**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |  |
| --- | --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) | |
| Title | **Draft D1-PHY Layer Operating Modes of** **MIMO C-OOK scheme** | |
| Date Submitted | November, 2021 | |
| Source | Huy Nguyen, Yeong Min Jang | Voice: [ ] Fax: [ ] E-mail: [yjang@kookmin.ac.kr] |
| Re: | Draft D1 Comment Resolution for MIMO C-OOK scheme | |
| Abstract | Details of Resolutions regarding to the submitted Comments on D1 are suggested MIMO C-OOK PHY Layer Operating Modes and PHY Specifications. | |
| Purpose | D1 Comments 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 Layer Operating Modes**

The Multiple Input Multiple Output Camera On-Off Keying (MIMO C-OOK) Modulation for high-speed OCC system uses the PHY VII – Multiple Point Source.

The PHY VII Operating Modes system specifications are given in Table 151. The additional PHY Operating Modes by MIMO C-OOK for Smart Device is presented the Table 151 – PHY VII Operating Modes.

**Table 151 – PHY VII Operating Modes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PHY Operating Modes** | | | | |
| **Modulation** | **RLL Code** | **Optical Clock Rate** | **FEC** | **Data Rate**  **(two LEDs)**  **(60 fps)** |
| MIMO C-OOK | Manchester | 8 kHz | Hamming (15,11) | 4.8 kbps |
| 4B6B | 8 kHz | RS (15,11) | 7.2 kbps |
| Manchester | 10 kHz | Hamming (15,11) | 6.0 kbps |
| 4B6B | 10 kHz | RS (15,11) | 9.0 kbps |

# **MIMO C-OOK**

The PHY VII with supported data rates and operating conditions is shown in Table 151 – PHY VII Operating Modes for Multiple Input Multiple Output Camera On-Off Keying (MIMO C-OOK) Modulation with high-speed OCC system uses the PHY VII - Multiple Point Source.

The proposed Multiple Input Multiple Output Camera On-Off Keying (MIMO C-OOK) was designed with the following characteristics:

* Modulation methods include line coding (Manchester code and 4B6B)
* The Optical clock rate is at 8 kHz or 10 kHz
* Sequence Number part was put in the head and tail of packets
* Multiple light sources were applied to increase the data rate
  1. Reference architecture

A reference architecture to implement MIMO C-OOK is shown in Figure 217



Figure 217. MIMO C-OOK block diagram

* 1. MIMO C-OOK encoder

2.1.1. Encoder configuration

A packet of data is modulated using OOK modulation. The optical clock rate is at 8 kHz or 10 kHz. The optical clock rate at which MIMO C-OOK symbols are clocked out is configurable over PHY PIB attribute *phyMimoCookOpticalClockRate.*



Figure 218. MIMO C-OOK data packet structure

The data packet structure is as shown in figure 218. A packet consists of multiple similar data sub-packets to avoid missing data between adjacent images' gap time. The number of repetitions depends on the communication mode specified later. The configuration of preamble shall be implemented over the PHY PIB attribute *phyMimoCookPreambleSymbol*.

Table 152—Data sub-packet format

|  |  |  |  |
| --- | --- | --- | --- |
| **Preamble** | **Data sub-packet payload** | | |
| **Start SN data**  **(Sequence Number)** | **Payload** | **End SN data**  **(Sequence Number)** |
| 011100 | Manchester coding | | |
| 0011111000 | 4B6B coding | | |

2.2.2. RLL coding

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

2.2.3. Sequence Number inserting

The data sub-packet payload shall consist of three subparts: the Start SN data, payload, and the End SN Data. The Start SN Data and the End SN Data shall carry the same information, which consists of the asynchronous information. SN shall be implemented over the PHY PIB attribute *phyMimoCookSn*

2.2.4. Forward error correction (FEC)

The data sub-packet payload may be coded by FEC to protect the payload from error. Hamming (8,4) or Reed Solomon (15,11) code may be used as an FEC. FEC shall be configured via the PHY PIB attribute *phyMimoCookFEC*

Table xx shows PHY PIB attributes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Range** | **Description** |
| *phyMimoCookOpticalClockRate* | Integer | 0-7 | The optical clock rate (or symbol rate) applied for MIMO C-OOK. 0: 8 kHz 1: 10 kHz Others: Reserved |
| *phyMimoCookRLLCode* | Integer | 0-7 | This specifies the RLL coding for MIMO C-OOK modulation. 0: Manchester 1: 4B6B coding Other values: Reserved |
| *phyMimoCookFec* | Integer | 0-7 | This attribute specifies FEC for MIMO C-OOK modulation. 0: None 1: Hamming (8/4) 2: Hamming (15/11) 3: Hamming (8/4), Other values: Reserved |
| *phyMimoCookPreambleSymbol* | Integer | 0-3 | This attribute specifies the preamble symbol of PSDU of MIMO C-OOK. 0: 6B symbol (preamble = 011100) 1: 10B symbol (preamble = 0011111000) 2–3: Reserved |
| *phyMimoCookSn* | Integer | 0-3 | This attribute specifies the length of Sequence Number per packet of MIMO C-OOK  0: 2 bits  1: 3 bits  2-3: reserved |

**Annex**

* 1. MIMO C-OOK decoder



MIMO C-OOK decoder architecture

* *Oversampling:*

When the frame rate of a rolling shutter camera is at least two times larger than the packet rate of the transmitter, the data packet is sampled multiple times causing the oversampling effect. When the packet is sampled more than once, errors of packet merger are created at the receiver’s end. The SN is added to the DS to deal with this problem because it improves the receiver’s ability to decrease the effect of the frame rate variation of the camera. The redundant data will be removed in the receiver when the same SN value is recognized in the DS of different packets. The receiver will eliminate consecutive packets with the same SN and choose packets with consecutive SN values (n − 1, n, n + 1) for the merger.



Merge Forward and Backward parts in each image (One SF in each image)



Full payload in each image (Multiple SF in each image)



Merging packet of MIMO C-OOK scheme in multiple images

* *Undersampling:*

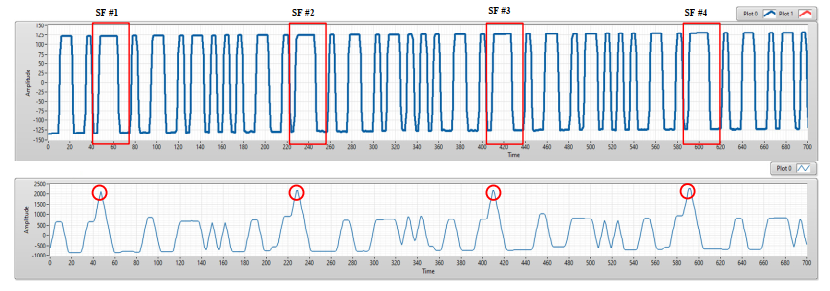
When the frame rate decreases below the packet rate of the transmitter, undersampling occurs. The payload will be lost in undersampling unlike in the case for oversampling. Figure below shows the scenario in which the missing payload is created and detected using the SN. In this case, the SN is long enough for the receiver to detect the missing payload. The SN of the data frame is increased depending on the sequences of the payload. If one payload is missing, the error can be detected by comparing the SN of the two adjacent DSs. The number of SNs in different states depends on the length of each sub-packet.



Error detection in grouping image during undersampling case

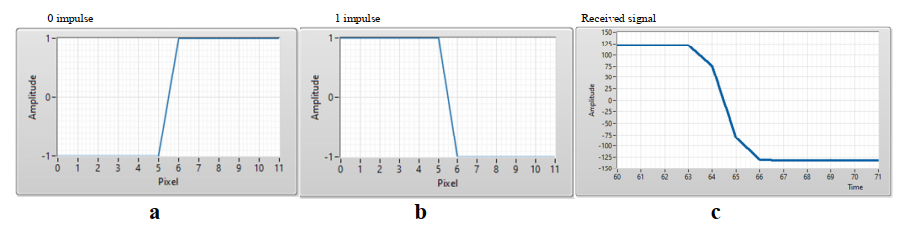
* *Matched filter*

The matched filter is a filter technology achieved by comparing a template signal with the real signal to determine the template signal in the real signal. The matched filter, which is one of the linear filter technologies, optimizes the SNR in the appearance of additive random noise. Matched filters are widely used in almost all wireless communication systems, such as mobile communications and radar systems to maximize the SNR of the system; these filters increase the system performance. To detect preamble positions, the received signal is multiplied with the known preamble signal via the convolution algorithm as Equation (1):



(a) An experimental result of COOK within a rolling image. (b) The results of preamble position detection based on matched filter

After the detect preamble, to decode data, we also used the matched filter to decode data to improve the system performance. By create known patterns as Figure below, the received signal is multiplied with the known preamble signal via the convolution algorithm. From convolution results, we can know that: which patterns are the most like the received signal? From that, it is easy to verify the value of signals (0 or 1). Same with the Manchester code, we can create 16 patterns of 4B6B code to decode data.



Manchester code signal patterns and COOK received signal. (**a**) 0 impulse, (**b**) 1 impulse, (**c**) COOK received signal.