data**

**Table Cx4 –encoding table over three color channels**

|  |  |  |
| --- | --- | --- |
|  | data bit =“0” | data bit = “1” |
| **Red channel** | 0 | 1 |
| **Green channel** | 0 | 1 |
| **Blue channel** | 0 | 1 |

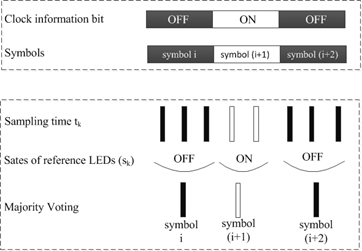
After encoding, each cell, a color channel (red, green or blue) shall carry a single data bit. Together 3 bits of data is mapped into a color of each data cell that human eye sees is shown in table Cx5.

**Table Cx5 – 8 colors encoding table**

|  |  |
| --- | --- |
| **3 bits**  **Input** | **Color**  **Output** |
| 000 | Black |
| 100 | Red |
| 010 | Green |
| 001 | Blue |
| 110 | Yellow |
| 101 | Magenta |
| 011 | Cyan |
| 111 | White |

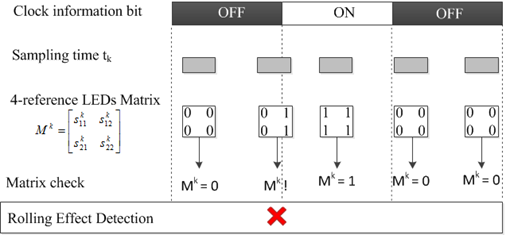
**15.1.3 Asynchronous Communication**

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 Cx4. After that, the majority voting shall be applied for each group of adjacent blocks to down-sample and mitigate error.



**Figure Cx4 –Asynchronous decoding**

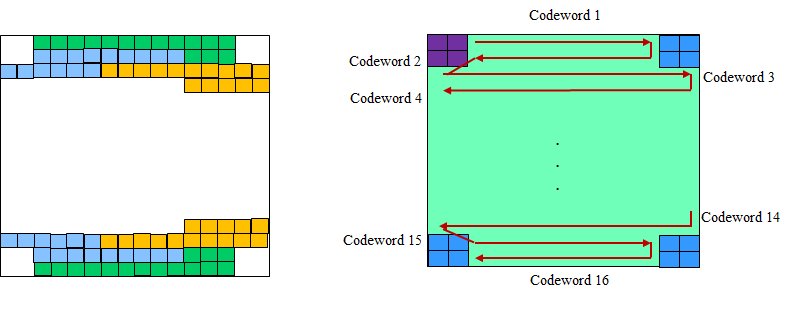
In case a rolling shutter camera captures in between the transition time of two adjacent blocks of data, four Ab bits at four reference cells on the captured image shall not be the same. The image is detected as a rolling affected image as shown in Figure Cx5, and be discarded.



**Figure Cx5 – 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.

**15.1.4 Error correction**

****

**Figure Cx6 –Example of Hamming (11,15) code for 16x16 cells Tx**

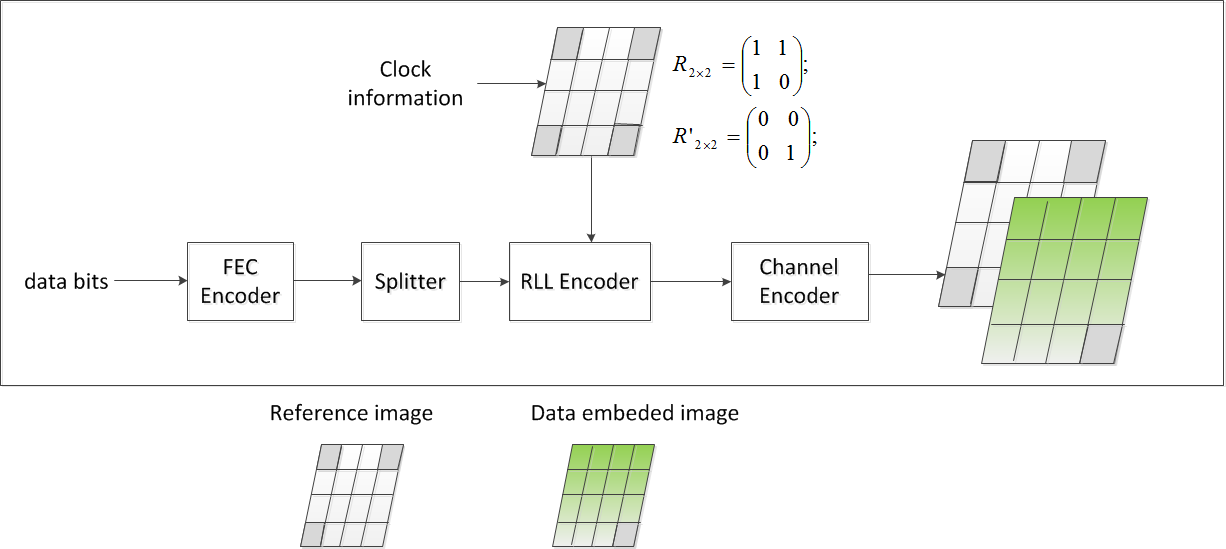
A temporal error correction is applied. The block rate (i.e. the optical clock rate) is 10 Hz, much less than the frame rate of the camera to ensure that every block of data is sampled more than once. The majority voting of all images those sampled on the block of data is to correct the error.

Also, Hamming (11, 15) is optionally used in within the data block. The value of PHY PIB attribute, *phyOccFec*, shall determine whether Hamming code is used or not. In an example of 16x16 cells Tx, 22 bytes information shall be coded into 16 codewords, each consists of 15 bits, to be transmitted at once. Likewise in 8x8 cells Tx, four blocks of data (each consists of 11 bits) shall be coded into four codewords (each consists of 15 bits).

## **15.6 Hidden A-QL**

### 15.6.1 Reference Architecture

The reference architecture for Hidden A-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**

### 15.6.2 Channel encoder



Data embedded image

Reference image (no data)

**Figure x2. Hidden A-QL channel encoder output**

The hidden A-QL encoder divides the Screen transmitter into nxm cells as shown in figure x2. The modulation of screen cells is imperceptible by human eyes but perceptible by the camera receiver.

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

A block of bits shall be modulated to the matrix of cells via a single channel. The optical clock rate at which data blocks are clocked out is configurable over PHY PIB attribute *phyOccOpticalClockRate.*

### 15.6.3 RLL encoder

Also, the data block utilizes 1/2-rate line coding (see table x1). The purpose of the RLL encoder is to enhance the performance of decoding. 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. The reference image does not embed any data information but the clock information via the reference-cells since all the data cells carry unchanged intensity.

**Table x1. RLL encoding table**

|  |  |
| --- | --- |
| **Binary Input** | **Code Output** |
| “0” | 0 0 |
| “1” | 0 1 |

0 state indicates the cell on the image does not change any intensity, whereas 1 state indicates the cell on the image changes.

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

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

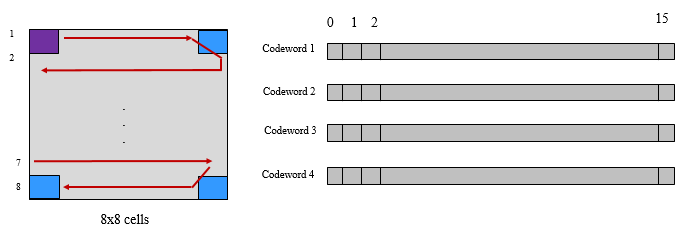
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Data block time (i) | | Data block time (i+1) | |
|  | Reference image | Data emdeded image | Reference image | Data emdeded image |
| cells Transmitter | invisible 1.png | C:\Users\Trang\Desktop\Warsaw\New folder\invisible\invisible 2.png | invisible 1.png | C:\Users\Trang\Desktop\Warsaw\New folder\invisible\invisible 2.png |
| reference-cells |  |  |  |  |
| Data bits |  | |  | |

where nxm is the number of cells in the transmitter.

R2×2 is the states of four reference-cells that carry the clock bit as well as the rotation identification of the starting corner of the code.

The clock-and-rotation information transmitted by four reference cells is to support the receiver decoding under the presence frame rate variation and the rotation of receiver (as shown in subsection 15.1.3. Asynchronous Communication and subsection 15.1.4. 360-degree Rotation Decoding Support).

### 15.6.4 FEC encoder



**Figure 326 –Example of Hamming (11,15) code for 8x8 cells Tx**

Hamming (11, 15) is optionally used in within the data block. The value of PHY PIB attribute, *phyOccFec*, shall determine whether Hamming code is used or not. In an example of 8x8 invisible cells Tx, four words of 11 information-bits shall be coded into four codewords, each consists of 15 bits, to be transmitted at once.

# Annex L:

# OCC Decoding Methods

### 13.4.2 S8-PSK Decoder

At a sampling time, a camera captures two groups of light sources; and from an image, each group forms a set of four ON/OFF states. 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 four states. For both cases, the captured set of a group shall be represented by **S\_Phase** as shown in Table A4.

**Table A4. S\_Phase Determination**

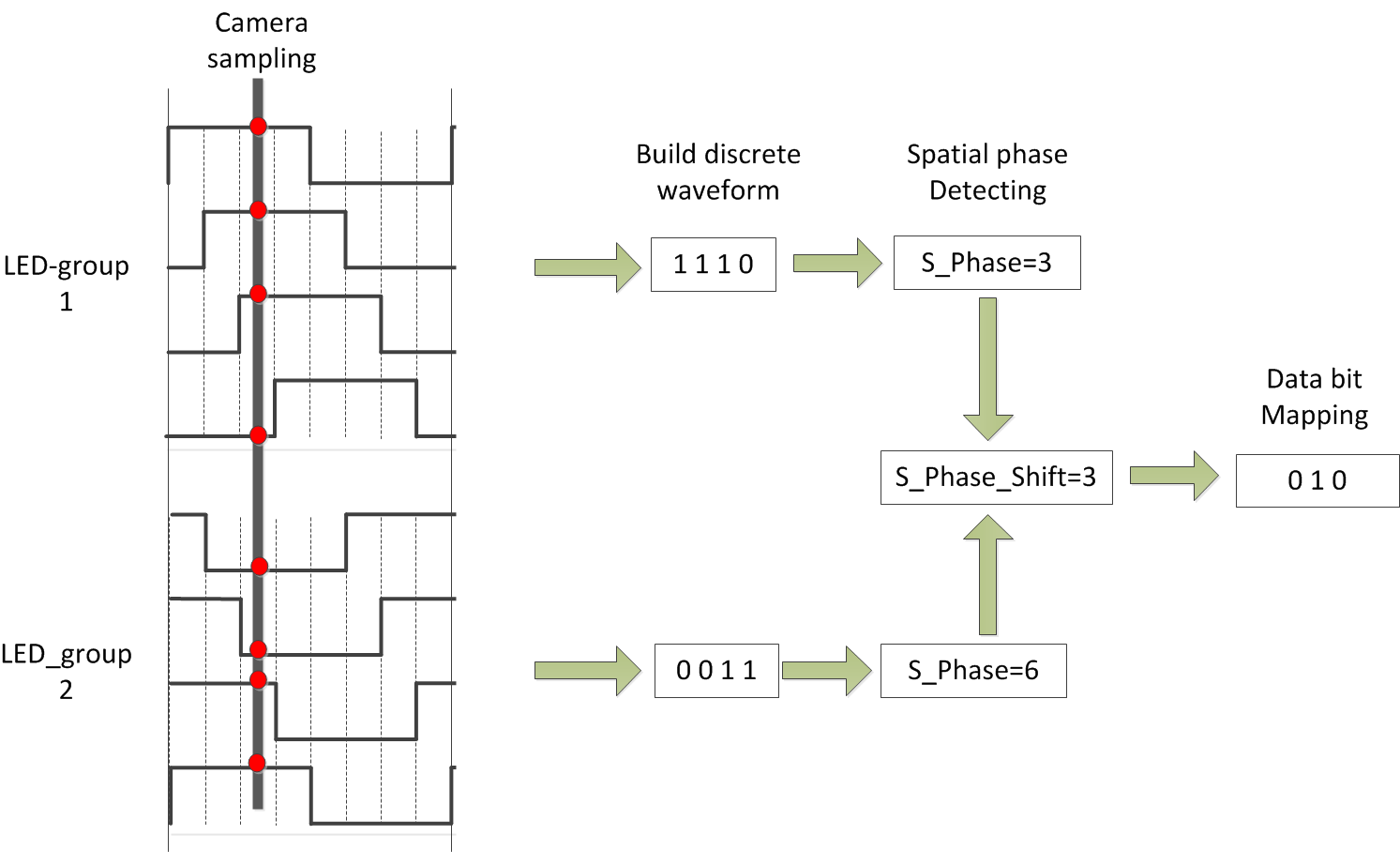
|  |  |  |
| --- | --- | --- |
| Set of 4-States  (w/o presence x\_state)  **Input** | Set of 4-States  (w/ presence of x\_state)  **Input** | S\_Phase  **Output** |
| 1000 | 1x00 | 1 |
| 1100 | 11x0 | 2 |
| 1110 | 111x | 3 |
| 1111 | x111 | 4 |
| 0111 | 0x11 | 5 |
| 0011 | 00x1 | 6 |
| 0001 | 000x | 7 |
| 0000 | x000 | 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)***

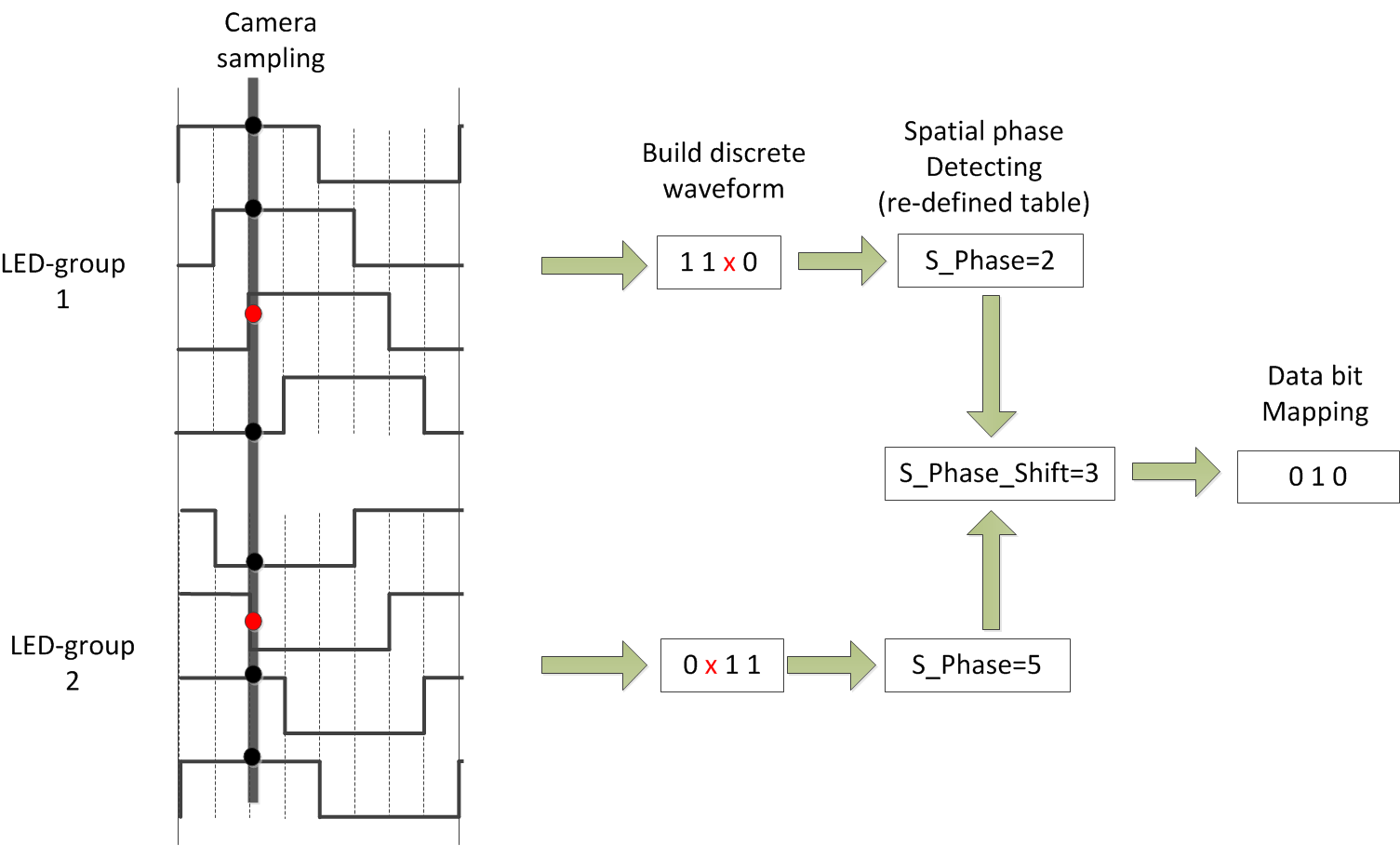
Finally, the de-mapping from S\_Phase\_Shift to 3 bits shall be done inversely as the mapping table (**Table A3)** showed.

**Example 1:** Decoding under none-presence of bad-sampling

****

**Figure A5 – An example of S8-PSK decoding**

**Example 2:** Decoding under presence of *bad-sampling*



**Figure A6. An example of S8-PSK decoding under the presence of bad-sampling**

### 13.5.1.2 DS8-PSK Decoder

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 captured set of a group shall be represented by **S\_Phase** as shown in Table A9.

**Table A9: 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 A10.

**Table A10: 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 A8)** showed.

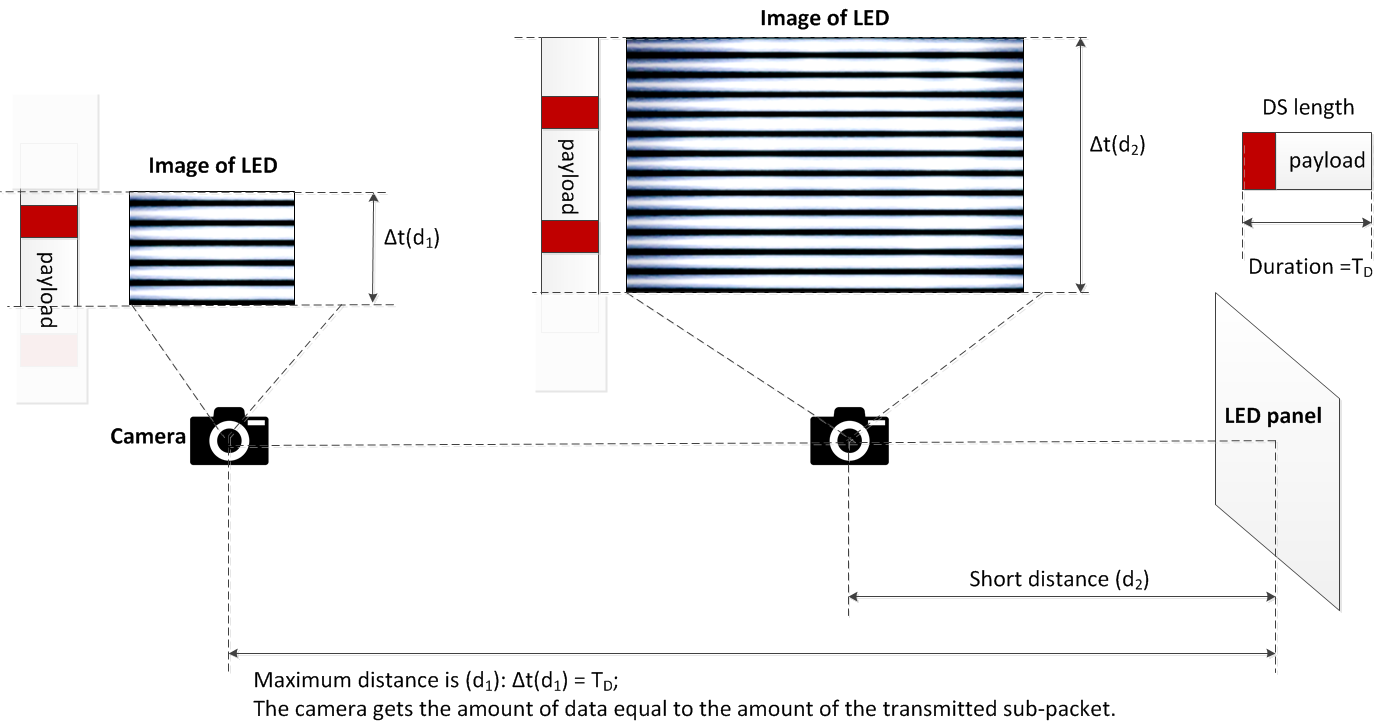
### 13.5.2.2 HS-PSK Decoder

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.

### 14.3.2 C-OOK Asynchronous Decoding

To demodulate the entire data sub-packet DS, the distance from a camera to the LED transmitter should be close enough. Figure B8 shows the relationship between the amount of data being captured by the camera and the distance from the camera to the LED transmitter.



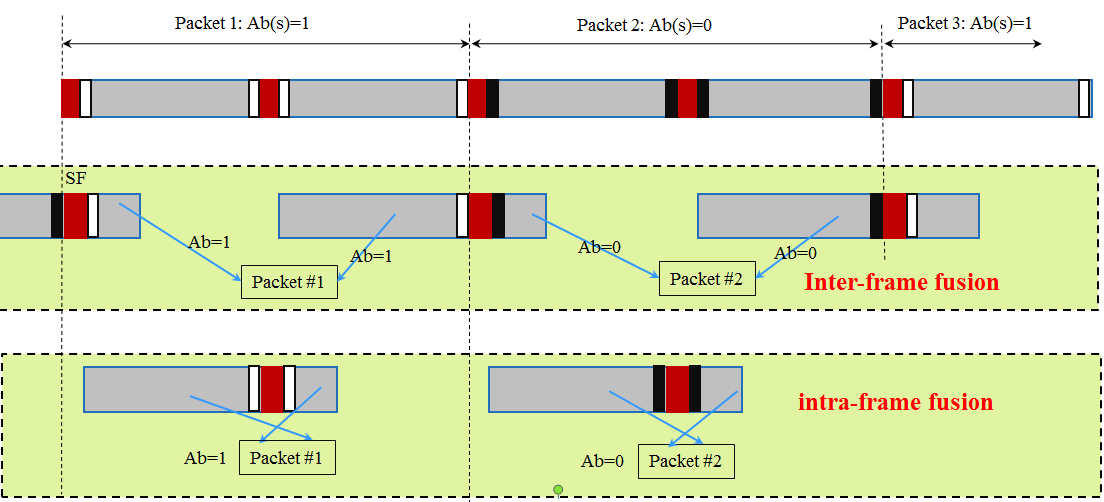
**Figure B8– Decoding scenario**

From the figure B8, 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 B8, 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 B9– Decoding algorithm at a far distance**

Figure B9 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 B8, 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 B10 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.

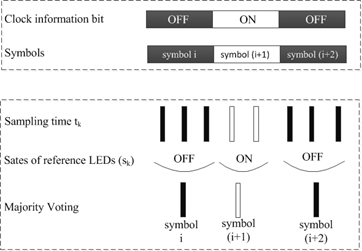
Assume that the camera frame rate may vary but be greater than the packet rate of transmission. Therefore, any extra data after fusion is useful for the error correction by grouping multiple images which belong to a sub-packet to vote. The voting is on the amount of data grouped from all of the forward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



**Figure B10– An example of decoding employing intra-frame fusion along with error correction.**

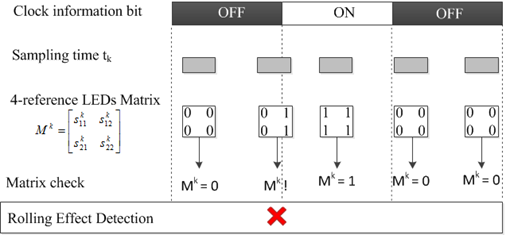
**15.1.3 A-QL Asynchronous Decoding**

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 Cx4. After that, the majority voting shall be applied for each group of adjacent blocks to down-sample and mitigate error.



**Figure Cx4 –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 Cx5, and be discarded.



**Figure Cx5 – 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.