**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |  |
| --- | --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) | |
| Title | **SNUST Text for draft D4 Comment Resolution Revision** | |
| Date Submitted | October, 2017 | |
| Source | Jaesang Cha (SNUST), Soo-Young Chang (CSUS), Vinayagam Mariappan (SNUST) | Voice: [ ] Fax: [ ] E-mail: [chajs@seoultech.ac.kr]1 |
| Re: | SNUST Text for draft D4 Comment Resolution based PHY-IV~VI Specification Revision | |
| Abstract | Details of Resolutions regarding to the submitted Comments on SNUST Text for draft D4 are suggested for PHY-IV ~ VI Specification Revision. The PHY-IV ~ VI is designed to operate on the application services like LED ID, LiFi/CamCom, Digital Signage with Advertisement Information etc. | |
| Purpose | SNUST Text for draft D4 Comments Resolutions and Editorial Revision. | |
| 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. | |

**~~Technical Editor Note …~~**

~~Due to lateness of the text from SNUST, the amount of text involved, and Intel issued computer hardware failure (resulting in a large project work backlog) it became impossible to integrate the SNUST text into draft D4 in a timely manner for review prior to the September meeting. Therefore, we are reviewing the SNUST text via this separate document with follow-on text and comments being integrated into draft D5.~~

**DOCUMENT COLOR REPRESNTATION**

**BLACK** -> Very Old Text

**RED** -> D4 Resolution Text

**GREEN** -> Text Modified by SNUST after D4 Resolution (Based on 0571r0)

**BLUE** -> Text Modified by SNUST after 0571r0 comments

# **Clause 4**

# **~~VTASC Superframe Structure~~**

# ~~The VTASC display transmitter schemes use unslotted ALOHA; that is, when the VTASC display transmitter has a packet to send, it just sends it. This support with beacon and without beacon support and the transmitter does not do a listen before talk channel activity check.~~

# **~~SS2DC Superframe Structure~~**

~~The SS2DC display transmitter schemes use unslotted ALOHA; that is, that is, when the SS2DC display transmitter has a packet to send, it just sends it. This support with beacon and without beacon support and the transmitter does not do a listen before talk channel activity check.~~

# **~~IDE Superframe Structure~~**

# **~~The IDE display transmitter schemes use unslotted ALOHA; that is, when the IDE display transmitter has a packet to send, it just sends it. This support with beacon and without beacon support and the transmitter does not do a listen before talk channel activity check.~~**

# **~~4.9.1 Offset-VPWM Superframe Structure~~**

# ~~The Offset-VPWM for smart device flash light PHY uses the unslotted ALOHA. That means, when the smart device flash light transmitter has a packet to send, it just transmit the data. This support with beacon and without beacon support and the transmitter does not do a listen before talk channel activity check.~~

# **4.8 Some access mechanism by PHY types**

# The UFSOOK, Twinkle VPPM, Offset-VPWM, MPM, VTASC, SS2DC, and IDE transmitter schemes use unslotted ALOHA; that is, when the transmitter has a packet to send, it just sends it. There is no beacon and the transmitter does not do a listen before talk channel activity check. The superframe structure Figure 19 consists of only the contention access period (see subclause 5.1.1.1.2).

# **Clause 5**

**5.2.1.1 Frame control field**

**5.2.1.1.4 PHY VI**

This field is not used except the VTASC, SS2DC, and IDE PHY modes.

**~~VTASC frame control~~**

The frame version subfield specifies the version number corresponding to the frame. This subfield shall be set to 0b01 to indicate a frame compatible with IEEE Standard 802.15.7r1 and all other subfield values shall be reserved for future use.

The frame type subfieldspecifies the frame type used in PHY VI modes MAC frame. This field shall be set to one of the non-reserved values listed in Table 10.

**Table** **10 – VTASC Frame Type Subfield**

|  |  |
| --- | --- |
| Frame Type Value  b2 b1 b0 | Description |
| 000 | Beacon |
| 001 | Data |
| 010 | Acknowledgement |
| 011 | Command |
| 100~111 | Reserved |

The security enabled subfieldspecifies the s~~S~~ecurity on d~~D~~ata f~~F~~rame is enable or not on transmission. This field is 1 bit in length, and it shall be set to one if the frame is protected by the MAC sublayer and shall be set to zero otherwise. The a~~A~~uxiliary s~~S~~ecurity h~~H~~eader field of the MHR shall be present only if the s~~S~~ecurity e~~E~~nabled subfield is set to one.

The frame pending subfield specifies the p~~P~~ending on d~~D~~ata f~~F~~rame is available or not on transmission. This field is 1 bit in length and shall be set to one if the device sending the frame has more data for the recipient. This subfield shall be set to zero otherwise.

The acknowledgment request subfield specifies whether an acknowledgment is required from the recipient device on receipt of a data or MAC command frame. This field is 1 bit in length and this subfield is set to one, the recipient device shall send an acknowledgment frame. If this subfield is set to zero, the recipient device shall not send an acknowledgment frame.

**~~SS2DC frame control~~**

~~The SS2DC frame control follows the same format as VTASC frame control. Refer VTASC frame control for more details.~~

**~~IDE frame control~~**

~~The IDE frame control follows the same format as VTASC frame control. Refer VTASC frame control for more details.~~

**5.2.1.2 Sequence Number field**

**5.2.1.2.4 PHY VI**

This field is not used except the VTASC, SS2DC, and IDE PHY modes.

**~~VTASC Sequence Number~~**

The Sequence Number field is 1 octet in length and specifies the sequence identifier for the frame.

~~For a beacon frame, the Sequence Number field shall specify a BSN. For a data, acknowledgment, or MAC command frame, the Sequence Number field shall specify a DSN that is used to match an acknowledgment frame to the data or MAC command frame.~~

**~~SS2DC Sequence Number~~**

~~The SS2DC sequence number follows the same format as VTASC sequence number. Refer VTASC sequence number for more details.~~

**~~IDE Sequence Number~~**

~~The IDE sequence number follows the same format as VTASC sequence number. Refer VTASC sequence number for more details.~~

**5.2.1.4 Destination Address field**

**5.2.1.4.4 PHY VI**

This field is not used except the VTASC, SS2DC, and IDE PHY modes.

**~~VTASC destination address~~**

The Destination Address field, when present, is either 2 octets or 8 octets in length, according to the value specified in the Destination Addressing Mode subfield of the frame control field, and specifies the address of the intended recipient of the frame.

A 16-bit value of 0xffff in this field shall represent the broadcast short address, which shall be accepted as a valid 16-bit short address by all devices currently listening to the channel.

This field shall be included in the MAC frame only if the Destination Addressing Mode subfield of the frame control field is nonzero.

**~~SS2DC destination address~~**

~~The SS2DC destination address field follows the same format as VTASC destination address field. Refer VTASC destination address field for more details.~~

**~~IDE destination address~~**

~~The IDE destination address field follows the same format as VTASC destination address field. Refer VTASC destination address field for more details.~~

**5.2.1.6 Source Address field**

**5.2.1.6.4 PHY VI**

This field is not used except the VTASC, SS2DC, and IDE PHY modes.

**~~VTASC source address~~**

The Source Address field, when present, is either 2 octets or 8 octets in length, according to the value specified in the Source Addressing Mode subfield of the frame control field, and specifies the address of the originator of the frame.

This field shall be included in the MAC frame only if the Source Addressing Mode subfield of the frame control field is 10 or 11.

**~~SS2DC source address~~**

~~The SS2DC source address field follows the same format as VTASC source address field. Refer VTASC source address field for more details.~~

**~~IDE source address~~**

~~The IDE source address field follows the same format as VTASC source address field. Refer VTASC source address field for more details.~~

**5.2.1.8 Frame Payload field**

**5.2.1.8.2 PHY IV**

**Offset-VPWM Frame Payload Field**

The offset-VPWM uses only the frame payload. The ~~F~~frame ~~P~~payload field has a variable length and contains information specific to individual frame types. ~~The offset-VPWM uses only the frame payload.~~ If the security subfield is enabled ~~PHY attribute phyOccFEC is set to FCS mode in the frame control field~~ then the frame payload is protected as defined by the security suite selected for that frame.

**5.2.1.8.4 PHY VI**

This field is not used except the VTASC, SS2DC, and IDE PHY modes.

**~~VTASC payload field~~**

The Frame Payload field has a variable length and contains information specific to individual frame types. If the Security Enabled subfield is set to one in the frame control field, the frame payload is protected as defined by the security suite selected for that frame.

**~~SS2DC payload field~~**

~~The SS2DC payload field follows the same format as VTASC payload field. Refer VTASC payload field for more details.~~

**~~IDE payload field~~**

~~The IDE payload field follows the same format as VTASC payload field. Refer VTASC payload field for more details.~~

**5.2.1.9 FCS Field**

**5.2.1.9.2 PHY IV**

**Offset-VPWM FCS Field**

The FCS is calculated over the MHR and MSDU parts of the frame. The FCS shall be only generated for payloads greater than zero bytes. The FCS is ~~option is given as~~ an optional and is given in Annex C. ~~option, it is adaptive to RS/CRC/NONE~~. ~~The FCS is configured through PHY attribute phyOccFEC~~*~~.~~*

~~The FEC supported methods for Offset VPWM is given in Table 11.~~

|  |  |  |
| --- | --- | --- |
| **~~No~~** | **~~RS Method~~** | **~~RS Rate~~** |
| ~~1~~ | ~~None~~ | ~~1~~ |
| ~~2~~ | ~~RS(15,2)~~ | ~~2/15~~ |
| ~~3~~ | ~~RS(15,4)~~ | ~~4/15~~ |

**~~Table 11 – Offset VPWM FEC Outer Code Support~~**

**5.2.1.9.4 PHY VI**

This field is not used except the VTASC, SS2DC, and IDE PHY modes.

**~~VTASC FCS field~~**

The FCS field is 2 octets in length and the FCS is calculated over the MHR and MSDU parts of the frame. The FCS shall be only generated for payloads greater than zero bytes. The FCS is an optional ~~filed~~ and is given in Annex C. ~~in MAC frame format and the field information generated based on~~ ~~payload and FCS option used in the MAC frame from RS (64, 32) / RS (160,128) / None~~.

~~The FEC support for VTASC is given in Table 12.~~

**~~Table 12 – VTASC, SS2DC, IDE FEC Support~~**

|  |  |  |
| --- | --- | --- |
| **~~No~~** | **~~RS Method~~** | **~~FECRate~~** |
| ~~1~~ | ~~None~~ | ~~1~~ |
| ~~2~~ | ~~RS(64,32)~~ | ~~32/64~~ |
| ~~3~~ | ~~RS(160,128)~~ | ~~128/160~~ |

**~~S2DC FCS field~~**

~~The SS2DC FCS field follows the same format as VTASC FCS field. Refer VTASC FCS field for more details.~~

**~~IDE FCS field~~**

~~The IDE FCS field follows the same format as VTASC FCS field. Refer VTASC FCS field for more details.~~

# **Clause 6**

**I. PHY TYPE IV MAC PIB Attributes**

# **6.4.2 MAC PIB Attributes for Offset-VPWM**

The MAC PIB attributes for Offset-VPWM is presented in the Table 97 - MAC PIB attributes (continued for Offset-VPWM).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Identifier** | **Type** | **Range** | **Description** | **Default** |
| ~~macOffsetVPWMDataUsage~~ | ~~0x81~~ | ~~Unsigned~~ | ~~0-255~~ | ~~This attribute indicates the type of data transmitted using Flash Light Transmitter.~~  ~~0 : LED ID without IP address~~  ~~1 : LED ID with IP address~~  ~~3 : Authentication Data~~ | ~~0~~ |

Table ~~97~~64 - MAC PIB attributes (continued for OffsetVPWM)

**II. PHY TYPE VI MAC PIB Attributes**

# **6.4.2 MAC PIB Attributes for VTASC, SS2DC, IDE**

The MAC PIB attributes for VATSC, SS2DC, and IDE is presented in the Table 97 - MAC PIB attributes (continued for VTASC, SS2DC, and IDE).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Identifier** | **Type** | **Range** | **Description** | **Default** |
| ~~mac2DCODETxDataType~~ | ~~0x94~~ | ~~Unsigned~~ | ~~0-255~~ | ~~This attribute indicates the type of data to be transmitted.~~  ~~0 : Normal Data (Media Content, Information Content based on the Application its used)~~  ~~1 : LED ID Data~~  ~~2 : Authentication Data~~ | ~~0~~ |
| ~~macMPDULength~~ | ~~0x95~~ | ~~Integer~~ | ~~0-65535~~ | ~~This attribute specify the length of the data to be transmitted~~ | ~~1~~ |

Table ~~97~~64 - MAC PIB attributes (continued for VTASC, SS2DC, and IDE)

# **~~Clause 8~~**

**~~Table 81 - PHY IV, V and VI Operating Modes~~**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **~~Modulation~~**  **~~(phyOccMcsID)~~** | **~~RLL~~**  **~~(phyOccRLLCode)~~** | **~~Optical Clock Rate~~**  **~~(phyOccOpticalClockRate)~~** | **~~FEC~~**  **~~(phyOccFEC)~~** | **~~Bit Rate~~** |
| ~~UFSOOK~~ | ~~NA~~ | ~~multiple of frame rate~~ | ~~MIMO path dependent~~ | ~~(1/2)\*(code rate)\*Frame Rate)~~ |
| ~~Twinkle VPPM~~ | ~~NA~~ | ~~4x bit rate~~ | ~~RS(15, 11)~~ | ~~4 kbps~~ |
| ~~S2-PSK~~ | ~~Differential code~~ | ~~10 Hz~~ | ~~Temporal error correction~~ | ~~5 bps~~ |
| ~~S8-PSK~~ | ~~Grey code~~ | ~~10 Hz~~ | ~~Temporal error correction~~ | ~~30 bps~~ |
| ~~HS-PSK~~ | ~~½ code rate for S2-PSK; None for DS8-PSK~~ | ~~10 kHz~~ | ~~Temporal error correction;~~  ~~Outer FEC with GF(16)~~ | ~~22 kbps~~ |
| ~~Offset-VPPM~~ | ~~None~~ | ~~25Hz~~ | ~~RS(15,2) / RS(15,4) / None~~ | ~~18bps~~  ~~(FEC None)~~ |

**~~Table 81 - PHY IV, V and VI Operating Modes (Continued)~~**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **~~Modulation~~**  **~~(phyOccMcsID)~~** | **~~RLL~~**  **~~(phyOccRLLCode)~~** | **~~Optical Clock Rate~~**  **~~(phyOccOpticalClockRate)~~** | **~~FEC~~**  **~~(phyOccFEC)~~** | **~~Bit Rate~~** |
| ~~RS-FSK~~ | ~~None~~ | ~~30 Hz~~ | ~~XOR FEC~~ | ~~60/90 bps~~ |
| ~~C-OOK~~ | ~~Manchester/ 4B6B~~ | ~~2.2 kHz/ 4.4 kHz~~ | ~~Temporal error correction~~  ~~(DS rate=100/ DS rate=60)~~ | ~~60/150/580/700 bps~~ |
| ~~CM-FSK~~ | ~~None~~ | ~~10 Hz~~ | ~~Temporal error correction~~ | ~~40/50/60~~  ~~bps~~ |
| ~~Packet PWM/PPM~~ | ~~None~~ | ~~100 kHz~~ | ~~Temporal error correction~~ | ~~5.5/ 8 kbps~~ |

**~~Table 81 - PHY IV, V and VI Operating Modes (Continued)~~**

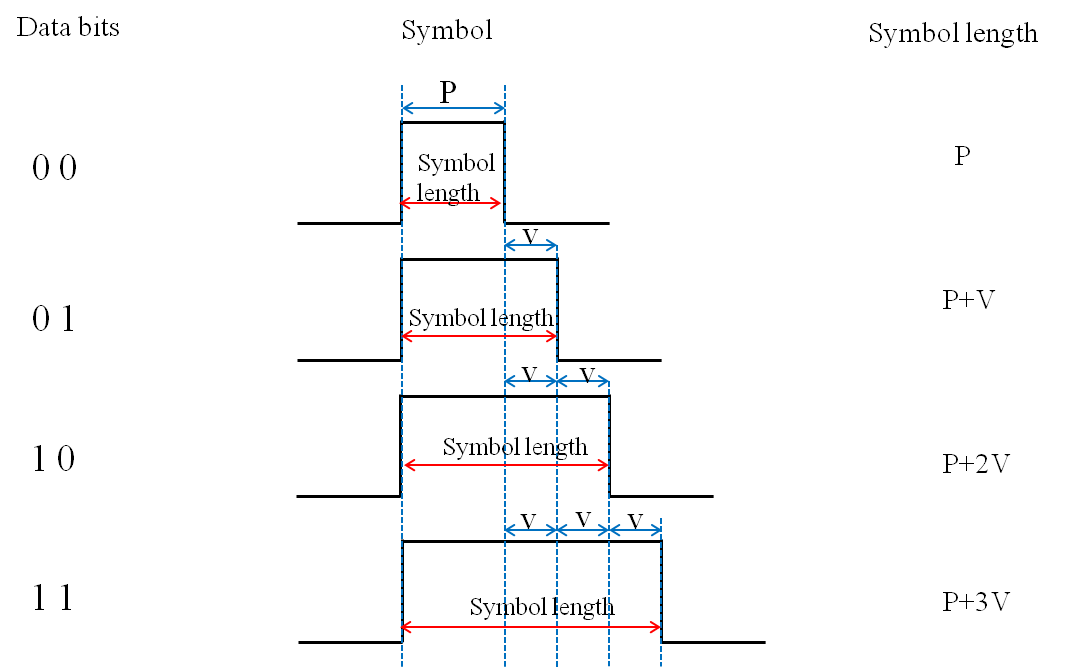
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **~~Modulation~~**  **~~(phyOccMcsID)~~** | **~~RLL~~**  **~~(phyOccRLLCode)~~** | **~~Optical Clock Rate~~**  **~~(phyOccOpticalClockRate)~~** | **~~FEC~~**  **~~(phyOccFEC)~~** | **~~Bit Rate~~** |
| ~~A-QL~~ | ~~None~~ | ~~10 Hz~~ | ~~Hamming (11,15)/ None~~ | ~~5.28/ 7.56 kbps~~  ~~(16x16 cells)~~ |
| ~~HA-QL~~ | ~~Differential code~~ | ~~10 Hz~~ | ~~Hamming (11,15)/ None~~ | ~~220/ 300 bps~~  ~~(8x8 cells)~~ |
| ~~VTASC~~ | ~~None~~ | ~~30 Hz~~ | ~~RS(64,32)/ RS(160,128)/ None~~ | ~~512 Kbps~~  ~~(FEC None)~~ |
| ~~SS2DC~~ | ~~None~~ | ~~30 Hz~~ | ~~RS(64,32)/ RS(160,128)/ None~~ | ~~368 Kbps~~  ~~(FEC None)~~ |
| ~~IDE-MPFSK- BLEND~~ | ~~None~~ | ~~30 Hz~~ | ~~RS(64,32)/ RS(160,128)/ None~~ | ~~32 Kbps~~  ~~(FEC None)~~ |
| ~~IDE-WM~~ | ~~None~~ | ~~30 Hz~~ | ~~RS(64,32)/ RS(160,128)/ None~~ | ~~128 Kbps~~  ~~(FEC None)~~ |

**8.5.2.4.2 Offset-VPWM dimming**

Keep the same text as it is in D4 page no 263.

~~In the Offset-VPWM for smart device flash light uses the smart devices camera LED Flash light sources and uses flickering so no need to concern about dimming. The camera LED flash light is not using for illumination since the flash light blinking speed is very low and can't control dimming.~~

~~The Figure 125 shows the 2bit symbol map flickering control for Offset-VPWM for smart device flash light.~~

~~~~

**~~Figure 125 – 2 Bit Symbol Flickering Control~~**

~~In accordance with the provisions of the symbol, depending on the data bit transmission because the High Pulse interval being determined brightness is adjustable (P > V, V>time error (jitter)).~~

# **~~IDE Dimming~~**

Change text at Page 264 line no 33 to “The ~~flickering and~~ dimming is not supported on PHY VI modes.”

~~The display to camera communication dimming control is depending on the mode of invisible embedding data on display system, rate at which data is repeatedly coding on video frame, and rate at which data refresh on display. The IDE based display transmitter for optical camera communication uses the invisibly embedding the data on video display frame by overlaying patterns on displays visual area using alpha blending and watermarking. Dimming is supported by controlling visual scene high frequency background color.~~

# **~~VTASC Dimming~~**

~~The display to camera communication dimming control is depending on the mode of visible embedding data on display system, rate at which data is repeatedly coding on video frame, and rate at which data refresh on display. The VTASC based display Transmitter for optical camera communication uses the visibly embedding the data on Video display frame by overlaying patterns on displays visual area at optical clock rate from 1Hz to 30Hz that is perceptible by human eyes, dimming is not supported.~~

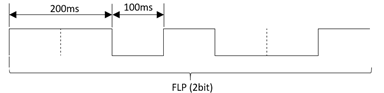
# **~~SS2DC Dimming~~**

~~The display to camera communication dimming control is depending on the mode of visible embedding data on display system, rate at which data is repeatedly coding on video frame, and rate at which data refresh on display. The SS2DCbased Display Transmitter for optical camera communication uses the visibly embedding the data on Video display frame by overlaying patterns on displays visual area at optical clock rate from 1Hz to 30Hz that is perceptible by human eyes, dimming is not supported.~~

**8.6.1.2.2 Offset-VPWM Preamble Field**

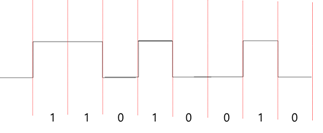
The preamble field is used by the transceiver to obtain optical clock synchronization with an incoming data. ~~The standard defines one fast locking pattern (FLP).~~ ~~The MAC shall select the optical clock rate for communication during the clock rate selection process. The preamble shall be sent at a clock rate chosen by the transmitter and supported by the receiver. The preamble is a time domain sequence and does not have any channel coding or line coding.~~

The preamble first starts with a fast locking pattern (FLP). The FLP is fixed as a pattern “11010010”. ~~The fast locking pattern length shall not exceed the maximum.~~ The timing information for preamble is shown in Figure 132.



**Figure 132 – Preamble Timing Diagram**

In the Offset-VPWM for smart device flash light PHY uses OOK modulation for preamble transmission using flash light. ~~The preamble bit mapping shown in Figure 133.~~



**~~Figure 133 – Preamble Transmission – OFFSET VPWM BIT MAPPING~~**

**~~5.2.1.9.4 PHY VI~~**

**8.6.1.4.1 IDE Preamble Field**

~~The SHR is used by the transceiver to obtain optical clock synchronization with an incoming message is called preamble. The SHR standard defines one fast locking pattern (FLP) followed by choice of four topology dependent patterns (TDPs) for the purposes of distinguishing different PHY topologies is shown in Table 86.~~

The SHR is used with one FLP followed by choice of four TDPs for the purposes of distinguishing different PHY topologies is shown in Table 129 (See 8.6.1.1).

**8.6.1.4.4 SS2DC Preamble Field**

This follows the IDE preamble field mode. See 8.6.1.4.1 for more details.

**8.6.1.4.5 VTASC Preamble Field**

This follows the IDE preamble field mode. See 8.6.1.4.1 for more details.

**8.6.2.4.1 IDE Header Field**

The ~~IDE~~ Header Field is described as shown in Table 97 and shall be transmitted with data to identify the PHY Mode, Data rate, and PSDU length to identify the transmission specification.

**Table 97 – PHY Header**

|  |  |  |
| --- | --- | --- |
| **PHY Header Field** | **Bit-Width** | **Explanation on usage** |
| ~~Burst Mode~~ | ~~1~~ | ~~Reduce Preamble and IFS~~ |
| MCS ID | 6 | Provide information about PHY types and data rate |
| PSDU Length | 16 | Length up to aMaxPHYFrameSize |
| Reserved Fields | 6 | Future use |

~~Burst Mode Field: The burst mode bit indicates that the next frame following the current frame is part of the burst mode. The Burst Mode bit shall be set TRUE if the burst mode is being used otherwise, the Burst Mode bit shall be set FALSE.~~

~~Channel Number Field: The channel number field for PHY shall be the band plan ID of the lowest wavelength. Refer to 9.3.1 for more detailed information.~~

MCS ID Field: The modulation and coding scheme (MCS) ID shall be indicated in the PHY header based on Table 81.

PSDU Length Field: The PSDU length field specifies the total number of octets contained in the PSDU.

**8.6.2.4.4 SS2DC Header Field**

This field follows same as the IDE header field mode. See 8.6.2.4.1 for more details.

**8.6.2.4.5 VTASC Header Field**

This field follows same as the IDE header field mode. See 8.6.2.4.1 for more details.

**8.6.5.2.2 Offset-VPWM PSDU field**

The Offset-VPWM PSDU field has a variable length and carries the arbitrary number of payload bits. The end of the PSDU field is indicated by the presences of another SFD, which means that preamble frame.

**8.6.5.4.1 IDE PSDU Field**

The ~~IDE~~ PSDU field has a variable length and carries the arbitrary number of payload bits based on the block selection. The structure of the PSDU field is as shown in Figure 153.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Block 1 | Block 2 | … | Block N-1 | Block N |
| Data Bits | Symbol 1 | Symbol 2 | … | Symbol N-1 | Symbol N |

**Figure 153 – IDE PSDU Field Structure**

Where the block is MxN pixels and the bits per symbol is as per modulation mode description in clause 15 PHY VI Specifications.

**8.6.5.4.4 SS2DC PSDU Field**

The PSDU field follows the IDE PSDU field mode. See 8.6.5.4.1 for more details.

**8.6.5.4.5 VTASC PSDU Field**

The PSDU field follows the IDE PSDU field mode. Refer 8.6.5.4.1 for more details.

# **Clause ~~10~~9**

**I. PHY TYPE IV PHY PIB Attributes**

# **9.5.2 PHY PIB Attributes for Offset-VPWM**

The PHY PIB attributes for Offset-VPWM is presented in the Table 122~~179~~ —PHY PIB attributes (continued for Offset-VPWM).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Identifier** | **Type** | **Range** | **Description** |
| ~~phySMFlashLIGHTApplicationSpecificMode~~ | ~~0x10~~ | ~~Unsigned~~ | ~~0~255~~ | ~~This attribute specifies the application specific PHY mode.~~  ~~0 : Normal Data (Media Content, Information Content based on the Application used)~~  ~~1 : ID Data~~  ~~2 : Authentication Data~~ |
| phyOffsetVPWMStdPERIOD | 0x11 | Integer | 0-65535 | This attribute specify the standard PWM period used to transmit the data (in micro secs) |
| phyOffsetVPWMOffsetPERIOD | 0x12 | Integer | 0-65535 | This attribute specify the Variable offset PWM period used to transmit the data (in micro secs) |

Table 122~~179~~ — PHY PIB attributes (continued for OffsetVPWM)

**II. PHY TYPE VI PHY PIB Attributes**

# **OCC PHY Modes Identification**

## **Table 126 (new): OCC PHY modes identification**

|  |  |
| --- | --- |
| **phyOccMcsID** | **PHY OCC mode Description** |
| 0 | UFSOOK |
| 1 | Twinkle VPPM |
| 2 | S2-PSK |
| 3 | HS-PSK |
| 4 | Offset-VPPM |
| 5 | RS-FSK |
| 6 | CM-FSK |
| 7 | C-OOK |
| 8 | MPM |
| 9 | A-QL |
| 10 | HA-QL |
| 11 | VTASC |
| 12 | IDE |
| 13 | SS2DC |
| 14-15 | Reserved |

**Note: From Kookmin 0532r0**

# **9.5.2 PHY PIB Attributes for VTASC**

The PHY PIB attributes for VATSC is presented in the Table 122~~179~~ —PHY PIB attributes (continued for VTASC).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PHY PIB Table 188 Additions** | | | | |
| **Attribute** | **Identifier** | **Type** | **Range** | **Description** |
| phyVTASCTxMode | 0x10 | Unsigned | 0-255 | This attribute indicates the VTASC PHY transmission modes.  0 : VTASC Mode  1 : SS VTASC Mode |
| ~~phyVTASCApplicationSpecificMode~~ | ~~0x10~~ | ~~Unsigned~~ | ~~0~255~~ | ~~This attribute specifies the application specific PHY mode.~~  ~~0 : Normal Data (Media Content, Information Content based on the Application used for)~~  ~~1 : LED ID Data~~  ~~2 : Authentication Data~~ |
| phyVTASCTxCameraEnable | 0x92 | Unsigned | 0-255 | This attribute indicates the Transmitter is Enabled with Camera or not for Interactive Receiver distance specific data transfer control.  0 : Camera not connected  1 : Camera connected |
| phyVTASCRxDistance | 0x93 | Unsigned | 0-255 | This attribute notify the Receiver distance from Transmitter |
| PhyVTASCFreq | 0x11 | Unsigned | 0~255 | This attribute specify the frame rate of VTASC sequence Transmission |
| phyVTASCCodeArea | 0x12 | Unsigned | 0~255 | This attribute specify the coded area of the ~~IDE~~ VTASC  0 : Full Display~~Screen~~ Mode  1 : Partial Display~~Screen~~ Mode  2 : LED Bulb Mode  3~255 : Reserved |
| phyVTASCCodeLocation | 0x13 | Unsigned | 0~255 | This attribute specify the Coded Location of the VTASC  0 : Center  1 : Bottom Right  2 : Bottom Left  3 : Top Right  4 : Top Left  5~255 : Reserved |
| phyVTASCTLevel | 0x14 | Unsigned | 0~255 | This attribute specify the transparency Level of the VTASC  0 : One Level (100 % transparency)  1 : Two Level (100 % & 50 % transparency)  2~255 : Reserved |
| phyVTASCALevel | 0x14 | Unsigned | 0~255 | This attribute specify the block size of the VTASC  0 : One Level  1 : Two Level  ~~2 : Three Level~~  2 : Four Level  3~255 : Reserved |
| phyVTASCSLevel | 0x14 | Unsigned | 0~255 | This attribute specify the number of shapes used in the VTASC  0 : One Shape  1 : Two Shapes  ~~2: Three Shapes~~  2 : Four Shapes  3~255 : Reserved |
| phyVTASCCLevel | 0x14 | Unsigned | 0~255 | This attribute specify the number of colors used in the VTASC  0 : One color  1 : Two colors  ~~2 : Three colors~~  2 : Four colors  ~~4 : Five colors~~  ~~5 : Six colors~~  ~~6 : Seven colors~~  3 : Eight colors  4~255 : Reserved |
| ~~phyVTASCSModel~~ | ~~0x17~~ | ~~Unsigned~~ | ~~0~255~~ | ~~This attribute specify the block shape Type used in the VTASC~~  ~~0 : Square~~  ~~1 : Circle~~  ~~3 : hexagon~~  ~~4 : star~~  ~~5~65535 : Reserved~~ |
| phyVTASCAHSize | 0x15 | Unsigned | 0~255 | This attribute specify the no of Horizontal Blocks in the VTASC |
| phyVTASCAVSize | 0x16 | Unsigned | 0~255 | This attribute specify the no of Vertical Blocks in the VTASC |
| phyVTASCScalRateCtrl | 0x18 | Unsigned | 0~255 | This attribute specify the Scalable Rate control mode  0 : No Scalable Bitrate control  1 : Multirate Scalable Controller  2: Distance Adaptive Scalable Controller  3: Distance adaptive with multirate scalable controller |
| phyVTACScalRegion1OpticalClockRate | 0x19 | Unsigned | 0~255 | This attribute specify the scalable optical clock rate of VTASC region 1 |
| phyVTACScalRegion2OpticalClockRate | 0x1A | Unsigned | 0~255 | This attribute specify the scalable optical clock rate of VTASC region 2 |
| phyVTACScalRegion3OpticalClockRate | 0x1B | Unsigned | 0~255 | This attribute specify the scalable optical clock rate of VTASC region 3 |
| phyVTACScalRegion4OpticalClockRate | 0x1C | Unsigned | 0~255 | This attribute specify the scalable optical clock rate of VTASC region4 |
| phyVTACScalRegion1DistanceRange | 0x19 | Unsigned | 0~255 | This attribute specify the distance adapted on VTASC region 1 |
| phyVTACScalRegion2DistanceRange | 0x1A | Unsigned | 0~255 | This attribute specify the distance adapted on VTASC region 2 |
| phyVTACScalRegion3DistanceRange | 0x1B | Unsigned | 0~255 | This attribute specify the distance adapted on VTASC region 3 |
| phyVTACScalRegion4DistanceRange | 0x1C | Unsigned | 0~255 | This attribute specify the distance adapted on VTASC region 4 |
| PhySSCode1Len | 0x1D | Unsigned | 0~255 | This attribute specify the spreading code length for SS Code 1 |
| PhySSCode2Len | 0x1E | Unsigned | 0~255 | This attribute specify the spreading code length for SS Code 2 |
| PhySSCode3Len | 0x1F | Unsigned | 0~255 | This attribute specify the spreading code length for SS Code 3 |
| PhySSCode4Len | 0x20 | Unsigned | 0~255 | This attribute specify the spreading code length for SS Code 4 |
| PhySSCode1FP00 | 0x21 | Integer | 0~65535 | This attribute specify the SS Code 1 pair code 0 |
| PhySSCode1FP01 | 0x22 | Integer | 0~65535 | This attribute specify the SS Code 1 pair code 1 |
| PhySSCode2FP00 | 0x23 | Integer | 0~65535 | This attribute specify the SS Code 2 pair code 0 |
| PhySSCode2FP01 | 0x24 | Integer | 0~65535 | This attribute specify the SS Code 2 pair code 1 |
| PhySSCode3FP00 | 0x25 | Integer | 0~65535 | This attribute specify the SS Code 3 pair code 0 |
| PhySSCode3FP01 | 0x26 | Integer | 0~65535 | This attribute specify the SS Code 3 pair code 1 |
| PhySSCode4FP00 | 0x27 | Integer | 0~65535 | This attribute specify the SS Code 4 pair code 0 |
| PhySSCode4FP01 | 0x28 | Integer | 0~65535 | This attribute specify the SS Code 4 pair code 1 |
| ~~phyVTASCCValue~~ | ~~0x29~~ | ~~Unsigned~~ | ~~0~255~~ | ~~This attribute specify the no of Colors used in the VTASC~~ |
| ~~phyVTASCTxHSize~~ | ~~0x3A~~ | ~~Integer~~ | ~~0-65535~~ | ~~This attribute specify the no of Horizontal Pixel in the 2D Display Transmitter~~ |
| ~~phyVTASCTxVSize~~ | ~~0x3B~~ | ~~Integer~~ | ~~0-65535~~ | ~~This attribute specify the no of Vertical Pixel in the 2D Display Transmitter~~ |

Table 122~~179~~ — PHY PIB attributes (continued for VTASC)

# **9.5.2 PHY PIB Attributes for SS2DC**

The PHY PIB attributes for SS2DC is presented in the Table 122~~179~~ —PHY PIB attributes (continued for SS2DC).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Identifier** | **Type** | **Range** | **Description** |
| phySS2DCTxMode | 0x10 | Unsigned | 0-255 | This attribute indicates the Sequential Scalable 2D Code PHY transmission modes.  0 : SS2DC Mode  1 : SS SS2DC Mode |
| ~~PhySS2DCApplicationSpecificMode~~ | ~~0x10~~ | ~~Unsigned~~ | ~~0~255~~ | ~~This attribute specifies the application specific PHY mode.~~  ~~0 : Normal Data (Media Content, Information Content based on the Application used for)~~  ~~1 : ID Data~~  ~~2 : Authentication Data~~ |
| phySS2DCTxCamerEnable | 0xA2 | Unsigned | 0-255 | This attribute indicates the Transmitter is Enabled with Camera or not for Interactive Receiver distance specific data transfer control.  0 : Camera not connected  1 : Camera connected |
| phySS2DCRxDistance | 0xA3 | Unsigned | 0-255 | This attribute notify the Receiver distance from Transmitter |
| PhySS2DCCodeArea | 0x11 | Unsigned | 0~255 | This attribute specify the coded area of the IDE  0 : Full Screen  1 : Partial Screen  2~255 : Reserved |
| PhySS2DCCodeLocation | 0x12 | Unsigned | 0~255 | This attribute specify the Coded Location of the SS2DC  0 : Center  1 : Bottom Right  2 : Bottom Left  3 : Top Right  4 : Top Left  5~255 : Reserved |
| phySS2DCTHSize | 0x13 | Unsigned | 0~255 | This attribute specify the no of horizontal blocks in the SS2DC |
| phySS2DCTVSize | 0x14 | Unsigned | 0~255 | This attribute specify the no of vertical blocks in the SS2DC |
| PhySS2DCCODEHSIZE | 0x15 | Unsigned | 0~255 | This attribute specify the horizontal size of the 2D code in the SS2DC |
| PhySS2DCCODEVSIZE | 0x16 | Unsigned | 0~255 | This attribute specify the vertical size of the 2D code in the SS2DC |
| phySS2DCTFrequency | 0x17 | Unsigned | 0~255 | This attribute specify the frame rate of SS2DC sequence Transmission |
| PhySS2DCTxHSize | 0x18 | Integer | 0-65535 | This attribute specify the no of Horizontal Pixel in the 2D Display Transmitter |
| PhySS2DCTxVSize | 0x19 | Integer | 0-65535 | This attribute specify the no of Vertical Pixel in the 2D Display Transmitter |

Table 122~~179~~ — PHY PIB attributes (continued for SS2DC)

# **9.5.2 PHY PIB Attributes for IDE**

The PHY PIB attributes for IDE is presented in the Table 122~~179~~ —PHY PIB attributes (continued for IDE).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PHY PIB Table 100 Additions** | | | | |
| **Attribute** | **Identifier** | **Type** | **Range** | **Description** |
| phyIDETxMode | 0x10 | Unsigned | 0-255 | This attribute indicates the Invisible Data Embedding transmission modes.  0 : IDE-BLENDING  1 : IDE-WATERMARK  2 : SS IDE-BLEND  3 : SS IDE-WATERMARK |
| ~~phyIDEApplicationSpecificMode~~ | ~~0x11~~ | ~~Unsigned~~ | ~~0~255~~ | ~~This attribute specifies the application specific PHY mode.~~  ~~0 : Normal Data (Media Content, Information Content based on the Application used for)~~  ~~1 : ID Data~~  ~~2 : Authentication Data~~  ~~3~255: Reserved~~ |
| phyIDETxCameraEnable | 0x12 | Unsigned | 0-255 | This attribute indicates the Transmitter is Enabled with Camera or not for Interactive Receiver distance specific data transfer control.  0 : Camera not connected  1 : Camera connected |
| phyIDERxDistance | 0x13 | Unsigned | 0-255 | This attribute notify the Receiver distance from Transmitter |
| phyIDEModulation | 0x14 | Unsigned | 0~255 | This attribute specifies the modulation.  ~~0 : M-FSK~~  0 : HYBRID-MPFSK  1 : 2D Binary Code  2~255: Reserved |
| phyIDENoFrequency | 0x15 | Unsigned | 0~255 | This attribute specifies the number of frequency used in M-FSK and Hybrid-MPFSK |
| phyIDENoPhase | 0x16 | Unsigned | 0~255 | This attribute specifies the number of phase used in Hybrid-MPFSK |
| phyIDEFreqBase | 0x15 | Unsigned | 0~255 | This attribute specifies the base frequency used in M-FSK and Hybrid-MPFSK |
| phyIDEFreqSeparation | 0x16 | Unsigned | 0~255 | This attribute specifies the frequency difference used in M-FSK and Hybrid-MPFSK |
| phyIDEPhaseBase | 0x15 | Unsigned | 0~255 | This attribute specifies the base Phase used in Hybrid-MPFSK |
| phyIDEPhaseSeparation | 0x16 | Unsigned | 0~255 | This attribute specifies the Phase difference used in Hybrid-MPFSK |
| phyIDECodedArea | 0x17 | Unsigned | 0~255 | This attribute specify the coded area of the IDE  0 : Full Screen  1 : Partial Screen  2~255 : Reserved |
| phyIDECodedLocation | 0x18 | Unsigned | 0~255 | This attribute specify the Coded Location of the IDE  0 : Center  1 : Bottom Right  2 : Bottom Left  3 : Top Right  4 : Top Left  5~255 : Reserved |
| phyIDEHSize | 0x19 | Integer | 0-65535 | This attribute specify the no of horizontal pixel in the display |
| phyIDEVSize | 0x1A | Integer | 0-65535 | This attribute specify the no of vertical Pixel in the display |
| phyIDEENCHozAreaSize | 0x1B | Integer | 0-65535 | This attribute specify the no of horizontal pixel area to Encode |
| phyIDEENCVerAreaSize | 0x1C | Integer | 0-65535 | This attribute specify the no of horizontal pixel area to Encode |
| phyIDEMxNBlockSize | 0x1D | Unsigned | 0~255 | This attribute specify the no of Horizontal pixels in Blocks in the IDE  0 – 16x16 pixels  1 – 32x32 pixels  2 – 64x64 pixels  3~255: Reserved |
| phyIDEFrequency | 0x1E | Unsigned | 0~255 | This attribute specify the frame rate of IDE sequence Transmission |
| PhyIDETxHSize | 0x1F | Integer | 0-65535 | This attribute specify the no of Horizontal Pixel in the 2D Display Transmitter |
| PhyIDETxVSize | 0x20 | Integer | 0-65535 | This attribute specify the no of Vertical Pixel in the 2D Display Transmitter |

Table 122~~179~~ - PHY PIB attributes (continued for IDE)

# **Clause 13**

**13.5 Offset-VPWM**

The Offset-VPWM supported data rates and operating conditions are shown in PHY IV operating modes Table 81.

**13.5.1 Reference Architecture**

A reference implementation of the Offset-VPWM modulator is shown in Figure OffsetVPWMNEW.

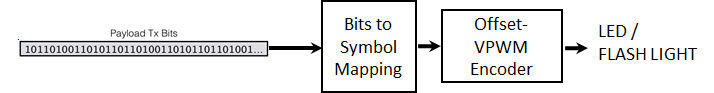


Figure OffsetVPWM – Offset-VPWM Block Diagram

**13.5.2 OFFSET-VPWM PHY Encoder**

~~The Offset-VPWM PHY supported data rates and operating conditions is shown in Table 81 – PHY IV, V, and VI Operating Modes for Offset Variable Pulse Width Modulation.~~

~~The Offset-VPWM characteristics are,~~

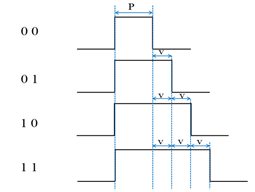
* ~~Modulation methods includes line coding~~
* ~~Defining the sum (P + nV) of the unit to be added to the minimum pulse (P) which is a reference pulse width (V) as a Symbol ( P > V, V > time error (jitter) )~~
* ~~Can specify a 2bit data symbol, 4bit data symbol according to number of added pulse~~
* ~~Data is expressed with offset pulse width, 2bits data (for example) were mapped into 4 Offset-VPWM symbols~~

The offset-VPWM encoder is built-in with line coding and encoded with thesum (P + nV)of the unit to be added to the minimum pulse (P) which is a reference pulse width (V) as a Symbol ( P > V, V > time error (jitter) ). The Offset-VPWM specify a 2bit data symbol, 4bit data symbol according to number of added pulse.

The data symbol map for two bits symbol with pulse width and respective symbol blinking waveform are shown in Table 137 and Figure 186 respectively.

|  |  |
| --- | --- |
| **Data Bits** | **Pulse Width** |
| 00 | P + 0V |
| 01 | P + 1V |
| 10 | P + 2V |
| 11 | P + 3V |

**Table 137 – Two Bits Symbol Mapping Truth Table**

****

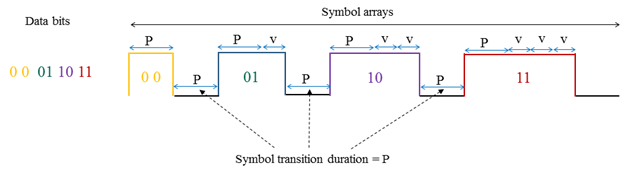
**Figure 186 – Two Bit Symbol Data Diagram**

In offset-VPWM, the data is expressed with offset pulse width, 4bits data (for example) were mapped into 16 Offset-VPWM symbols. The 4 bits symbol mapping truth table is shown in Table 138.

|  |  |
| --- | --- |
| **Data Bits** | **Pulse Width** |
| 0000 | P+0V |
| 0001 | P+1V |
| 0010 | P+2V |
| 0011 | P+3V |
| 0100 | P+4V |
| 0101 | P+5V |
| 0110 | P+6V |
| 0111 | P+7V |
| 1000 | P+8V |
| 1001 | P+9V |
| 1010 | P+10V |
| 1011 | P+11V |
| 1100 | P+12V |
| 1101 | P+13V |
| 1110 | P+14V |
| 1111 | P+15V |

**Table 138 – Four Bits Symbol Mapping Truth Table**

The symbol arrays mapping is described in waveform pattern as shown in Figure 187.



**Figure 187 – Symbol Array Mapping Timing Diagram**

The optimum value of P is 40 msec and V is vary from 10~50 % of P. The default V is 50% of P. The P and V are configurable over PHY attributes phyOffsetVPWMStdPERIOD, phyOffsetVPWMOffsetPERIOD respectively.

# **Clause 15**

# **15. PHY VI Specifications**

# **15.2. VTASC Specifications**

The VTASC works with variable transparency levels, sizes, shapes, and colors of the symbols~~patterns~~. The VTASC PHY supported data rates and operating conditions are shown in PHY VI operating modes Table 81.

# **15.2.1 VTASC Reference Architecture**

The reference PHY architecture for VTASC is illustrated in Figure 215. The data sequence including SHR (first byte), PSDU data packet length (second and third ~~First two~~ bytes used for PSDU data length), and PSDU encoded on the screen symbol. So the PSDU data length and PSDU shall be feed into the VTASC encoder. The data embedded on visual frame by overlaying VTASC symbols ~~visual patterns~~ in defined ~~displays~~ visual area. After spread spectrum, data is transformed into VTASC coded ~~patterns~~ symbols according to the mapping rule on the transparency levels, sizes, shapes, and colors by the VTASC coding ~~pattern blocks~~ symbols.

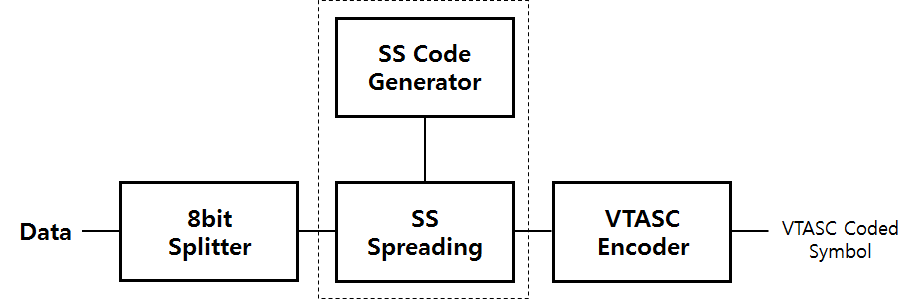
****

Figure 215 – Reference architecture for VTASC PHY System

The spread spectrum used with VTASC to have effective asynchronous, distance adaptive scalable data rate controlled OWC. The VTASC is used for enhanced display to camera communication in the real-time application usage scenario. The VTASC specific working features are given in Annex I.3.1.

The receiver specific information for VTASC Data Decoder is given in Annex J.9.

**15.2.2 Synchronization Sequence ~~Spread Spectrum~~**

The spread spectrum code used as a synchronization sequence. The spread spectrum used with VTASC, SS2DC, and IDE based display to camera OWC to have effective asynchronous, distance adaptive scalable data rate controlled communication. The display to camera communication adopted the binary zero-correlation duration (ZCD) code sequences as an optical spread code with the spreading code length. The initial basic matrix G used to generate binary ZCD is defined as,

1 1 1 -1

1 1 -1 1

1 -1 1 1

-1 1 1 1

G =

The binary ZCD sequences constructed cyclically from the chip-shift operation using family of codes {SN(a),SN(b)} shown in (xyz). Any row of G or –G is denoted as S4(a) = (S0(a), S1(a), S2(a), S3(a)), S4(b) = (S0(b), S1(b), S2(b), S3(b)) is generated from S4(a), where Sq(b) = Sq(a) (q = 0, 1, 2, 3).

{SN(a),SN(b),TΔ[SN(a)],TΔ[SN(b)], T2Δ[SN(a)],T2Δ[SN(b)] ,…,T(k-1)Δ[SN(a)],T(k-1)Δ[SN(b)], TkΔ[SN(a)],TkΔ[SN(b)] } ----------------------------------------------------- (xyz)

Where,

* SN(a),SN(b) are the pair of family sequence and N is family size
* Tl is chip shift operator, which shifts a sequence cyclically to the left by l chips
* Δ is a chip-shift increment and k is a the maximum number of chips-shifts for a sequence and Δ and k should satisfy |(k+1) Δ| ≤ |N/4 + 1| ,Δ is a positive and k a non-negative integer

The binary ZCD based optical spreading code used for a specific data rate or distance transmission is defined in Table 147.

Table 147 – Optical Spreading Code for different data rate or receiver distance

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Spread Sequence** | **Spreading Code** | | | | | | | | **Distance (meters)** | **Frame Refresh Rate (Hz)** |
| SC1#00 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | 30 |
| SC1#01 | 1 | -1 | 1 | 1 | -1 | 1 | 1 | 1 |
| SC2#00 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 | 2 | 30 |
| SC2#01 | 1 | 1 | 1 | -1 | 1 | 1 | -1 | 1 |
| SC3#00 | -1 | -1 | -1 | 1 | 1 | 1 | -1 | 1 | 3 | 30 |
| SC3#01 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 |
| SC4#00 | -1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | above 4 | 30 |
| SC4#01 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 |

There are four set of code used for scalable and distance adaptive transmission (see in 15.2.6) and each set of code coupled with pair of codes for synchronization (see in 15.2.5). The data spreading with spreading factor 1 is illustrated in Figure 216.

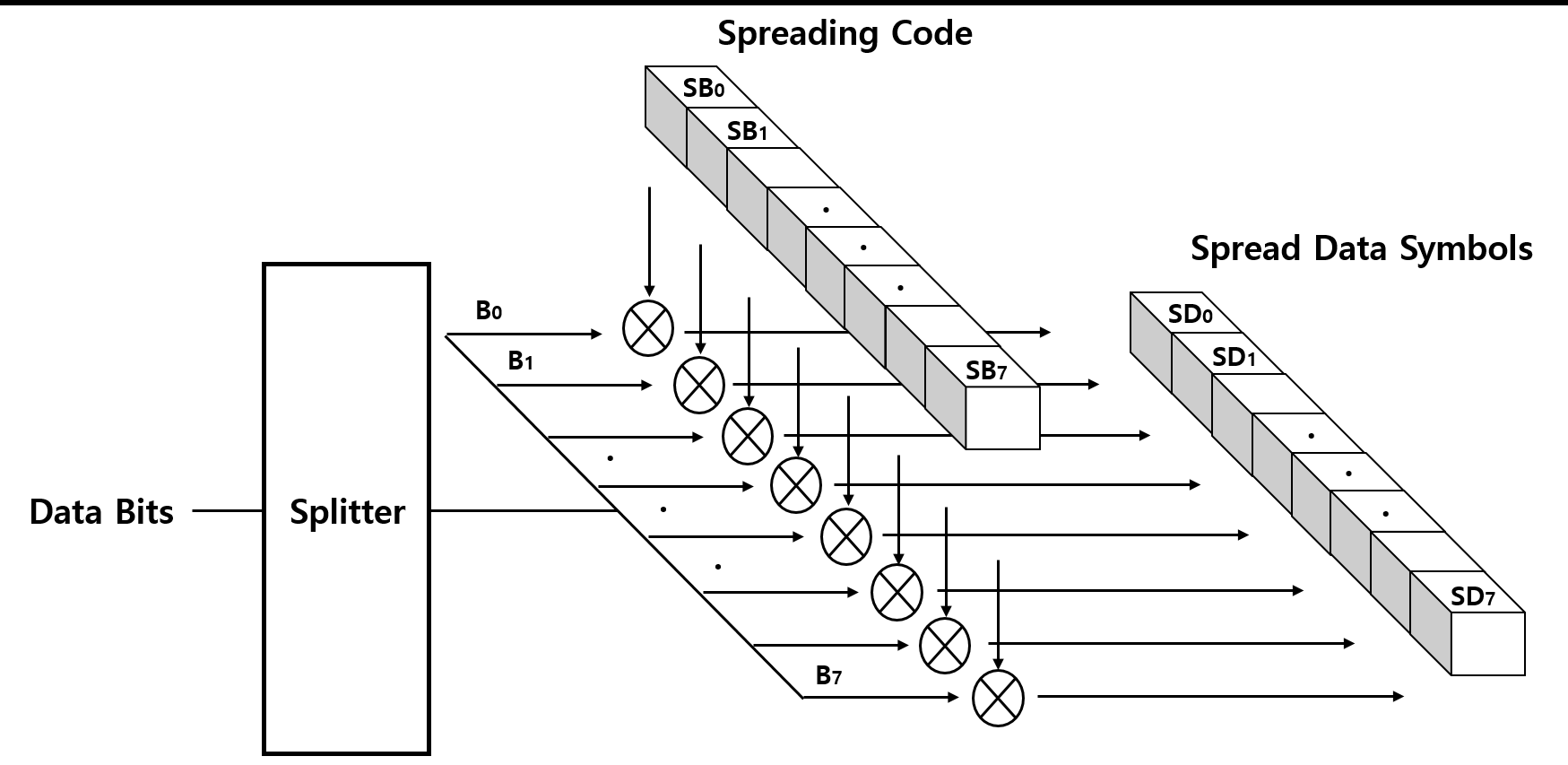


Figure 216 - SS Spreading Example

The SS code length configurable over PHY PIB attributes PhySSCode1Len, PhySSCode2Len, PhySSCode3Len, PhySSCode4Len and SS code pair is configurable over the PHY PIB attributesPhySSCode1FP00,PhySSCode1FP01,PhySSCode2FP00,PhySSCode2FP01,PhySSCode3FP00,PhySSCode3FP01,PhySSCode4FP00,PhySSCode4FP01.

**15.2.3 VTASC Code Design**

VTASC is a modulation scheme for visible-light communication involving single or multiple ~~screens~~ display (Panel, LED etc) or light bulbs with variable transparency levels, sizes, shape models, and colors. ~~VTASC enhances the OWC system performance with improved OWC throughput by increasing the bit per symbol rate, and avoiding the single color interference.~~

The VTASC is encoded by T (Transparency level) / A (Amplitude nothing but block size) / S (Shapes) / C (Colors) State as described in the Figure 217.

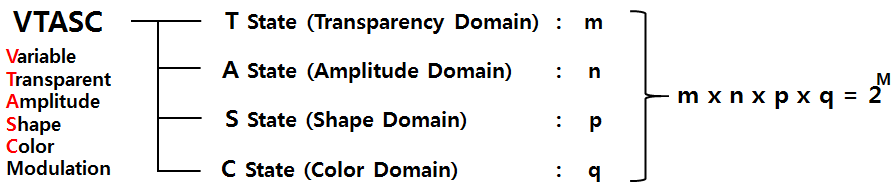


Figure 217 - VTASC Code ~~Design~~ States

The number of code levels in the VTASC modulation is (m x n x p x q) with two transparency levels, four block sizes, four shape models, and eight colors is 256 = 28 and this makes place to code 8 bit symbol with two levels of transparency, four size of blocks, four models of shape, and eight colors. The shape model design shown in Figure NEW1.

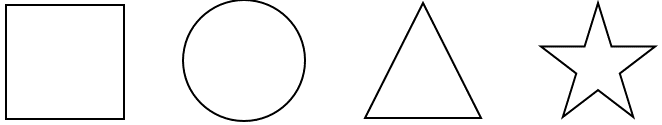
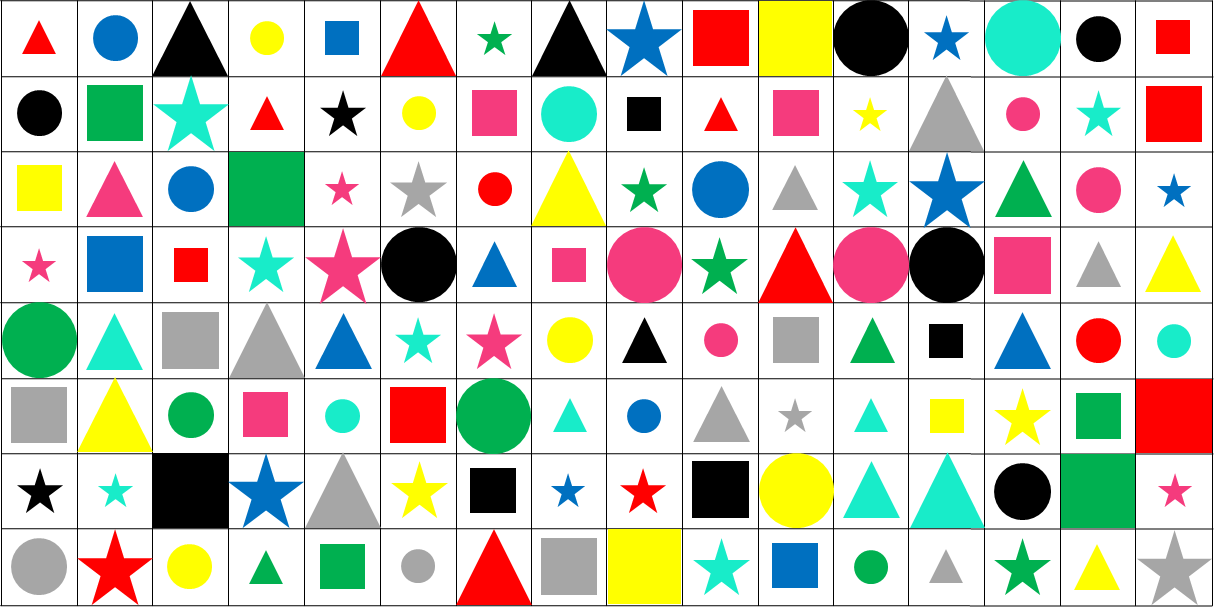
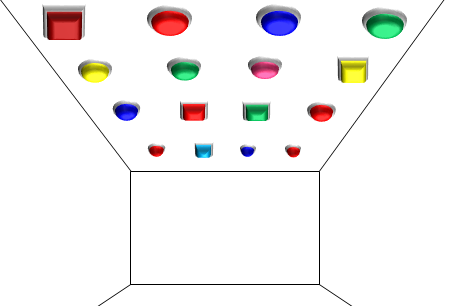


Figure NEW1 – VTASC Shape Models Design

~~In the VTASC coded screen symbol,~~ The shapes inside the ~~block~~ symbols pixel region is equally spaced in the VTASC symbols coding region. The coded symbols are ordered sequentially row by row same order as English text order and the coded region background color used shall be white. The zero padded VTASC coded symbols generated if the available number of data bits is less than the symbol mapping in the defined coding region. The VTASC code ~~example model~~ illustration is given Figure 218.

(a) Display (Panel, LED, etc.) Source (b) Light Bulb Source

Figure 218 - VTASC Code Symbols ~~Pattern Example~~ Illustration

The coding states are configurable over PHY PIB attributes phyVTASCTLevel, phyVTASCALevel, phyVTASCSLevel, and phyVTASCCLevel .Table 148 describes the bits per symbol for VTASC code design.

Table 148 - Bits per symbol for VTASC code~~d block models~~

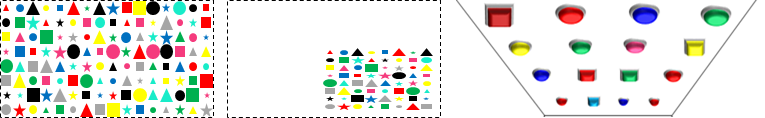
|  |  |  |
| --- | --- | --- |
| **Coding (m,n,p,q) States** | **Number of Coded Symbols (m\*n\*p\*q)** | **Bits per symbol** |
| m = 2, n = 4,p = 4, q = 2 | 64 = 26 | 6 |
| m = 2, n = 4,p = 4, q = 4 | 128 = 27 | 7 |
| m = 2, n = 4,p = 4, q = 8 | 256 = 28 | 8 |

Table 149 describes the data bits to coding states mapping for VTASC code design.

Table 149 - VTASC code~~d~~ symbol bit mapping with coding states (m, n, p, q)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits Per Symbol** | **Data Bits** | | | | | | | |
| **B7** | **B6** | **B5** | **B4** | **B3** | **B2** | **B1** | **B0** |
| 6 | - | - | m | n | n | p | p | q |
| 7 | - | m | n | n | p | p | q | q |
| 8 | m | n | n | p | p | q | q | q |

The number of horizontal and vertical symbols ~~blocks~~ depends on the partial or full ~~screen~~ display coded mode by PHY PIB attributes phyVTASCCodedArea. The example VTASC coded symbol of the full & partial display and Light Bulb coded mode is shown in Figure NEW2 (a) , NEW2 (b) , and NEW2 (C) respectively.



1. Full Display Coded Mode (b) Partial Display Coded Mode (c) Light Bulb Coded Mode

Figure NEW2 – VTASC Coded Modes

In partial ~~screen~~ display mode, the number of horizontal and vertical blocks configurable over the PHY PIB attributes phyVTASCAHSize, phyVTASCAVSize. In full display ~~screen~~ mode, the number of horizontal and vertical blocks estimated based on the display ~~screen~~ size, resolution, aspect ratio, and the relative pixel ratio in reference with 42 inches full HD (1920 pixels of width and 1080 pixels of height with 16:9 aspect ratio) display.

The size of the block is vary with display ~~screen~~ size and aspect ratio. For an example the block size of 21 inches full HD display looks compared to reference 42 inches full HD display. To generate block size same as a reference 42 inches full HD display, need to calculate the pixel ratio according to display specifications so that all display transmitter can generate same block size as reference display (For an example [32x32] pixel block is transformed into [32\*Pixelratio x 32\*Pixelratio] pixel block ). The pixel ratio calculation formula is,

Pixelratio =

Where,

* hNewResolutiion is horizontal resolution of display pixel to be estimated
* hRefResolutiion is horizontal resolution of reference display
* InchesRef is inches of display pixel to be estimated
* InchesNew in inches of reference display
* AspectRatioNew is aspect ratio of the used display and aspect ratio is expressed width by height
* AspectRatioRef is aspect ratio of reference display

**15.2.4 VTASC Encoder**

The display light based transmitter with VTASC encoder works by overlaying the data mapped color code on visual scene as show in Figure 219.

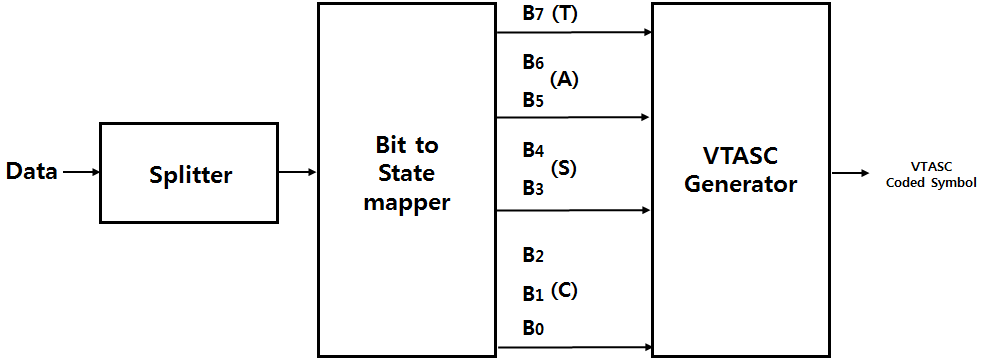


Figure 219 – VTASC Encoder

The data coded on display by overlaying visual symbols ~~patterns~~ in ~~displays~~ visual area. The overlaying visual symbols means that updating coded region pixel value according to VTASC coded symbol on display frame buffer to refresh on display ~~screen~~. The overlaying coded symbol on frame buffer and data rate achievement vary based on the kind of display used to design the transmitter and the distance between transmitter and receiver. Table 150 describes the example data rate supported by VTASC code design with symbol ~~block~~ size of 32x32 pixels on 42 inches full HD display with 16:9 aspect radio.

Table 150 – VTASC Data Rate Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modulation**  **(mxnxpxq)** | **RLL Code** | **Optical Clock Rate** | **FEC** | **Data Rate (Kbps)** |
| 2 Color VTASC Code  (m = 2,n = 4,p = 4,q = 2) | None | 30Hz | RS(64,32)/ RS(160,128)/ None | 384 Kbps  (FEC None) |
| 4 Color VTASC Code  (m = 2,n = 4,p = 4,q = 4) | None | 30Hz | RS(64,32)/ RS(160,128)/ None | 448 Kbps  (FEC None) |
| 8 Color VTASC Code  (m = 2,n = 4,p = 4,q = 8) | None | 30Hz | RS(64,32)/ RS(160,128)/ None | 512 Kbps  (FEC None) |
| 2 Color SS VTASC Code  (m = 2,n = 4,p = 4,q = 2) | None | 30Hz | None | 192 Kbps1 |
| 4 Color SS VTASC Code  (m = 2,n = 4,p = 4,q = 4) | None | 30Hz | None | 224 Kbps1 |
| 8 Color SS VTASC Code  (m = 2,n = 4,p = 4,q = 8) | None | 30Hz | None | 256 Kbps1 |

1 where spreading factor is 2

The data rate calculation for display based transmitter is described below,

DataRate = (NoofSymbols \* BitsPerSymbol \* OpticalClockrate \* FECRate) / CodeLength)

Where,

* CodeLength is 1 for without SS spreading and respective spreading code factor used for with SS spreading
* DisplayWidth = 1920; DisplayHeight = 1080;
* SymbolWidth = 32; SymbolHeight = 32 ;
* NoOfHorizontalSymbols = (DisplayWidth / SymbolWidth) = 60 (Approx. to even for coding efficiency)
* NoOfVerticalSymbols = (DisplayHeight / SymbolHeight) = 32 (Approx. to even for coding efficiency)
* NoofSymbols = (NoofHorizontalSymbols\* NoofVerticalSymbols)
* BitsPerSymbol = 7 (Refer Table 206)
* OpticalClockrate = 30 Hz
* FECRate = 1 (Refer Table 206)

The Data Rate for 2 Color VTASC Code with 4 amplitude ~~8 size~~ scalability & 4 shapes & 2 transparency Level without SS spreading code (CodeLength is 1)

DataRate = (NoOfHorizontalSymbols \* NoOfVerticalSymbols \* BitsPerSymbol \* OpticalClockrate \* 1) / 1) = 403200 = 390 Kbps (Approx.)

VTASC uses the two transparency levels in code design. The transparency defines the pixel with an observed color when given the pixel and a background VTASC coded ~~block~~ symbol color. The symbol to bitmapping for transparency is shown in Table 151.

Table 151 – Symbol to bit mapping for transparency level

|  |  |
| --- | --- |
| **Symbol Bit (B7)** | **Transparency Level (%)** |
| 1 | 100 |
| 0 | 50 |

VTASC ~~block~~ symbol size represented by amplitude state in code design. The symbol to bitmapping for symbol ~~block~~ size is shown in Table 152.

Table 152 – Symbol to Bit Mapping for Amplitude Level

|  |  |
| --- | --- |
| **Symbol Bits (B6,B5)** | **Block Size (MxN Pixels)** |
| 00 | 128x128 |
| 01 | 96x96 |
| 10 | 64x64 |
| 11 | 32x32 |

VTASC uses the four shape model in code design. The four shape models are shown in Figure NEW1. The symbol to bitmapping for shape is shown in Table 153.

Table 153 – Symbol to bit mapping for shape model

|  |  |
| --- | --- |
| **Symbol Bits (B4,B3)** | **Shapes** |
| 00 | Square |
| 01 | Circle |
| 10 | Triangle |
| 11 | Star |

VTASC uses the eight color in code design. The symbol to bitmapping for color channel is shown in Table 154. The coded region background color is white.

Table 154 – Symbol to bit mapping for color channel

|  |  |
| --- | --- |
| **Symbol Bits (B2,B1, B0)** | **Color Channel** |
| 000 | Black |
| 001 | Red |
| 010 | Green |
| 011 | Blue |
| 100 | Yellow |
| 101 | Magenta |
| 110 | Cyan |
| 111 | Gray |

**15.2.5 Asynchronous Communication**

Transmitter does not use any reference block for receiver synchronization with transmitter. The spreading codes used to support receiver to perform asynchronous data decoding irrespective of receiver frame rate variation. To provide efficient receiver synchronization, every frame in the video sequence spreading with one spreading code and the alternative frames use the spreading code pairs sequentially. The spreading sequence order in the video frame sequence is shown in Figure 220.

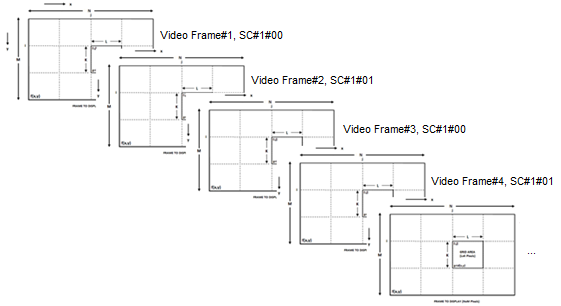


Figure 220 – Video Frame Sequence SS Code Assignment

The receiver decoding for asynchronous communication and receiver error mitigation due to rolling effect is given in Annex J.9.1.

**15.2.6 Scalable Bitrate Controller**

The VTASC and SS2DC PHY for display based OWC designed with built-in scalable bitrate controller. There are two types of scalable bitrate controller supported,

* Receiver framerate adaptive multirate controller
* Receiver distance adaptive data rate controller

The scalable bitrate control mode selection configurable over the PHY PIB phyVTASCScalRateCtrl.

**15.2.6.1 Receiver frame rate adaptive multirate controller**

The screen is divided into 2x2 regions and each region encode with different optical rate and renders the visual scene on screen. The different optical rate encoded region is spreaded by pair of spread code as defined in Table 147. The same encoded symbol~~pattern~~ rendered repeatedly at the rate of (displayRefreshRate / OpticalClockRate) to control the multirate data rate control on single screen. To achieve robust communication, the scalable multirate data transmission in PHY model design is shown in Figure 222. The region based optical clock rate and SS code con configurable over the PHY PIB attributes phyVTACScalRegion1OpticalClockRate to phyVTACScalRegion4OpticalClockRate, PhySSCode1FP00 to PhySSCode4FP01.

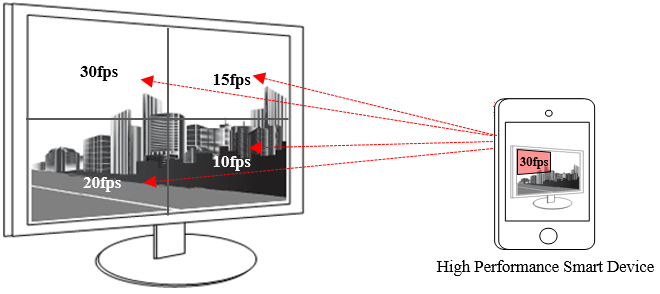
****

Figure 222 – Scalable Bitrate Controller

**15.2.6.2 Receiver distance adaptive data rate controller**

The receiver distance adaptive data rate control is by changing the ~~block pattern~~ symbol size small for short distance (example: 32x32 but vary with display) and big for long distance (example: 128x128 but vary with display). The receiver distance calculated using camera installed on transmitter. The camera availability configured by phyVTASCTxCameraEnable PHY attribute. The region based distance adaptation is configurable PHY PIB attributes phyVTACScalRegion1DistanceRange, phyVTACScalRegion2DistanceRange, phyVTACScalRegion3DistanceRange, and phyVTACScalRegion4DistanceRange.

The different distance range encoding is spreaded by pair of spread code as defined in Table 147. In this case the transmitter built-in with camera features as shown in Figure 223 to estimate the receivers distance using camera. The receiver distance estimation is not part of this standard. The distance based optical clock rate and SS code configurable over the PHY PIB attributes phyVTACScalRegion1OpticalClockRate to phyVTACScalRegion4OpticalClockRate, PhyVTACSSCode1FP00 to PhyVTACSSCode4FP01.

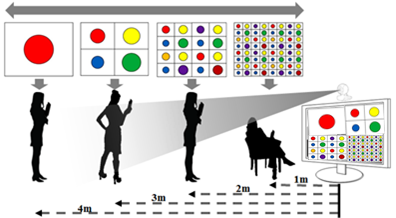
****

Figure 223 – Distance Adaptive Data rate Control

**15.3 SS2DC Specification**

The Sequential Scalable 2D Code (SS2DC) works with different 2D codes organized in a combination of one or more codes sequentially in row and column manner. The SS2DC uses QR, VTASC, A-QL, HA-QL, and IDE 2D codes. The SS2DC PHY supported data rates and operating conditions are shown in PHY VI operating modes Table 81.

# **15.3.1 SS2DC Reference Architecture**

The reference PHY architecture for SS2DC is illustrated in Figure 224. The data embedded on visual frame by overlaying 2D coded symbols ~~patterns~~ in defined displays visual area. After spread spectrum, data is transformed into SS2C coded 2D symbols ~~patterns~~ according to the 2D Code encoder.

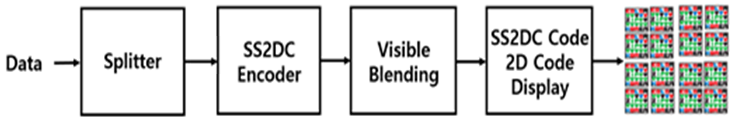
****

Figure 224 – Reference architecture for SS2DC PHY System

~~The spread spectrum used with SS2DC to have effective asynchronous and communication, distance adaptive scalable data rate controlled OWC.~~ The SS2DC is used for enhanced display to camera communication in the real-time application usage scenario. The SS2DC specific working features are given in Annex I.3.2.

The receiver specific information for SS2DC Data Decoder is given in Annex J.10.

**~~15.3.2 Spread Spectrum~~**

~~The spread spectrum used with SS2DC coded display based transmitter to add built-in adaptation on data recovery in addition to achieve the asynchronous communication with angle free and distance adaptive communication between transmitter and receiver. The subclause 15.2.2 contains more information about spread spectrum.~~

**15.3.2 SS2DC Code Design**

SS2DC is a two dimensional design using different 2D codes for improved OWC throughput by sequentially arranging the coded screen symbols in a 2D order as shown in Figure 225. The data is encoded as per 2D code principle and displayed on the display screen or panels. The number of horizontal and vertical 2D code blocks is configurable over the PHY PIB phyVTASCAHSize, phyVTASCAVSize. The horizontal and vertical size of the 2D code is configurable over the PHY PIB PhySS2DCCODEHSIZE and PhySS2DCCODEVSIZE.



(a) 2D QR Code (b) 2D Color Code (c) Combination of 2D Codes

Figure 225 – SS2DC Code Design Examples

**15.3.3 SS2DC Encoder**

The display light based transmitter with SS2DC encoder works by overlaying the data mapped color code on visual scene as show in Figure 226.

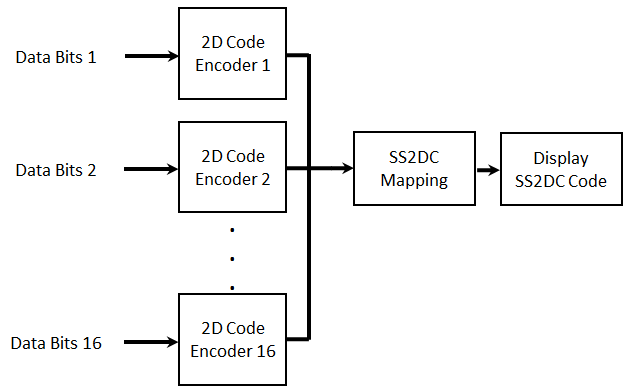


Figure 226: Sequential Scalable 2D Code Data Encoder

The data coded on display by overlaying visual symbols~~patterns~~ in displays visual area. The overlaying visual sysmbol means that updating coded region pixel value according to SS2DC coded symbol on display frame buffer to refresh on display screen. The overlaying coded symbol on frame buffer and data rate achievement vary based on the kind of display used to design the transmitter and the distance between transmitter and receiver. SS2DC uses one or more 2D codes from QR, VTASC, A-QL, HA-QL, and IDE 2D codes.

The QR based data encoder uses QR code version 40 and follows the ISO/IEC 18004 standard. The A-QL based data encoding informations a described in subclause 15.1. The VTASC based data encoding informations a described in subclause 15.2. The IDE based data encoding informations a described in subclause 15.4. The VTASC based data encoding informations a described in subclause 15.6. The minimum QR code size must be equal to (scanning distance / 10) to have an effective QR detection.

Table 155 describes the data rate supported by SS2DC code design with QR code. The Table 155 data rates differs according to the 2D code specification.

Table 155 – SS2DC Data Rate Table Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modulation** | **RLL Code** | **Optical Clock Rate** | **FEC** | **Data Rate (Kbps)** |
| 2x2 SS2DC | None | 2DCodeDecodingRate | RS(64,32)/ RS(160,128)/None | 92 Kbps  (FEC None) |
| 4x4 SS2DC | None | 2DCodeDecodingRate | RS(64,32)/ RS(160,128)/None | 368 Kbps  (FEC None) |
| ~~2x2 SS2DC~~ | ~~None~~ | ~~2DCodeDecodingRate~~ | ~~None~~ | ~~46 Kbps~~~~1~~ |
| ~~4x4 SS2DC~~ | ~~None~~ | ~~2DCodeDecodingRate~~ | ~~None~~ | ~~184 Kbps~~~~1~~ |

1 ~~where spreading factor is 2~~

The data rate calculation is described below,

DataRate = NoOfCodeSequence\* (2DCodeDataCapacity \* OpticalClockrate \* FECRate) ~~/ CodeLength~~)

Where,

* ~~CodeLength is spreading code factor used for SS spreading (1 for without SS)~~
* NoofHorizontalBlocks is no of horizontal 2D code sequence
* NoofVerticalBlocks is no of vertical 2D code sequence
* NoOfCodeSequence = (NoofHorizontalBlocks\* NoofVerticalBlocks)

The data rate for 2x2 SS2DC ~~without SS spreading~~ with QR code (The maximum data capacity is 2953 bytes) and DataRate = 4\* (2953 \* 8)\* 1 \* 1) / 1) = 94494 = 92 Kbps (Approx.).

~~The SS2DC PHY transmitter supports asynchronous communication by spreading data stream with SS code. The subclause 15.2.5 contains more information about asynchronous communication. The SS2DC PHY Transmitter with SS2DC designed with built-in Scalable bitrate Controller. The subclause 15.2.6 contains more information about scalable bitrate controller.~~

**15.4 IDE Specification**

The Invisible Data Embedding (IDE) works on embedding data on visual frame in unobtrusive mode using blending and watermarking. The IDE PHY supported data rates and operating conditions are shown in PHY VI operating modes Table 81.

# **15.4.1 IDE Reference Architecture**

The reference PHY architecture for IDE is illustrated in Figure 298. The data sequence including PSDU data packet length (First two bytes used for PSDU data length), and PSDU encoded on the screen symbol. So the PSDU data length and PSDU shall be feed into the IDE encoder. The data embedded on visual frame by invisible image blending and watermarking in the defined displays visual area. After spread spectrum, data is transformed into IDE encoding according to the invisible blending and watermarking rules as described in 227.

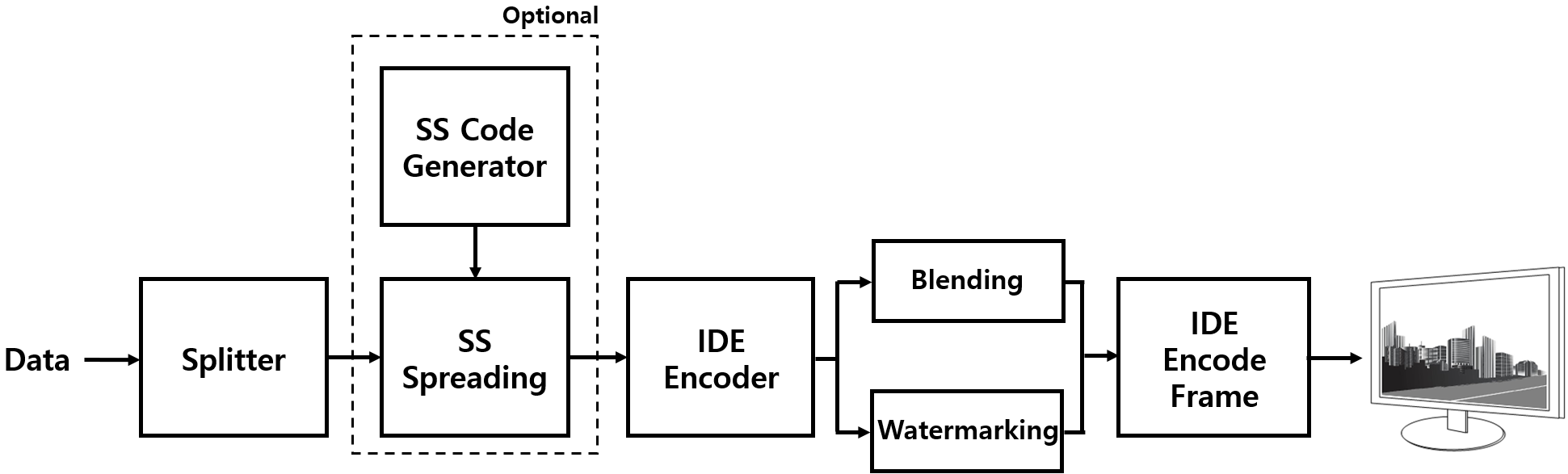
****

Figure 227 – Reference architecture for IDE PHY System

The spread spectrum used with IDE to have effective asynchronous and communication, distance adaptive scalable data rate controlled OWC. The IDE is used for enhanced display to camera communication in the real-time application usage scenario. The IDE specific working features are given in Annex I.3.3.

The receiver specific information for IDE data decoder is given in Annex J.11.

**15.4.2 Spread Spectrum**

The spread spectrum used with IDE coded display based transmitter to add built-in adaptation on data recovery in addition to achieve the asynchronous communication with angle free and distance adaptive communication between transmitter and receiver. The subclause 15.2.2 contains more information about spread spectrum.

**15.4.3 IDE Encoder**

IDE is a two dimensional block based imperceptible data encoder for unobtrusive OWC communication between screen and camera. The visual display frame is divided into MxN pixel blocks and the human imperceptible data encoded on visual scene block using M-FSK/PSK by image blending and 2D-Binary code by image watermarking. The display light based transmitter with IDE encoder works by invisible overlaying the data mapped on visual scene as show in Figure 228. The ~~M-FSK,~~ Hybrid-MPFSK, 2D Binary Code, Blending, and Watermarking are described in 15.4.3.1, 15.4.3.2, 15.4.3.3, 15.4.3.4, 15.4.3.5 respectively.

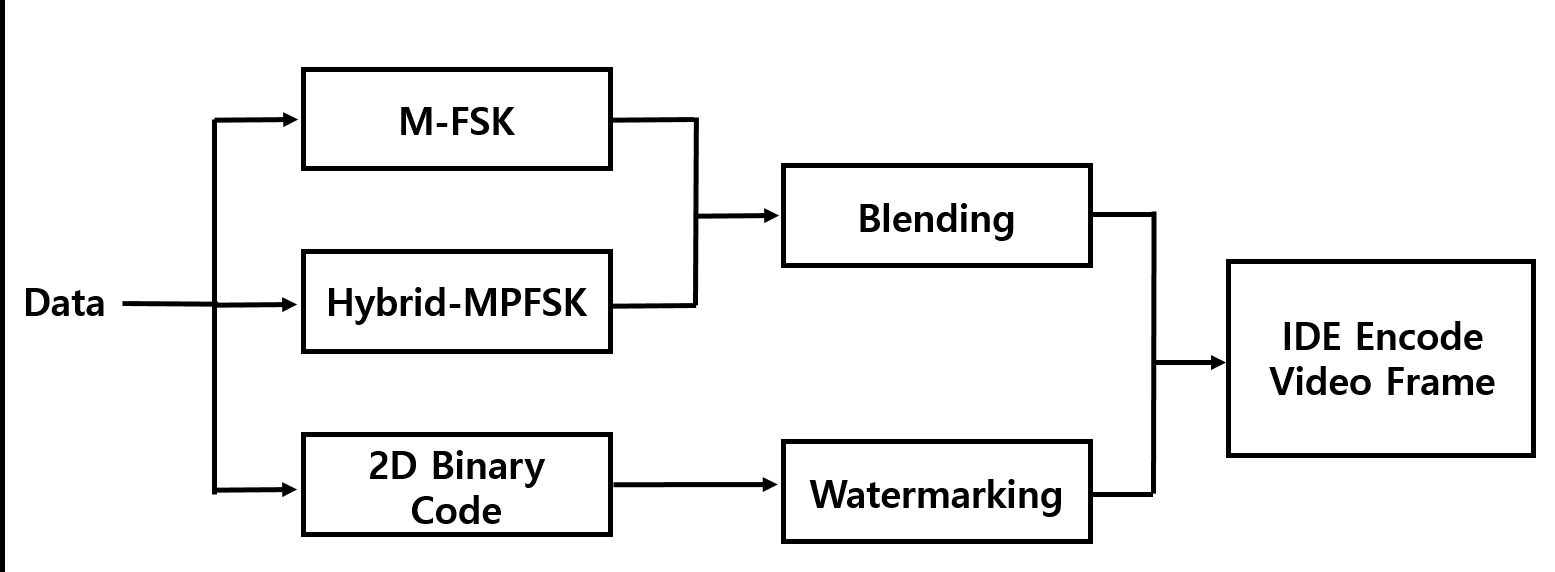


Figure 228 – IDE Data Encoder

The IDE transmitter mode, modulation are configurable over the PHY PIB phyIDETxMode, phyIDEModulation. The 16x16, 32x32, and 64x64 are three blocks size is used and configurable over the PHY PIB phyIDEMxNBlockSize.

Table 156 describes the data rate supported modulation schemes. The IDE encoder data rate estimated with reference to full HD display. The aspect ratio, size (inches) does not impact the distance does change the scale of the block design.

Table 156 – IDE PHY Data Rate Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modulation** | **RLL Code** | **Optical Clock Rate** | **FEC** | **Data Rate (Kbps)** |
| ~~M-FSK-BLEND~~ | ~~None~~ | ~~30Hz~~ | ~~RS(64,32)/ RS(160,128)/None~~ | ~~16 Kbps~~  ~~(FEC None)~~ |
| HYBRID-PFSK-BLEND | None | 30Hz | RS(64,32)/ RS(160,128)/None | 32 Kbps  (FEC None) |
| 2DBIN-WM | None | 30Hz | RS(64,32)/ RS(160,128)/None | 128 Kbps  (FEC None) |
| ~~SS-M-FSK-BLEND~~ | ~~None~~ | ~~30Hz~~ | ~~None~~ | ~~8 Kbps~~~~1~~ |
| SS-HYBRID-PFSK-BLEND | None | 30Hz | None | 16 Kbps1 |
| SS- 2DBIN-WM | None | 30Hz | None | 64 Kbps1 |

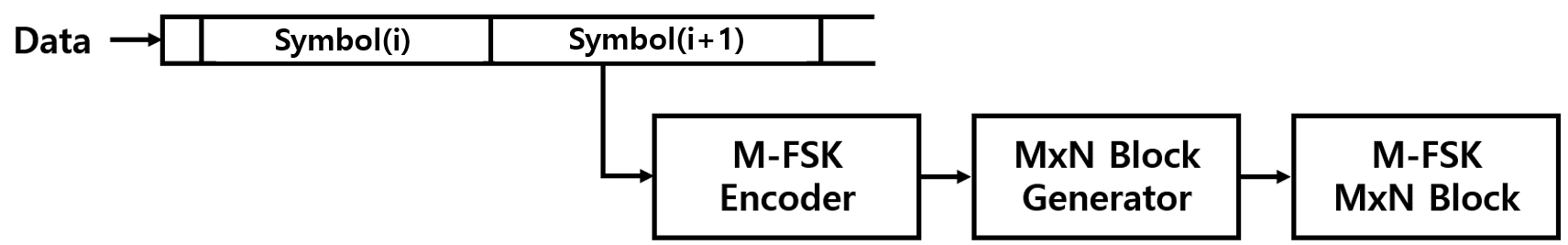
1 where spreading factor is 2

The IDE PHY transmitter supports asynchronous communication by spreading data stream using SS code. The subclause 15.2.4 contains more information about asynchronous communication.

**~~15.4.3.1 M-FSK Modulation~~**

~~The M-FSK uses the 16 frequency ranges to map data symbol. The SS spreaded data bits sequence is splited as a 4 bit symbol and map into a selected 16 frequencies as shown in Figure 229. The number of frequencies used to map data shall be configured over the PHY PIB attribute phyIDEFSKNoFrequency.~~

~~The M-FSK encoder generate the sine waveform for the symbol mapped frequency according to Table 157 and store the sine waveform in an MxN 2D pattern. The M-FSK encoded symbol 2D pattern of MxN pixel block is blended with visual scene to be rendered on screen. The data blending is described in 15.4.3.4.~~

~~~~

~~Figure 229 – IDE M-FSK Data Encoder~~

~~The 4 bit symbol to bitmapping for M-FSK is shown in Table 157. The symbol coded sine waveform patterns are generated dynamically based on PHY PIB attributes and stored in lookup table (LUT).~~

~~Table 157 – Symbol to bit mapping for M-FSK~~

|  |  |
| --- | --- |
| **~~Symbol Bits (B3, B2,B1, B0)~~** | **~~Frequency Mapping~~** |
| ~~0000~~ | ~~f~~~~0~~ |
| ~~0001~~ | ~~f~~~~1~~ |
| ~~0010~~ | ~~f~~~~2~~ |
| ~~0011~~ | ~~f~~~~3~~~~­~~ |
| ~~….~~ | ~~…~~ |
| ~~1101~~ | ~~f~~~~13~~ |
| ~~1110~~ | ~~f~~~~14~~ |
| ~~1111~~ | ~~f~~~~15~~ |

~~The frequencies f~~~~0~~~~~f~~~~15~~ ~~are selected to encode data and the relationship between frequencies are,~~

~~f~~~~i~~ ~~= f~~~~0~~ ~~+ i.df (I = 1.2……..15) , Where df is the selected frequency separation value.~~

~~The selection of all frequencies shall be configured over the first frequency (f~~~~0~~~~) which shall be implemented over the PHY PIB attribute phyIDEFreqBase (200Hz by default) and the frequency separation which shall be implemented over the PHY PIB attribute phyIDEFreqSeparation (50Hz).~~

**15.4.3.2 Hybrid-MPFSK Modulation**

Hybrid scheme used to achieve double the data rate ~~of M-PSK or F-FSK~~ by combining frequency and phase on the modulation. The Hybrid-MPFSK uses the 16 frequency and two phase ranges to map data symbol. The data bits spreaded with SS sequence is splited as a 5 bit symbol and map into a selected 16 frequencies conjunction with two phases as shown in Figure 230. The number of frequencies and phase used to map data shall be configured over the PHY PIB attribute phyIDEFSKNoFrequency, and phyIDEPSKNoPhase.

The Hybrid-MPFSK encoder generate the sine waveform for the symbol mapped frequency and phase according to Table 158 and store the sine waveform in an MxN 2D pattern. The Hybrid-MPFSK encoded symbol generates 2D pattern of MxN pixel block to blend with visual scene to be rendered on screen. The data blending is described in 15.4.3.4.

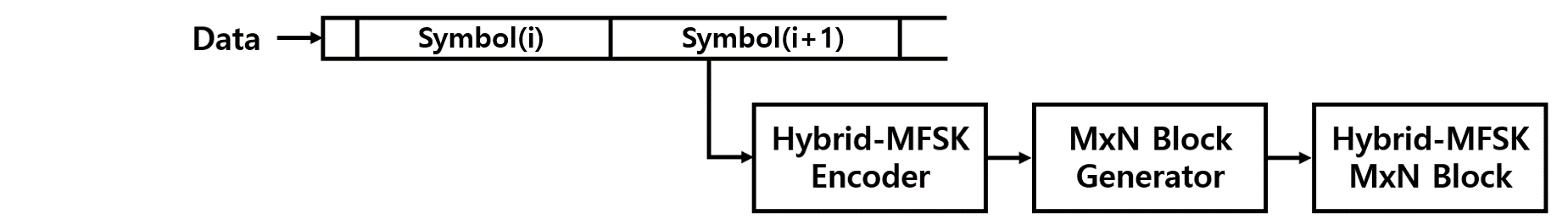


Figure 230– IDE Hybrid-MPFSK Data Encoder

The 5 bit symbol to bitmapping for Hybrid-MPFSK is shown in Table 158. The symbol coded patterns are generated dynamically based on PHY PIB attributes and stored in lookup table (LUT).

Table 158 – Symbol to bit mapping for Hybrid-MPFSK

|  |  |  |
| --- | --- | --- |
| **Symbol Bits (B4, B3 B2,B1, B0)** | **Phase Mapping** | **Frequency Mapping** |
| 00000 | P0 | f0 |
| 00001 | P0 | f1 |
| … | … | … |
| 01111 | P0­ | f15­ |
| 10000 | P1 | f0 |
| 10001 | P­1 | f1 |
| … | P1 | … |
| 11111 | P1 | f15 |

The frequency selection and configuration are described in 15.4.3.1.

The phase P0~P1 are selected to encode data and the relationship between phases are,

Pi = P0 + i.dP (I = 1.2), Where dP is the selected phase separation value.

The selection of all phases shall be configured over the first phase (P0) which shall be implemented over the PHY PIB attribute phyIDEPhaseBase, (0 by default) and the phase separation which shall be implemented over the PHY PIB attribute phyIDEPhaseSeparation (180).

**15.4.3.3 2D Binary Code**

The horizontal and vertical encoding area pixel ranges shall be configured over the PHY PIB attribute phyIDEENCHozAreaSize, phyIDEENCVerAreaSize. The 2D binary encoder, first calculate the number of MxN pixels blocks in visual display by dividing phyIDEENCHozAreaSize and phyIDEENCVerAreaSize by 8. The MxN number of encoding blocks sized SS spreaded data is extracted and converted to 2D format of MxN dimension. The described 2D Binary Data encoder is shown in Figure 231.

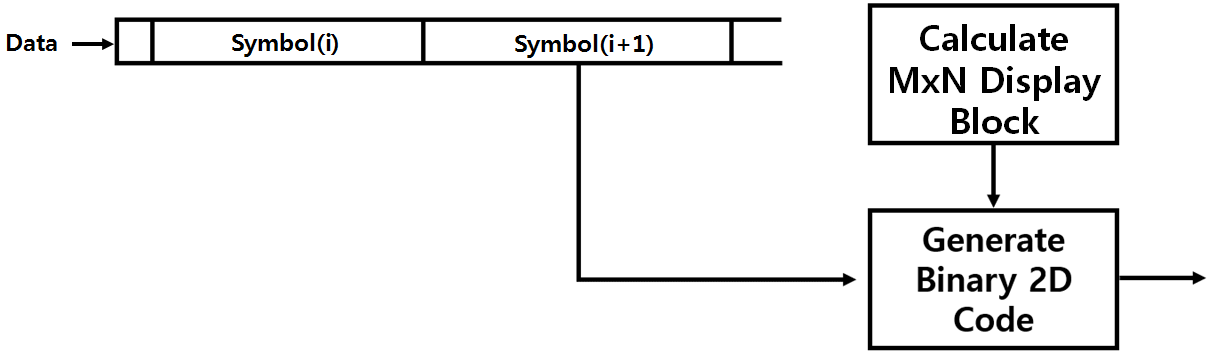


Figure 231 – IDE 2D Binary Data Encoder

**15.4.3.4 Invisible Data Blending**

The invisible data blending is the process of overlaying a foreground image with transparency over a visual rendering frame .The ~~M-FSK or~~ Hybrid-MPFSK coded MxN block is blended with in visual frame sequential in every block ~~by cell row to cell column manner~~ and rendered on display screen ~~as shown in~~ as shown in Figure 232.

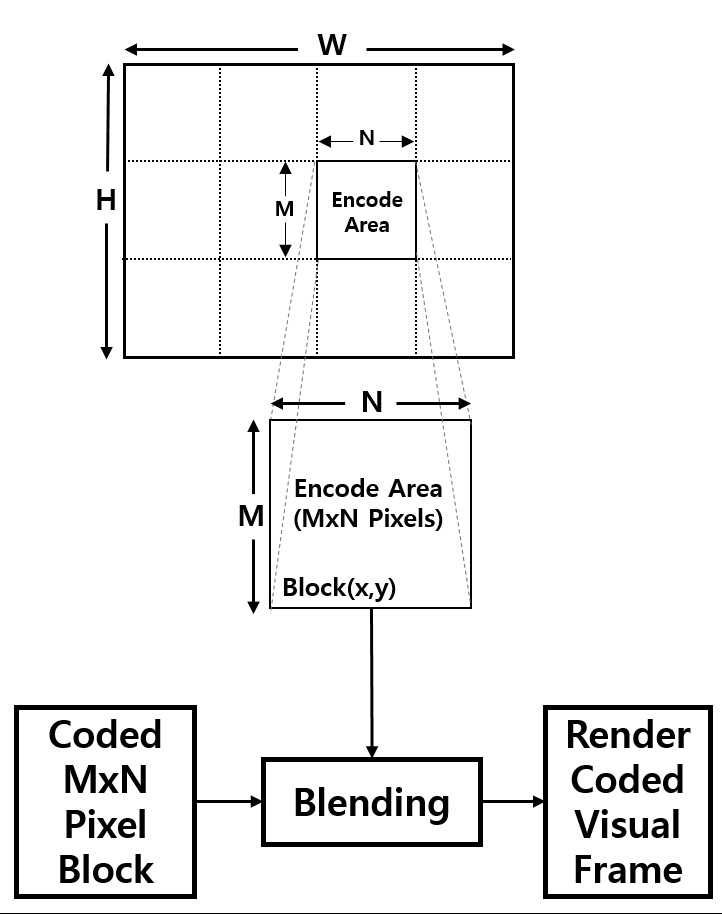


Figure 232 – IDE Encoder Blending

The visual blending rule is,

IDEEncodedFRAME = α. VisualFrameBlock(x, y) + (1-α) M-PFSKCodeBlock.

Where

* α is blending factor ~~α is~~ and α = 0.0~0.3 for invisible blending
* x is current row of MxN block in visual frame
* y is current column of MxN block in visual frame

**15.4.3.5 Invisible Watermarking**

The human eye is more sensitive to lower frequency components than to higher-frequency components. This means that most of the important information in an image is contained in the lower-frequency components. So, the higher frequency components can be discarded without visible degrading the image. The invisible watermarking utilize the human eye visual imperceptibility in middle-high frequency component of every 8x8 block of the visual frame.

The binary 2D coded MxN block is watermarked imperceptibly with in visual frame of every 8x8 block by row to column manner and rendered on display screen as shown in as shown in Figure 233.

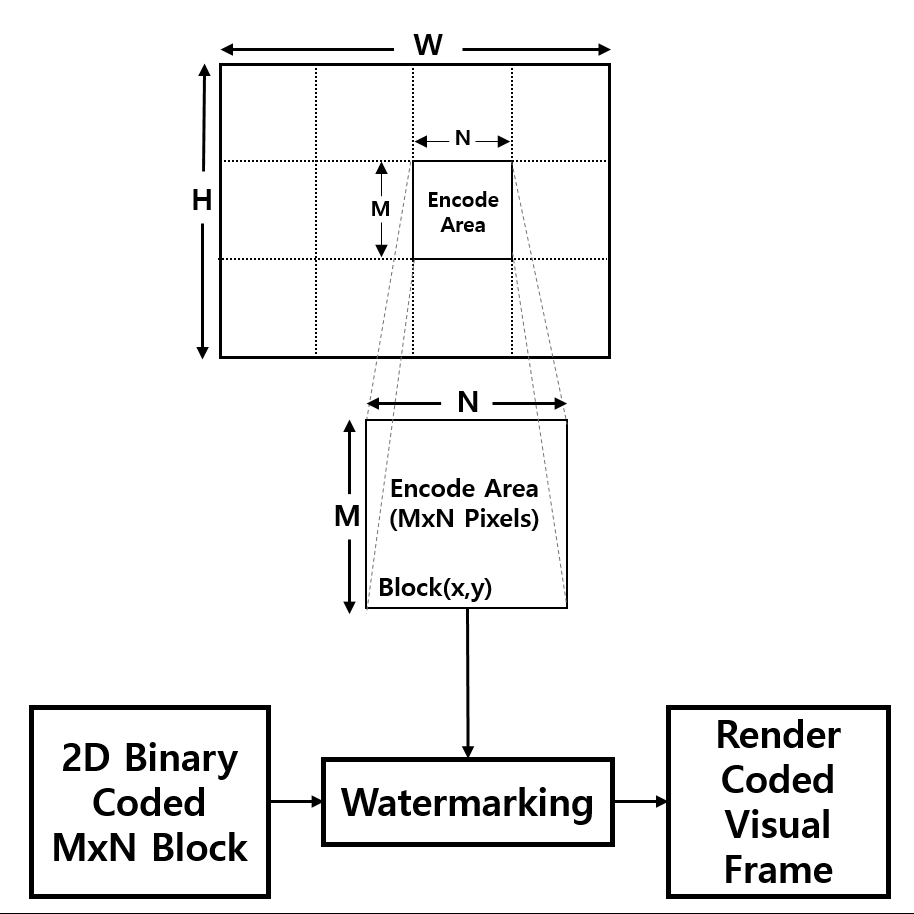


Figure 233 – IDE Encoder - Watermarking

This frequency based Transform domain watermarking is dominant on for invisible data embedding using the discrete cosine transform (DCT). The detailed frequency based invisible data embedding procedure using DCT is described in Annex ~~M~~L.

# **~~Annex G~~ Annex J**

**Annex J (Informative)**

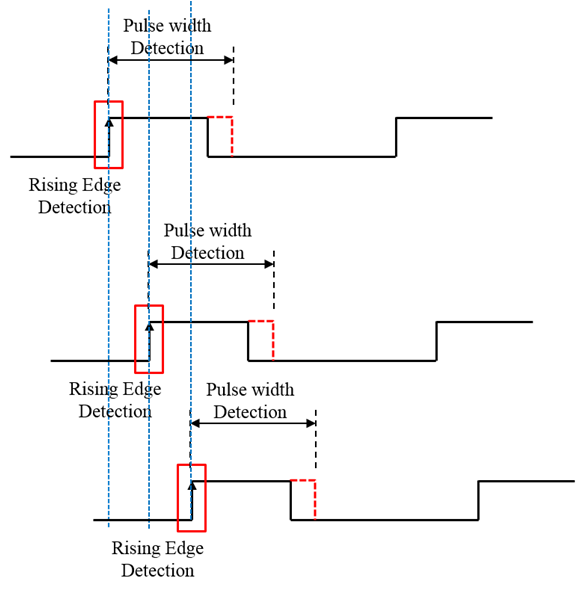
**~~OFFSET-VPWM Receiver Processing Guide~~**

**J.8 ~~Receiver Detection Method~~ Offset-VPWM Decoding Method**

The receiver camera that has frame rate same as the optical clock rate shall be used to receive data i.e. 25 fps. To decode the data symbol, the smart device capture the visual frame using rolling-shutter mode. The offset-VPWM symbol is decoded with three consecutive frame and receiver can synchronize rising edge and check pulse width length for each symbols decoding.

The pulse width duration is calculated using PHY attributes phyOffsetVPWMStdPERIOD, phyOffsetVPWMOffsetPERIOD. The total pulse width duration is (phyOffsetVPWMStdPERIOD + phyOffsetVPWMOffsetPERIOD +/- 5% of phyOffsetVPWMStdPERIOD).

The receiver detection process in the wave formatted approach is show in Figure J.12.



**Figure J.12 – Receiver Detection Process**

# **~~Annex K~~**

**~~Annex K (Informative)~~**

**J.9 VTASC Decoder Method**

VTASC data decoding is shown Figure J.13.

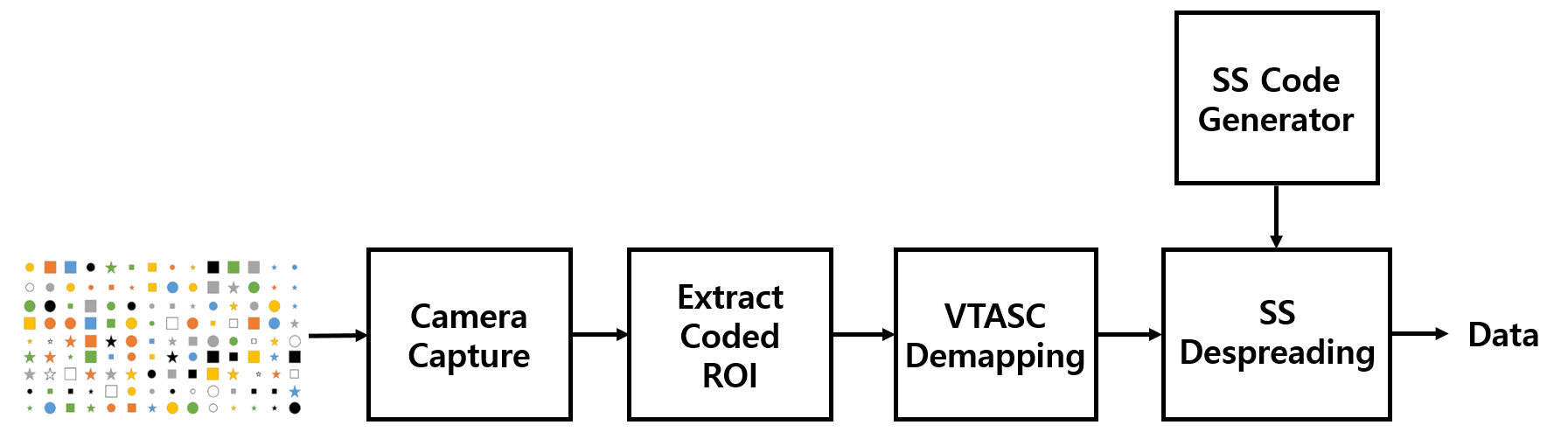


Figure J.13 – VTASC Receiver Functional Block Diagram

The ROI of screen visual area is extracted from the captured visual frame and then detect the VTASC coded symbols ~~patterns~~ based on mapping scheme applied on the transmitter. The data recovered by applying SS despreading on the VTASC decoded data streams. The receiver frame rate required for optimum decoding is twice the optical clock rate of the transmitter.

**J.9.1 Asynchronous Communication**

The optical clock rate and SS codes are configurable over the PHY PIB phyVTACScalRegion1OpticalClockRate to phyVTACScalRegion4OpticalClockRate, PhySSCode1FP00 to PhySSCode4FP01.The receiver synchronized using SS code (any one of the four pair SS code at first time) and decoded the data.

If camera captured same frame receive twice in a sequence, then receiver will discard the second video frames when SS detection fails with next code of pair of SS code that means SS detection is true with previous frames SS code. If the current frame decoding detects the both SS codes in a single frame, then that frame is rolling effect fault capture frame and must be discarded.

**J.9.2 Angle Free Communication**

The angle free communication between transmitter and receiver is shown in Figure J.14.

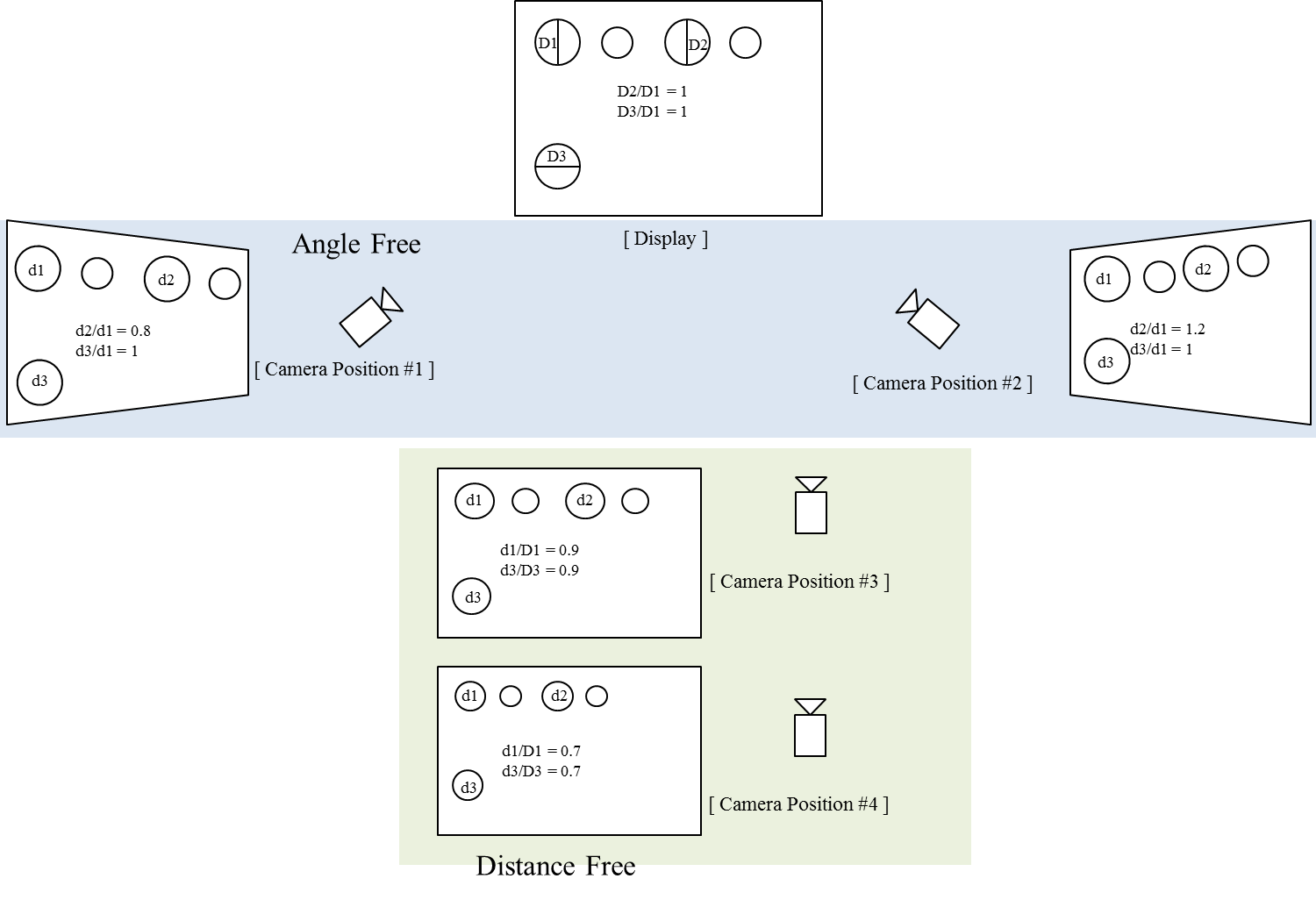


Figure J.14 – Angle free and distance adaptive communication

The angle free communication is achieved by warping the ROI of the transmitter to get the original shape alignment and then the decoded data synchronizing with spread code to extract original information transmitted. ~~transferred on transmitter.~~ ~~The kind automatic synchronization in receiver is time consuming function but the communication is robust.~~

~~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.~~

~~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.~~

In case a rolling shutter camera captures in between the transition time of two adjacent symbols of data, the SS code on the captured image shall not be the same. The image is identified as a rolling affected image and be discarded.

Also, to support a rotated camera decoding, the decoder rotates the captured frame and apply the decoding procedure if the captured image fails on SS detection of SHR symbol.

**J.10 SS2DC Decoder Method**

SS2DC data decoding is shown Figure J.15.

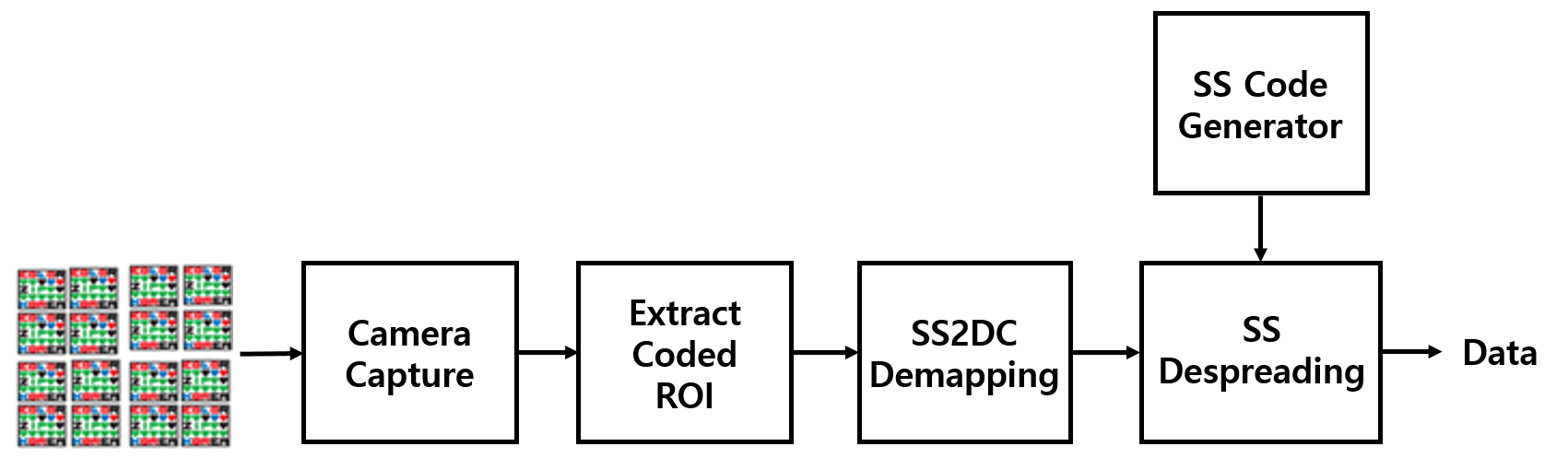


Figure J.15 – SS2DC Receiver Functional Block Diagram

The ROI of Screen Visual Area is extracted from the captured visual frame and then apply the Sequential Scalable 2D Code detector based on mapping scheme applied on the transmitter. The data recovered by applying SS despreading on the SS2DC data decoded. The receiver frame rate required for optimum decoding is twice the optical clock rate of the transmitter.

**J.11 IDE Decoder Method**

IDE data decoding is shown Figure J.16.

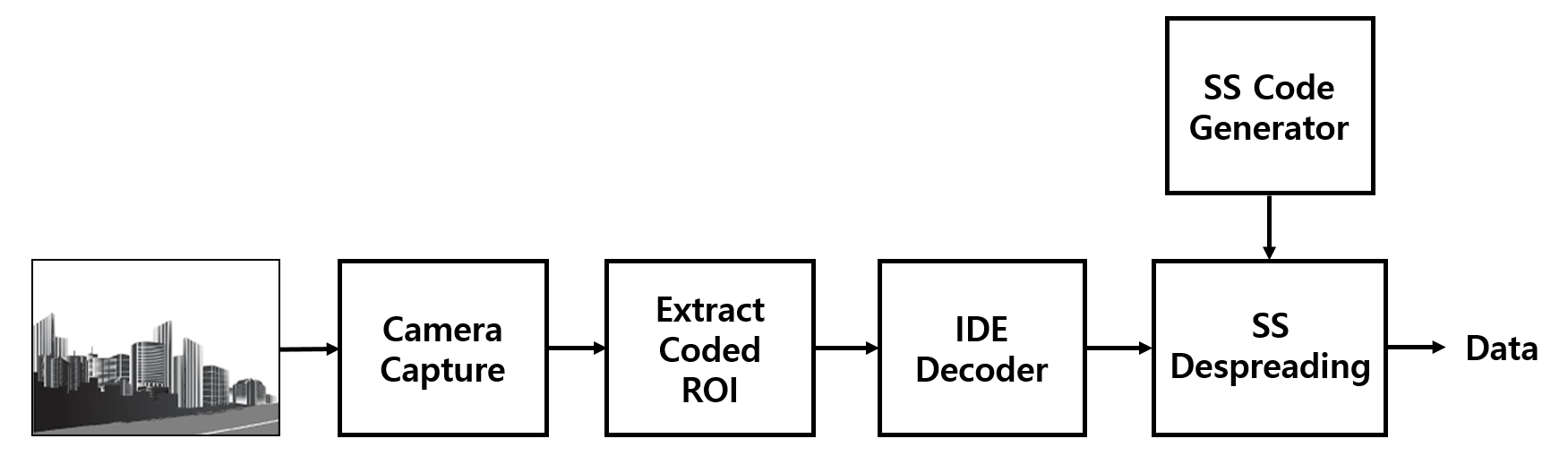


Figure J.16 – IDE Receiver Functional Block Diagram

To decode the data stream, the ROI of display visual area is extracted from the captured visual frame using image processing methods (like canny edge detection , Gaussian blurring, Contour Extraction, Perspective Transform, and Warping) and then invisibly embedded data extracted using blending or watermark extraction procedure.

The blending or watermark based data extraction ~~procedure~~ is applied based on modulation ~~scheme~~ used to invisibly embedding the data on the transmitter ~~system (modulation scheme is described in 2.1).~~ The receiver frame rate required for optimum decoding is twice the optical clock rate of the transmitter.

The blending encoded the data ~~works~~ with combination of M-PSK and M-FSK. ~~The decoder demodulate the data from every MxN pixels block and.~~ The decoder applies the FFT on translucency changes to demodulate the data from MxN pixels block window on captured video frame. The decoder detects the coded frequency and phase to extract the encoded data symbols.

The data embedded using high frequency visual coefficients on visual frame is extracted by applying DCT on every 8x8 block and then extracted data by extracting high frequency coefficient values. The recovered high frequency coefficient based data is SS coded data so SS decoding is applied to recover original data from the visual sequence.

In case a rolling shutter camera captures in between the transition time of two adjacent symbols of data, the SS code on the captured image shall not be the same. The image is identified as a rolling affected image and be discarded.

Also, to support a rotated camera decoding, the decoder rotates the captured frame and apply the decoding procedure if the captured image fails on SS detection of SHR symbol.

# **~~Annex L~~ Annex I**

**Annex I (Informative)**

**~~PHY VI – VTASC, SS2DC, and IDE Specific Characteristics~~**

**I.3 PHY Mode Specific Characteristics**

**I.3.1 VTASC**

The VTASC works on,

* Receiver angle free and distance adaptive communication
* Receiver distance adaptive communication achieved by screen with interactive Camera
* Asynchronous and receiver frame rate independent communication
* Scalable bitrate controller for distance adaptive data rate control
* Enhanced multi-display model for transmission

**I.3.2 SS2DC**

The SS2DC works on,

* Receiver angle free and distance adaptive communication
* Receiver distance adaptive communication achieved by screen with interactive Camera
* Asynchronous and receiver frame rate independent communication
* Scalable bitrate controller for distance adaptive data rate control
* Enhanced multi-display model for transmission

**I.3.3 IDE**

The IDE works on,

* Unobtrusive to screen viewer on dynamic visual Scene
* Receiver angle free and distance adaptive communication
* Asynchronous and receiver frame rate independent communication
* Enhanced multi-display model for transmission

# **Annex M**

**Annex L (Normative)**

**~~Invisible Watermarking~~**

**L.1 Frequency Based Invisible Watermarking**

The human eyes are more sensitive to noise in lower frequency range than its higher frequency counterpart, while the energy of most natural images are concentrated on the lower frequency range so the reasonable trade-off is to embed the watermark into the middle-high frequency range of the image.

The discrete cosine transform (DCT) based watermarking method used for frequency based image watermarking which could survive several kinds of image processing. The frequency based invisible watermarking using DCT is described in principle by the block diagram in Figure M.1.

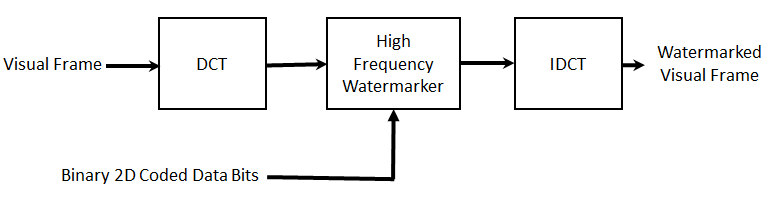


Figure M.1 – DCT based Watermarking Block Diagram

The visual rendering frame is divided into 8x8 blocks of pixels, and the 2-D DCT is applied independently to each block. Then, select the four coefficients of high -frequency range from the DCT coefficients for watermarking, an example of defining the high-frequency coefficients is shown in Figure M2.

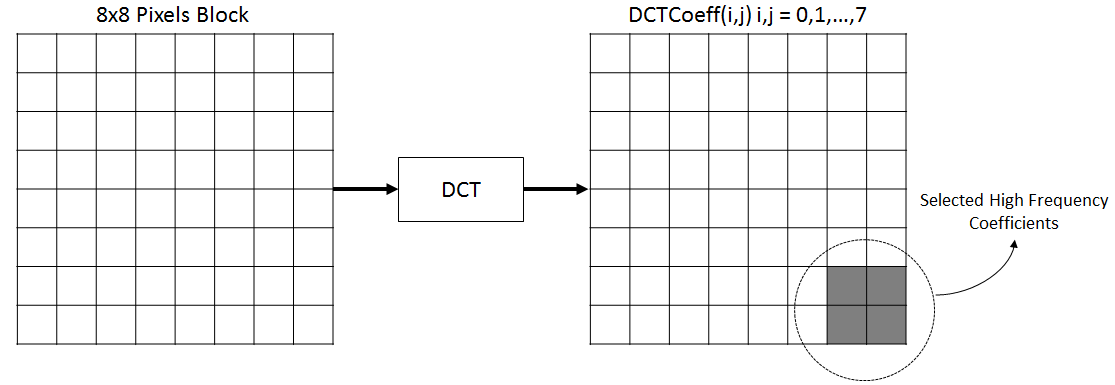


Figure M.2 – High-Frequency Coefficients Selection for Watermarking

The selected DCT coefficients for embedding data using watermarking are DCTCoeff (6, 6), DCTCoeff (6, 7), DCTCoeff (7, 6), and DCTCoeff (7, 7). The data is embedded on DCT coefficients as below,

WaterMARKDCTCoeff(i,j) =

The inverse discrete cosine transform (IDCT) applied on watermarked DCT coefficients to restore the visual frame on spatial domain to rendering on screen for OWC.