



## Baumer HXC CameraLink®カメラレジスタ設定ガイド



# 目次 :

|                                   |           |
|-----------------------------------|-----------|
| <b>1. 基本情報</b> .....              | <b>5</b>  |
| <b>2. シリアルインターフェイス</b> .....      | <b>5</b>  |
| 2.1. シリアル通信のデフォルト設定 .....         | 5         |
| 2.2. シリアルコマンド .....               | 5         |
| 2.2.1. レジスタ読み出し .....             | 7         |
| 2.2.2. レジスタ読み出しの応答 .....          | 8         |
| 2.2.3. レジスタ書き込み .....             | 8         |
| 2.2.4. レジスタ書き込みの応答 .....          | 10        |
| 2.2.5. ステータス値 .....               | 10        |
| 2.2.6. コードサンプル .....              | 11        |
| <b>3. レジスタ内容</b> .....            | <b>16</b> |
| 3.1. デバイス情報 .....                 | 16        |
| 3.1.1. 製造会社名 .....                | 16        |
| 3.1.2. モデル名 .....                 | 17        |
| 3.1.3. Firmware / CID .....       | 18        |
| 3.1.4. シリアル番号 .....               | 18        |
| 3.1.5. ユーザー定義名 .....              | 19        |
| 3.2. センサーバージョン .....              | 20        |
| 3.3. ユーザーセット .....                | 20        |
| 3.3.1. デフォルトユーザーセット選択 .....       | 21        |
| 3.4. CameraLink® インターフェイス機能 ..... | 22        |
| 3.4.1. シリアル通信速度 .....             | 22        |
| 3.4.2. タップ制御 .....                | 23        |
| 3.4.3. センサー読み出し速度 (クロック) .....    | 24        |
| 3.5. 画像取得 .....                   | 24        |
| 3.5.1. 画像取得の停止 .....              | 24        |
| 3.5.2. 露光時間 .....                 | 24        |
| 3.5.3. PRNU/DSNU補正 .....          | 25        |
| 3.5.4. ハイダイナミックレンジ (HDR) .....    | 25        |
| 3.5.5. 読み出しモード .....              | 27        |
| 3.5.6. 固定フレームレート .....            | 27        |
| 3.5.7. パーシャルスキャン (ROI) .....      | 28        |
| 3.5.8. マルチROI .....               | 29        |
| 3.5.9. ビニングとサブサンプル .....          | 30        |
| 3.5.10. 画像反転 .....                | 31        |
| 3.5.11. センサーデジタイザータップ .....       | 31        |
| 3.5.12. テストパターン .....             | 31        |
| 3.6. ピクセルフォーマット .....             | 32        |
| 3.7. ゲインとオフセット .....              | 32        |
| 3.7.1. ゲイン .....                  | 32        |
| 3.7.2. オフセット .....                | 32        |
| 3.7.3. カラーゲイン .....               | 33        |
| 3.8. LUT .....                    | 34        |
| 3.8.1. 基本仕様 .....                 | 34        |
| 3.8.2. ガンマ補正 .....                | 35        |
| 3.9. 欠陥画素補正 .....                 | 35        |

|                            |    |
|----------------------------|----|
| 3.10. シーケンサー .....         | 36 |
| 3.10.1. 概要 .....           | 36 |
| 3.10.2. 設定領域 .....         | 37 |
| 3.10.3. ステータス .....        | 38 |
| 3.10.4. パラメーターセット .....    | 39 |
| 3.11. IO制御、トリガーとタイマー ..... | 41 |
| 3.11.1. IO制御 .....         | 41 |
| 3.11.2. トリガー .....         | 44 |
| 3.11.3. タイマー制御 .....       | 45 |
| 3.12. カメラステータスと情報 .....    | 46 |
| 3.12.1. センサー温度 .....       | 46 |
| 3.12.2. フレームカウンター .....    | 46 |
| 3.12.3. 現在の画素数 .....       | 46 |
| 3.12.4. 読み出し時間 .....       | 46 |

## 1. 基本情報

BaumerHXCシリーズのカメラの設定はいくつかのレジスタにアクセスする事で行われます。現在のカメラ設定の情報を得るために、これらのレジスタを特別な読み出しコマンドで読み取る事ができます。設定の変更は書き込みコマンドで行われます。

特に記載がない場合は全ての数字はビッグエンディアンの16進数として記載されています。また、全てのASCII文字は""で囲まれて表記されています。  
例："12345678"

### 警告！



記載されていない変更を行うとカメラが不具合を起こすかもしれません。マニュアルに記載されているレジスタだけを操作して下さい。

## 2. シリアルインターフェイス

### 2.1. シリアル通信のデフォルト設定

通信にはRS232Cを使用しており、以下の設定内容です。

| パラメーター   | 値         |
|----------|-----------|
| 通信速度     | 9600 baud |
| データビット   | 8*        |
| ストップビット  | 1*        |
| パリティチェック | なし*       |
| ハンドシェイク  | なし*       |

\*) 値は固定化されておりユーザー側で変更できません。

### 2.2. シリアルコマンド

カメラの内部レジスタは全てダブルワード型 (DWord) で調整されています。つまり、各値は常に32bitか4byteデータで構成されています。また、アドレスに格納されている32bitデータは4byteデータとして調整するため下位2bitが常に0です。

例：

有効なアドレス：0xf000404c

次の有効なアドレス：0xf0004050

設定するレジスタ値の間隔に関して連続ではない場合がある事に注意して下さい。これはカメラに存在する全てのアドレスが使用可能ではないからです。また、利用可能なアドレスでもアクセス制限のルールによってはアクセスを拒否されます。例えば読み出しのみのレジスタに対して書き込みを行った場合などです。

### コマンドタイプ :

カメラ内のレジスタは書き込みもしくは読み出しを使う事でアクセス可能です。  
その為、ユーザーは読み出しコマンド (R) や書き込みコマンド (W) をカメラに送信します。  
各コマンドに応じてカメラが読み出し応答や書き込み応答を返し、応答にはステータス情報や、読み込みコマンドによって要求されたデータが含まれます。  
ステータス情報はコマンド処理が正常に終了したかどうかをユーザーに伝えます。  
コマンドタイプに関する記述はセクション2.2.1.から2.2.5.に詳細が掲載されています。

### 開始と終了コマンド :

開始コマンドは最もはじめに送信されるコマンドで、ラインフィード (LF,0x0a) を読み込みコマンドや書き込みコマンドの直前に送信します。その他のコマンドを送信するとエラーとなり破棄されます。尚、このコマンドは1byte文字です。  
終了コマンドは最後に送信されるコマンドで、ラインフィード (LF,0x0a) とキャリッジリターン (CR,0x0d) を送信します。

### アドレスと長さ :

制御プロトコルにより単一のDWordデータの読み出し及び書き込みや、連続したDWordデータの読み出しや書き込みが可能です。これらの処理のために開始アドレスと読み出す長さ情報を与える必要があります。単一のDWordデータの処理の場合、開始アドレスは読み込みや書き込みを行うレジスタのアドレス値です。連続したデータ処理の場合は連続したアクセスを行う範囲の最初のアドレス値です。  
アドレスは常に32bit値です。

長さ情報は16bit値で、読み出したり書き込んだりするDWordデータの数を表します。  
下位互換性の為にこの値はDWordの数と同じ値ではなく、下記の計算式に従って算出します。

$$\text{長さ} = 4 \times (\text{処理するDwordデータの数} - 1)$$

つまり、1つのDWordデータを読み出す場合、対応する長さは0x0000となり、2つのDwordデータの場合は0x0004となり、3つのDwordデータの場合は0x0008となります。

#### 注意

アドレスや長さは、常に下位2bitが0になります。

### データ :

全てのデータは書き込みコマンドと共に送信されるか、読み出しコマンドの応答として受け取る32bit単位の単一もしくは複数のDWordで構成されたデータです。

## 文字エンコード：

コマンド送信は人間が読みやすい書式で行われます。つまり開始コマンドと終了コマンド以外の全てのコマンドはASCII書式で記述されておりバイナリデータを含んでいません。

アドレスや長さ、データやステータス情報などはビッグエンディアンの16進数表記として対応したASCII文字列に変換します。

例えば16bitのデータで1234の10進数の値は16進数では0x04d2となります。

ASCII書式で"04d2"とビッグエンディアンで変換すれば、これはバイナリ値では0x30 0x34 0x64 0x32を表します。

### 2.2.1. レジスタ読み出し

レジスタ読み出しコマンドでは、32bit単位のデータで最大16384DWord ("FFFC"が許容可能な最大長)の連続したデータを読み出し可能です。この場合、最初のアドレスのみにアクセスすれば、その後のアドレスは自動的に内部でインクリメントされます。

この読み出しコマンドによって、カメラは読み出し応答を返します。



コマンド長の合計は常に16byteになります。

例1：

1つの32bitDword値をアドレス0xf0004050から読み出すには以下のコマンドを送信します。



16進数値は変換されバイナリデータとしてカメラに転送されます。

0a 52 66 30 30 30 34 40 35 30 30 30 30 0a 0d

例2：

3つの連続したDword値をアドレス0xf0010000から読み出すには以下のコマンドを送信します。



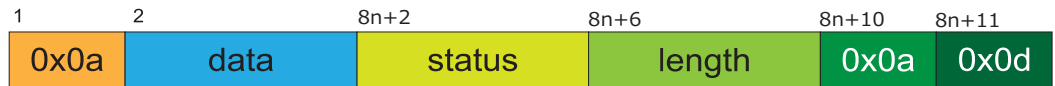
16進数値は変換されバイナリデータとしてカメラに転送されます。

0a 52 66 30 30 31 30 30 30 30 30 30 38 0a 0d

## 2.2.2. レジスタ読み出し応答

レジスタ応答は読み出しコマンドで与えられたレジスタ値に応じてカメラからユーザーへと送信されます。レジスタ読み出し応答の長さは、要求されたレジスタ読み出しのDword値の数に依存します。その為、データ長は0x0000 (1Dword) の長さで8byte、0x0004 (2Dword) で16byteになります。

ステータスも長さも同じ16bitの値で構成され、各フィールドでは4byteのデータに相当します。ステータス値の内容については2.2.5.に記載されています。



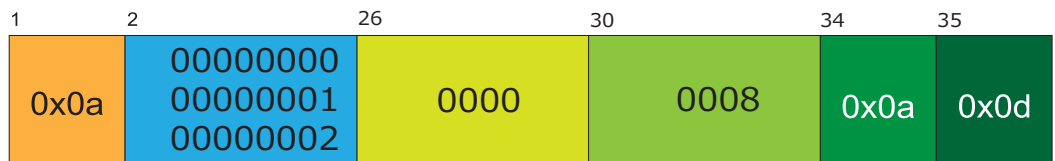
コマンドの長さの合計は読みだすデータの数に依存し、 $11+8n$  ( $n$ はDwordの数) の式で表されます。

### 注意

ステータスがGEV\_STATUS\_SUCCESSではない場合、応答は一切のデータを含んでおらず、コマンドの長さも11byteになるでしょう。この場合、長さのフィールドを無視しなければなりません。

例：

以下の応答例ではアドレス0xf0010000から3つのDwordデータを受け取っています。



3つのDwordデータを含むため、コマンドは35byteの長さになります。16進数値は変換されバイナリデータとして転送されます。

0a

30 30 30 30 30 30 30 30

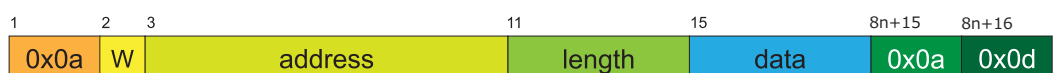
30 30 30 30 30 30 30 31

30 30 30 30 30 30 30 32

30 30 30 30 30 30 30 38 0a 0d

## 2.2.3. レジスタ書き込み

レジスタ書き込みコマンドでは、32bit単位のデータで最大16384Dwordの連続したデータを書き込み可能です。この場合、最初のアドレスのみにアクセスすれば、その後のアドレスは自動的に内部でインクリメントされます。コマンドの長さは送信するデータの数に依存します。この書き込みコマンドによって、カメラは書き込み応答を返します。



コマンドの長さの合計は書き込むデータの数に依存し、 $16+8n$  ( $n$ はDwordの数) の式で表されます。



例1 :

露光時間を4000 $\mu$ sに変更するため、アドレス0xf0004010に32bitデータで0x00000fa0（10進数で4000）を送信します。



1つのDwordデータを送信するため、コマンドは24byteの長さになります。  
16進数値は変換されバイナリデータとして転送されます。

0a 57 66 30 30 30 34 30 31 30 30 30 30 30  
30 30 30 30 30 66 61 30  
0a  
0d

例2 :

ルックアップテーブルに0x100,0xabc,0xcdeの3つの値を設定するため、アドレス0xf0010000にこれら3つの値を送信します。この場合、3つのDwordデータを書き込むため長さのフィールドは0x0008を指定します。

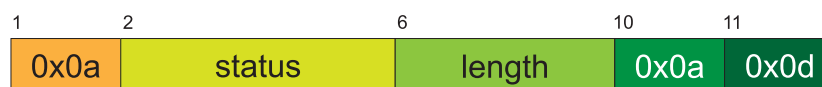


3つのDwordデータを送信するため、コマンドは40byteの長さになります。  
16進数値は変換されバイナリデータとして転送されます。

0a 57 66 30 30 31 30 30 30 30 30 30 30 38  
30 30 30 30 30 31 30 30  
30 30 30 30 30 61 62 63  
30 30 30 30 30 63 64 65  
0a  
0d

## 2.2.4. レジスタ書き込み応答

レジスタ書き込み応答は書き込みコマンドで与えられたレジスタ値に応じてカメラからユーザーへと送信されます。ステータスも長さも同じ16bitの値で構成され、各フィールドでは4byteのデータに相当します。ステータス値については2.2.5.に記載されています。



コマンドの長さの合計は常に11byteになります。

例：

存在しないアドレス値0xf0001234を書き込みした場合の応答例は以下の通りです。



16進数値は変換されバイナリデータとして転送されます。

0a 38 30 30 33 30 30 30 30 0a 0d

## 2.2.5. ステータス値

ステータス値はGigEVision規格と互換性を持っており、以下の通りです。

| 16進数値  | ステータス値                         | 内容   |
|--------|--------------------------------|--|
| 0x0000 | GEV_STATUS_SUCCESS             | コマンドは正常に処理されました。                               |
| 0x8003 | GEV_STATUS_INVALID_ADDRES<br>S | そのアドレスは存在しません。                                 |
| 0x8005 | GEV_STATUS_BAD_ALIGNMENT       | 指定されたアドレスのオフセットが間違っています。                       |
| 0x8006 | GEV_STATUS_ACCESS_DENIED       | 指定されたアドレスは無効です。<br>読み込み専用レジスタに値を書き込んだ場合に発生します。 |

## 2.2.6. コードサンプル

C++でシリアル通信部分を記述する場合のお参考例は以下の通りです。

### 2.2.6.1. シリアル読み出しコマンド

カメラからデータを読み出すサンプルです。

シリアル読み出しコマンドの書式は2.2.2.に記載されています。

```
CLUINT32 SerialRead(CLUINT32 iPort, void * serialRef, unsigned int address,
unsigned int numberofbytes)
{
    CLUINT32 iErr = 0;
    CLUINT32 bufSize = 17;
    char writeBuf[17];
    char * pWriteBuf = writebuf;
    int writelen = 4;
    char * endptr = NULL;

    memset((char*)writeBuf, 0x00, 17);

    //set LF sign
    *pWriteBuf++ = 0x0a;
    //set command sign
    *pWriteBuf++ = 'R';
    //set address
    sprintf(pWriteBuf, "%08x", address);
    pWriteBuf += 8;
    //set length
    sprintf(pWriteBuf, "04x", numberofbytes - 4);
    pWriteBuf += 4;
    //set LF and CR sign
    *pWriteBuf++ = 0a0d;
    *pWriteBuf++ = 0x0d

    //flush serial port
    iErr = clFlushPort(serialRef);
    if(iErr != CL_ERR_NO_ERR)
    {
        printf("clFlushPort failed with %s\n", iErr);
    }

    //write command
    iErr = clSerialWrite(seriarRef, (char*)writeBuf, &bufSize, timeout);
    if(iErr != CL_ERRR_NO_ERR)
    {
        printf("clSerialWrite failed with %s\n", iErr);
    }
    else
    {
        unsigned char readBuf[1024];
        unsigned char dataBuf[1024];
        CLUINT32 respSize = 1024;
        memset((char*)readBuf, 0, 1024);
        memset((char*)dataBuf, 0, 1024);

        iErr = clSerialRead(seriarRef, (char*)readBuf, &respSize, timeout);
        if(iErr != CL_ERR_NO_ERR && !((int)iErr == -10004&& respSize > 0))
        {
            printf("clSerialRead failed\n");
        }
        else
        {
            //parse status
            printf("Status:");
            unsigned char chr[5];
            memset(chr, 0, 5);
            memset(chr, (char*)&readBuf[respSize-10], 4);
```

```

unsigned int status = 0;
status = strtol((char*)chr, &endptr, 16);
switch(status)
{
    case GEV_STATUS_SUCCESS:
        printf("GEV_STATUS_SUCCESS\n");
        break;
    case GEV_STATUS_PACKET_RESEND:
        printf("GEV_STATUS_PACKET_RESEND\n");
        break;
    case GEV_STATUS_NOT_IMPLEMENTED:
        printf("GEV_STATUS_NOT_IMPLEMENTED\n");
        break;
    case GEV_STATUS_INVALID_PARAMETER:
        printf("GEV_STATUS_INVALID_PARAMETER\n");
        break;
    case GEV_STATUS_INVALID_ADDRESS:
        printf("GEV_STATUS_INVALID_ADDRESS\n");
        break;
    case GEV_STATUS_WRITE_PROTECT:
        printf("GEV_STATUS_WRITE_PROTECT\n");
        break;
    case GEV_STATUS_BAD_ALIGNMENT:
        printf("GEV_STATUS_BAD_ALIGNMENT\n");
        break;
    case GEV_STATUS_ACCESS_DENIED:
        printf("GEV_STATUS_ACCESS_DENIED\n");
        break;
    case GEV_STATUS_BUSY:
        printf("GEV_STATUS_BUSY\n");
        break;
    case GEV_STATUS_LOCAL_PROBLEM:
        printf("GEV_STATUS_LOCAL_PROBLEM\n");
        break;
    case GEV_STATUS_MSG_MISMATCH:
        printf("GEV_STATUS_MSG_MISMATCH\n");
        break;
    case GEV_STATUS_INVALID_PROTOCOL:
        printf("GEV_STATUS_INVALID_PROTOCOL\n");
        break;
    case GEV_STATUS_NO_MSG:
        printf("GEV_STATUS_NO_MSG\n");
        break;
    case GEV_STATUS_PACKET_UNAVAILABLE:
        printf("GEV_STATUS_PACKET_UNAVAILABLE\n");
        break;
    case GEV_STATUS_DATA_OVERRUN:
        printf("GEV_STATUS_DATA_OVERRUN\n");
        break;
    case GEV_STATUS_INVALID_HEADER:
        printf("GEV_STATUS_INVALID_HEADER\n");
        break;
    case GEV_STATUS_WRONG_CONFIG:
        printf("GEV_STATUS_WRONG_CONFIG\n");
        break;
    case GEV_STATUS_PACKET_NOT_YET_AVAILABLE:
        printf("GEV_STATUS_PACKET_NOT_YET_AVAILABLE\n");
        break;
    case GEV_STATUS_PACKET_AND_PREV_REMOVED_FROM_MEMORY:
        printf("GEV_STATUS_PACKET_AND_PREV_REMOVED_FROM_MEMORY\n");
        break;
    case GEV_STATUS_PACKET_REMOVED_FROM_MEMORY:
        printf("GEV_STATUS_PACKET_REMOVED_FROM_MEMORY\n");
        break;
    case GEV_STATUS_ERROR:
        printf("GEV_STATUS_ERROR\n");
        break;
    default:
        printf("unknown status 0x%x\n", status);
        break;
}

//parse length
memcpy(chr, (char*)&readBuf[respSize-6], 4);
unsigned int rlen = strtol((char*)chr, &endptr, 16);
printf("Length: %d bytes\n", rlen+4);

```

```

        //parse data
        memcpy((char*)dataBuf, (char*)&readBuf[1], (rlen + 4)*2);
        printf("Data: %s\n", (char*)dataBuf);
    }
}
return iErr;
}

```

## 2.2.6.2. レジスタ書き込み応答

カメラにデータを書き込むサンプルです。  
書き込みコマンドの書式は2.2.3.に掲載されています。

```

CLUINT32 SerialWrite(CLUINT32 iPort, void* serialRef, unsigned int address, unsigned int data)
{
    CLUINT32 iErr = 0;
    CLUINT32 bufSize = 25;
    char writeBuf[25];
    CLUINT32 timeout = 2000;
    int writelen = 4;

    char * pWriteBuf = writeBuf;
    char * pWriteBuf = writeBuf;
    char * endptr = NULL;

    memset((char *)writeBuf, 0x00, 25);

    //sel LF sign
    *pWriteBuf++ = 0x0a;
    //set command sign
    *pWriteBuf++ = 'W';
    //set address
    sprintf(pWriteBuf, "%08x", address);
    pWriteBuf += 8;
    //set length
    sprintf(pWriteBuf, "%04x", writelen - 4);
    pWriteBuf += 4;
    //set data
    sprintf(pWriteBuf, "%08x", data);
    pWriteBuf += 8;
    //set LF and CR sign
    *pWriteBuf++ = 0x0a;
    *pWriteBuf++ = 0x0d;

    printf("Write command %s\n", (char*)writeBuf);

    //flush serial port
    iErr = clFlushPort(serialRef);
    if(iErr != CL_ERR_NO_ERR)
    {
        printf("clFlushPort failed with %s\n", iErr);
    }

    iErr = clSerialWrite(serialRef, (char*)writeBuf, &bufSize, timeout);
    if(iErr != CL_ERR_NO_ERR)
    {
        printf("clSerialWrite failed\n");
    }
    else
    {
        unsigned char readBuf[1024];
        CLUINT32 respSize = 1024;
        memset((char *)readBuf, 0, 1024);

        iErr = clSerialRead(serialRef, (char *)readBuf, &respSize, timeout);
        if(iErr != CL_ERR_NO_ERR && !((int)iErr == -10004 && respSize > 0))
        {
            printf("clSerialRead failed\n");
        }
    }
}

```

```

else
{
//parse status
printf("Status:");
unsigned char chr[5];
memset(chr, 0, 5);
memcpy(chr, (char*)&readBuf[1], 4);
unsigned int status = 0;
status = strtol((char*)chr, &endptr, 16);
switch(status)
{
case GEV_STATUS_SUCCESS:
printf("GEV_STATUS_SUCCESS\n");
break;
case GEV_STATUS_PACKET_RESEND:
printf("GEV_STATUS_PACKET_RESEND\n");
break;
case GEV_STATUS_NOT_IMPLEMENTED:
printf("GEV_STATUS_NOT_IMPLEMENTED\n");
break;
case GEV_STATUS_INVALID_PARAMETER:
printf("GEV_STATUS_INVALID_PARAMETER\n");
break;
case GEV_STATUS_INVALID_ADDRESS:
printf("GEV_STATUS_INVALID_ADDRESS\n");
break;
case GEV_STATUS_WRITE_PROTECT:
printf("GEV_STATUS_WRITE_PROTECT\n");
break;
case GEV_STATUS_BAD_ALIGNMENT:
printf("GEV_STATUS_BAD_ALIGNMENT\n");
break;
case GEV_STATUS_ACCESS_DENIED:
printf("GEV_STATUS_ACCESS_DENIED\n");
break;
case GEV_STATUS_BUSY:
printf("GEV_STATUS_BUSY\n");
break;
case GEV_STATUS_LOCAL_PROBLEM:
printf("GEV_STATUS_LOCAL_PROBLEM\n");
break;
case GEV_STATUS_MSG_MISMATCH:
printf("GEV_STATUS_MSG_MISMATCH\n");
break;
case GEV_STATUS_INVALID_PROTOCOL:
printf("GEV_STATUS_INVALID_PROTOCOL\n");
break;
case GEV_STATUS_NO_MSG:
printf("GEV_STATUS_NO_MSG\n");
break;
case GEV_STATUS_PACKET_UNAVAILABLE:
printf("GEV_STATUS_PACKET_UNAVAILABLE\n");
break;
case GEV_STATUS_DATA_OVERRUN:
printf("GEV_STATUS_DATA_OVERRUN\n");
break;
case GEV_STATUS_INVALID_HEADER:
printf("GEV_STATUS_INVALID_HEADER\n");
break;
case GEV_STATUS_WRONG_CONFIG:
printf("GEV_STATUS_WRONG_CONFIG\n");
break;
case GEV_STATUS_PACKET_NOT_YET_AVAILABLE:
printf("GEV_STATUS_PACKET_NOT_YET_AVAILABLE\n");
break;
case GEV_STATUS_PACKET_AND_PREV_REMOVED_FROM_MEMORY:
printf("GEV_STATUS_PACKET_AND_PREV_REMOVED_FROM_MEMORY\n");
break;
case GEV_STATUS_PACKET_REMOVED_FROM_MEMORY:
printf("GEV_STATUS_PACKET_REMOVED_FROM_MEMORY\n");
break;
case GEV_STATUS_ERROR:
printf("GEV_STATUS_ERROR\n");
break;
default:
printf("unknown status 0x%x\n", status);
break;
}
}
}

```

```
    }  
    //parse length  
    memcpy(chr, (char*)&readBuf[5], 4);  
    unsigned int rlen = strtol((char*)chr, &endptr, 16);  
    printf("Length: %d bytes\n", rlen + 4);  
    }  
    }  
    return iErr;  
}
```

### 3. レジスタ内容

#### 3.1. デバイス情報

Baumer HXCカメラにはカメラを識別するためのいくつかのデバイス情報が備わっています。ほとんどの値は0の文字で終端された文字列です。つまり、文字内容の左側は常に0で埋められています。

##### 3.1.1. 製造会社名

|            |  |
|------------|--|
| アドレス       | 0x00000048   |
| 長さ (DWord) | 8 (32Byte : 64文字相当)  |
| アクセスタイプ    | R (読み出しのみ)   |
| 設定内容と“出力例” | “Baumer Optronic”<br>0x4261756d - “Baum”<br>0x6572204f - “er O”<br>0x7074726f - “ptro”<br>0x6e696300 - “nic”<br>0x00<br>0x00<br>0x00<br>0x00 |

例：

値を読み出すために以下のコマンドをカメラに送信します。



バイナリデータだと以下の通りです。

0a 52 30 30 30 30 30 30 34 38 30 30 31 63 0a 0d



カメラは要求された情報を含む読み出し応答を返します。

**注意**

見て分かる通り、“Baumer Optronic”の文字の空間は0で埋められています。

|      |   |   |          |      |      |
|------|---|---|----------|------|------|
| 1    | 2 |   | 70       | 74   | 75   |
| 0x0a |   | 4261756d<br>6572204f<br>7074726f<br>6e696300<br>(4回 0x00000000) | 0000001c | 0x0a | 0x0d |

バイナリデータだと以下の通りです。

0a

```
34 32 36 31 37 35 36 64  36 35 37 32 32 30 34 66
37 30 37 34 37 32 36 66  36 65 36 39 36 33 30 30
30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30
30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30
30 30 30 30 30 30 31 63
```

0a

0d

### 3.1.2. モデル名

|            |                     |          |
|------------|---------------------|----------|
| アドレス       | 0x00000068          |          |
| 長さ (DWord) | 0x00000048          |          |
| アクセスタイプ    | R                   |          |
| 設定内容と"出力例" | "HXC40"             |          |
|            | 0x48584334          | - "HXC4" |
|            | 0x30 0x00 0x00 0x00 | - "0"    |
|            | 0x00                |          |
|            | 0x00                |          |
|            | 0x00                |          |
|            | 0x00                |          |
|            | 0x00                |          |
|            | 0x00                |          |
|            | 0x00                |          |

### 3.1.3. Firmware / CID

ファームウェアバージョンはCID (Configuration ID) から参照可能です。  
カメラ内で使用されている設定ファイルには完全な設定情報が含まれますが、Baumerでは数値  
しか参照しません。また、このレジスタは3.1.1.の記述のように0の終端文字を含みます。

ファームウェアVer1.0のCIDは02から始まり、ファームウェアVer2.0のCIDは03から始まりま  
す。(CID:030004 = Firmware2.0)

|            |  |
|------------|--|
| アドレス       | 0x00000088   |
| 長さ (DWord) | 8  |
| アクセスタイプ    | R  |
| 設定内容と"出力例" | "030004"<br>0x30333030 - "0300"<br>0x30340000 - "04"<br>0x00<br>0x00<br>0x00<br>0x00<br>0x00<br>0x00<br>0x00 |

### 3.1.4. シリアル番号

デバイスのシリアル番号です。もちろん全てのデバイスはそれぞれ固有の番号を持っています。  
このレジスタは3.1.1.の記述のように0の終端文字を含みます。

|            |   |
|------------|---|
| アドレス       | 0x000000d8  |
| 長さ (DWord) | 8   |
| アクセスタイプ    | R   |
| 設定内容と"出力例" | "00001111910"<br>0x30303030 - "0000"<br>0x31313131 - "1111"<br>0x39313000 - "910"<br>0x00<br>0x00<br>0x00<br>0x00<br>0x00 |

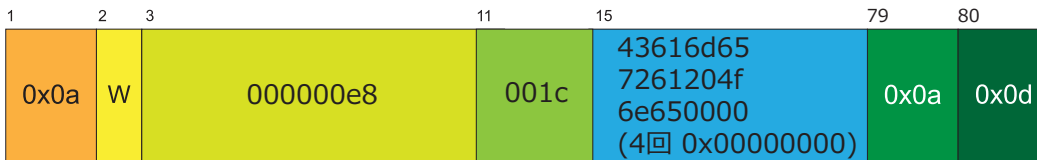
### 3.1.5. ユーザ一定義名

デフォルトのユーザー定義名はモデル名 (3.1.2.参照) と同じです。ただ、ユーザーがそれを自由な値に変更する事が可能です。また、このレジスタは3.1.1.の記述のように0の終端文字を含みます。ユーザー定義名の変更は内部フラッシュ領域に格納されるので、電源を落としても記録されます。

|            |   |
|------------|---|
| アドレス       | 0x000000e8  |
| 長さ (DWord) | 8   |
| アクセスタイプ    | RW (読み出しと書き込み)  |
| 設定内容と"出力例" | "Camera One"<br>0x43616D65 - "Came"<br>0x7261204F - "ra O"<br>0x6E650000 - "ne"<br>0x00<br>0x00<br>0x00<br>0x00<br>0x00 |

例 :

ユーザー定義名を"Camera One"に設定したい場合、以下の例のコマンドを実行しなければなりません。

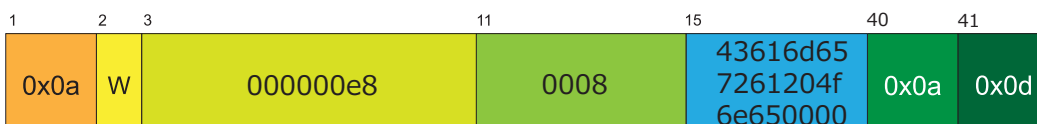


バイナリデータだと以下の通りです。

```

0a  57  30 30 30 30 30 30 30 65 38   30 30 31 63
34 33 36 31 36 64 36 35  37 32 36 31 32 30 34 66
36 65 36 35 30 30 30 30  30 30 30 30 30 30 30 30
30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30
30 30 30 30 30 30 30 30
0a  0d
  
```

送信するデータ内容で10文字しか使用しない場合、8Dwordの代わりに3Dwordを長さの値として使用し、8を指定する事ができます。その場合以下のコマンドになります。



バイナリデータだと以下の通りです。

0a 57 30 30 30 30 30 30 65 38 30 30 30 38  
34 33 36 31 36 64 36 35  
37 32 36 31 32 30 34 66  
36 65 36 35 30 30 30 30  
0a 0d

### 3.2. センサーバージョン

Baumer HXCシリーズでは異なるセンサーバージョンが使用されています。  
どのセンサーを使用しているのか読み出す事が可能です。

|            |  |
|------------|--|
| アドレス       | 0xF0004600   |
| 長さ (DWord) | 1  |
| アクセスタイプ    | R  |
| 設定内容と“出力例” | 0x00000020 - CMV2000 V2<br>0x00000040 - CMV4000 V2<br>0x00000023 - CMV2000 V3<br>0x00000043 - CMV4000 V3 |

### 3.3. ユーザーセット

3つのユーザーセット（1～3）がBaumer HXCカメラシリーズでは利用可能です。  
ユーザーセットには以下の情報を含みます。

| パラメーター               | パラメーター                           |
|----------------------|----------------------------------|
| Binning Mode         | Mirroring Control                |
| CameraLink® Control  | Offset                           |
| Defectpixellist      | Partial Scan                     |
| Digital I/O Settings | Pixelformat                      |
| Exposure Time        | Readout Mode / Digitization Taps |
| Gain Factor          | Testpattern                      |
| Look-Up-Table        | Trigger Settings                 |
| Shutter Mode         | Fixed Frame rate                 |
| Color Gains          | Gamma                            |
| Speed Mode           | IO-Settings                      |

3つのユーザーセットの内容はカメラ内にのみ記録でき、デバイス外に設定内容を入力することはできません。

|            |   |
|------------|---|
| アドレス       | 0x7000001c  |
| 長さ (DWord) | 1   |
| アクセスタイプ    | RW  |
| 設定内容と"出力例" | <p>ユーザーセットの読み出し</p> <p>User Set 0 : 0x81000000 (工場出荷設定)<br/> User Set 1 : 0x81000001<br/> User Set 2 : 0x81000002<br/> User Set 3 : 0x81000003</p> <p>ユーザーセットの書き出し</p> <p>User Set 0 : 工場出荷設定は書き込み保護されています<br/> User Set 1 : 0x82000001<br/> User Set 2 : 0x82000002<br/> User Set 3 : 0x82000003</p> |

### 注意

カメラの再起動 : 0x84000000

### 3.3.1. デフォルトユーザーセット選択

デフォルトユーザーセット選択では、カメラ起動時に自動的にパラメータを読み出すユーザーセットを選択します。

### 注意

このレジスタには他にも重要なカメラ設定の値を含みます。  
デフォルトユーザーセットを指定する前に必ず現在値を確認し、  
bit[0]とbit[1]の値以外は変更しないで下さい。

|            |   |
|------------|---|
| アドレス       | 0x0000a004 - bit[0]とbit[1]のみを操作   |
| 長さ (DWord) | 1   |
| アクセスタイプ    | RW  |
| 設定内容と"出力例" | <p>bit[0],bit[1]のみを操作</p> <p>User Set 0 : 0x*****0 (工場出荷設定)<br/> User Set 1 : 0x*****1<br/> User Set 2 : 0x*****2<br/> User Set 3 : 0x*****3</p> <p>(*****は読み出した値に依存)</p> |

## 3.4. CameraLink® インターフェイス機能

### 3.4.1. シリアル通信速度

シリアル通信インターフェイスの速度はレジスタ0x80000104で制御可能です。値に従ってボーレートが変更されます。デフォルトでは互換性確保のため最も低い値に設定されています。

|            |                         |
|------------|-------------------------|
| アドレス       | 0x80000104              |
| 長さ (DWord) | 1                       |
| アクセスタイプ    | RW                      |
| 設定内容と“出力例” | 0x00000002 - 19200 baud |

#### 注意

この設定はユーザーセットには保存されません。カメラは常に9600baudで起動します。

| 設定値        | ボーレート       |
|------------|-------------|
| 0x00000001 | 9600 baud   |
| 0x00000002 | 19200 baud  |
| 0x00000004 | 38400 baud  |
| 0x00000008 | 57600 baud  |
| 0x00000010 | 115200 baud |
| 0x00000020 | 230400 baud |
| 0x00000040 | 460800 baud |
| 0x00000080 | 921000 baud |

#### 注意

ボーレートを変更した後、設定はただちに適用されます。ボーレートを変更した後の書き込み応答は新しいボーレートで送信される為、素早く通信ソフト側のボーレートの切り替えを行わなければ書き込み応答は失われます。  
回避策として、設定変更した直後かしばらくしてから通信ソフトのシリアルポートを新しいボーレートで再設定して下さい。

再びレジスタの値を得るために読み出しコマンドを実行し、読み出し応答を受け取って下さい。

### 3.4.2. タップ制御

BaumerHXCシリーズで使用されているCameraLink®インターフェイスでは異なるCameraLinkタップ設定を利用可能です。使用するフレームグラバー次第ですが、最大10Tap設定までデータ転送に利用可能です。詳細はカメラのユーザーガイドを確認して下さい。

|            |                   |
|------------|-------------------|
| アドレス       | 0x80000008        |
| 長さ (DWord) | 1                 |
| アクセスタイプ    | RW                |
| 設定内容と"出力例" | 0x00000002 - 2Tap |

| 設定値        |              |
|------------|--------------|
| 0x00000000 | 0 Tap (動作停止) |
| 0x00000001 | 1 Tap        |
| 0x00000002 | 2 Tap        |
| 0x00000003 | 3 Tap        |
| 0x00000004 | 4 Tap        |
| 0x00000008 | 8 Tap        |
| 0x0000000a | 10 Tap       |

#### 注意

そのTap設定が利用可能かどうかは、ピクセルフォーマットの設定に依存します。例えば10bitもしくは12bitのフォーマットの場合、最大4Tapまでの設定しか利用できません。10bitのフォーマットで8Tapを指定してもカメラは4Tapを使用します。8bitのフォーマットの時のみ全てのTap設定が利用可能です。

### 3.4.3. センサー読み出し速度（クロック）

センサーの読み出し速度はセンサー速度レジスタで設定可能です。  
より低い速度を使用した場合、出力される信号の伝送品質が上がり、消費電力は下がります。

|            |  |
|------------|--|
| アドレス       | 0xf0004814   |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と“出力例” | 0x00000000 - 48MHz (8/10bit)<br>0x00000001 - 40MHz (8/10/12bit)<br>0x00000002 - 24MHz (8/10bit, 20MHz 12bit) |

## 3.5. 画像取得

### 3.5.1. 画像取得の停止

画像取得を停止や開始するために、画像取得停止機能を使用可能です。使用するにはレジスタ 0xf0004000のbit[1]を変更します。元に戻せば画像取得は再開されます。

|            |  |
|------------|--|
| アドレス       | 0xf0004000 - bit[1]のみを操作   |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と“出力例” | 0x0000020D ([bit]1 = 0) - 画像取得開始<br>0x0000020F ([bit]1 = 1) - 画像取得停止 |

#### 注意

このレジスタには、他にも重要なカメラ設定の値（シーケンサーやLUT等）を含みます。  
画像取得の停止と開始を行うために必ず現在値を確認し、bit[1]の値以外は変更しないで下さい。

### 3.5.2. 露光時間

カメラの露光時間はマイクロセカンド単位で設定する事が可能です。

|            |                                |
|------------|--------------------------------|
| アドレス       | 0xf0004010                     |
| 長さ (DWord) | 1                              |
| アクセスタイプ    | RW                             |
| 設定内容と“出力例” | 0x00000FA0 - 露光時間 4000 $\mu$ s |



### 3.5.3. PRNU/DSNU補正 (FPN - 固定パターンノイズ補正)

PRNU / DSNU補正はアドレス0xf0004000のbit[27]を変更する事で行います。  
変更する前にそのレジスタの現在値を読み出し、bit[27]だけを書き換える事を推奨します。

PRNU / DSNU補正を有効にするにはレジスタの値に対してビット単位でのOR演算を  
0x08000000のビットマスクを使用して行います。

PRNU / DSNU補正を無効にするにはレジスタの値に対してビット単位でのAND演算を  
0xF7FFFFFFのビットマスクを使用して行います。

|            |  |
|------------|--|
| アドレス       | 0xf0000400 - bit[27]のみを操作  |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と"出力例" | <u>PRNU / DSNU 有効</u><br><br>読み出した現在値 : 0x02000A0D<br>OR演算のマスク : 0x08000000<br>書き込む変更値 : 0x0A000A0D<br><br><u>PRNU / DSNU 無効</u><br><br>読み出した現在値 : 0x0A00A0D<br>AND演算のマスク : 0xF7FFFFFF<br>書き込む変更値 : 0x02000A0D |

### 3.5.4. ハイダイナミックレンジ (HDR)

HDRはいくつかの技術により画像センサーの標準のダイナミックレンジをより明るさの幅（最も明るい部分から最も暗い部分）を持つダイナミックレンジに拡張する機能です。

HDRはアドレス0xf0004000のbit[22]を変更する事で行います。  
変更する前にそのレジスタの現在値を読み出し、bit[22]だけを書き換える事を推奨します。

HDRを有効にするにはレジスタの値に対してビット単位でのOR演算を0x00400000のビットマスクを使用して行います。

HDRを無効にするにはレジスタの値に対してビット単位でのAND演算を0xFFBFFFFFFのビットマスクを使用して行います。

#### HDRの有効化

|            |  |
|------------|--|
| アドレス       | 0xf0004000 - bit[22]のみを操作  |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と"出力例" | <u>HDRの有効化</u><br><br>読み出した現在値 : 0x02000A0D<br>OR演算のマスク : 0x00400000<br>書き込む変更値 : 0x02400A0D |

## HDRトリガー自動モード

このモードはアドレス0xf0004000のbit[6]を変更する事で行います。  
変更する前にそのレジスタの現在値を読み出し、bit[6]だけを書き換える事を推奨します。

HDRトリガー自動モードを有効にするにはレジスタの値に対してビット単位でのOR演算を0x00000040のビットマスクを使用して行います。

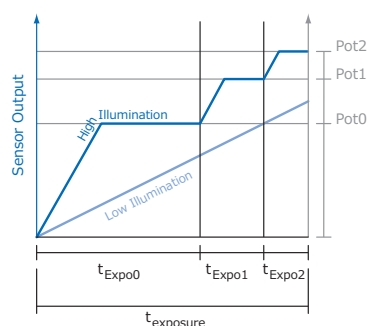
HDRトリガー自動モードを無効にするにはレジスタの値に対してビット単位でのAND演算を0xFFFFFBFのビットマスクを使用して行います。

### HDRトリガー自動モードの有効化

|            |  |
|------------|--|
| アドレス       | 0xf0000400 - bit[6]のみを操作   |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と“出力例” | HDRトリガー自動モードの有効化<br><br>読み出した現在値 : 0x02400A0D<br>OR演算のマスク : 0x00000040<br>書き込む変更値 : 0x02400A4D |

### HDRの設定値

|                                |   |
|--------------------------------|---|
| アドレス                           | 0xf0004500 + (idxHDR * 4)<br><br>$t_{exp}$ の番号 = 0~1 = idxHDR<br>bit[0~5] : 電荷レベル (Pot)<br>bit[8~15] : 総露光時間に対する割合 ( $t_{exp}$ )  |
| 長さ (DWord)                     | 1   |
| アクセスタイプ                        | RW  |
| 設定内容と“出力例”                     | $t_{exp0}$ の設定 : アドレス0xf0004500<br>Pot0 = 42 = 0x2A<br>$t_{exp0}$ = 191 = 0xBF<br>書き込む設定値 : 0x0000BF2A<br><br>$t_{exp1}$ の設定 : アドレス0xf0004504<br>Pot1 = 21 = 0x15<br>$t_{exp1}$ = 50 = 0x32<br>書き込む設定値 : 0x00003215 |
| <b>注意</b>                      |   |
| $t_{exp0} + t_{exp1} \leq 255$ |   |



### 3.5.5. 読み出しモード

カメラはオーバーラップとシーケンシャルの2つの読み出しモードをサポートします。  
オーバーラップモードを使用しない場合、レジスタ0xf0004014のbit[0]を0に指定します。

|            |  |
|------------|--|
| アドレス       | 0xf0004014 - bit[0]のみを操作                                       |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と“出力例” | 0x00000001 - オーバーラップ読み出しモード有効<br>0x00000000 - オーバーラップ読み出しモード無効 |

### 3.5.6. 固定フレームレート

カメラのフレームレートは通常露光時間とセンサーの読み出し時間に依存します。最速のフレームレートにするにはオーバーラップモードで読み出し、露光時間が読み出し時間よりも短くなる必要があります。この場合、読み出し時間によって最大フレームレートが決まります。この機能はそれとは異なりより低いフレームレートで出力するための機能になります。

固定フレームレートでは読み出し時間を $\mu\text{s}$ 単位で指定します。  
つまり10fpsで出力したい場合、100000 $\mu\text{s}$  (100ms) に指定します。  
機能を有効化するには対象レジスタのbit[31]を変更します。

読み出しに関するさらなる情報はユーザーガイドをご確認ください。

|   |  |           |   |  |                  |
|---|--|-----------|---|--|------------------|
| アドレス  | 0xf000403c - bit[0~29] : 読み出し時間の長さ<br>- bit[31] : 機能の有効/無効 |           |   |  |                  |
| 長さ (DWord)  | 1  |           |   |  |                  |
| アクセスタイプ   | RW   |           |   |  |                  |
| 設定内容と“出力例”  | 0x00000000 - 機能のオフ<br>0x8000C350 - 20fpsに固定                |           |   |  |                  |
| <table border="1"><tr><td><b>注意</b></td></tr><tr><td><math display="block">\text{fps} = \frac{1000000}{\text{設定値}}</math></td></tr><tr><td><math display="block">\frac{1000000}{50000} = 20 \text{ fps}</math></td></tr><tr><td><math display="block">50000 = 0xC350</math></td></tr></table> |  | <b>注意</b> | $\text{fps} = \frac{1000000}{\text{設定値}}$ | $\frac{1000000}{50000} = 20 \text{ fps}$ | $50000 = 0xC350$ |
| <b>注意</b>   |  |           |   |  |                  |
| $\text{fps} = \frac{1000000}{\text{設定値}}$   |  |           |   |  |                  |
| $\frac{1000000}{50000} = 20 \text{ fps}$  |  |           |   |  |                  |
| $50000 = 0xC350$  |  |           |   |  |                  |

### 3.5.7. パーシャルスキャン (ROI)

BaumerHXCカメラは“パーシャルスキャン (ROI)”をサポートしており、これにより読み出し時間を削減し、フレームレートの最大値を向上できます。ROIを設定するには読み出しを行う画素領域の起点とサイズを指定します。

HXCカメラではROIは常にセンサーの最大解像度まで適用可能で、これはビニングモードも同様です。

ROIは16bit長のStartX, StartY, SizeX, SizeYの4つのパラメーターで構成されています。これらのパラメーターはROI-StartとROI-Sizeの2つのレジスタにそれぞれ格納されます。

|                                 |                                  |
|---------------------------------|----------------------------------|
| StartX: ROI-Startレジスタのbit[0~15] | StartY: ROI-Startレジスタのbit[16~31] |
| SizeX: ROI-Sizeレジスタのbit[0~15]   | SizeY: ROI-Sizeレジスタのbit[16~31]   |

例えば、X座標96、Y座標217から横256画素、縦123画素の内容でROIを設定する場合、以下の通りです。

|               |              |
|---------------|--------------|
| <b>StartX</b> | 96 = 0x0060  |
| <b>StartY</b> | 217 = 0x00D9 |
| <b>SizeX</b>  | 256 = 0x0100 |
| <b>SizeY</b>  | 123 = 0x007B |

レジスタには以下の内容で設定します。

|                  |            |
|------------------|------------|
| <b>ROI-Start</b> | 0x00D90060 |
| <b>ROI-Size</b>  | 0x007B0100 |

|                   |  |
|-------------------|--|
| <b>アドレス</b>       | 0xf0004018 - ROI-Start<br>0xf000401c - ROI-Size  |
| <b>長さ (DWord)</b> | 1  |
| <b>アクセスタイプ</b>    | RW   |
| <b>設定内容と“出力例”</b> | HXC40の最大解像度に戻す場合<br>アドレス0xf0004018 -> 書き込む設定値 : 0x00000000<br>アドレス0xf000401c -> 書き込む設定値 : 0x08000800 |

### 3.5.8. マルチROI

#### 注意

マルチROIはビニングやサブサンプルの機能と併用できません。

BaumerHXCカメラは“マルチROI”をサポートしており、いくつかのROIを設定する事が可能です。設定では最大8箇所の縦横それぞれの境界線（最大64箇所のROI）を定義可能です。重なりあうマルチROI領域はカメラ内で結合され、整列して出力されます。

#### マルチROIの有効化

|            |  |
|------------|--|
| アドレス       | 0xf0004438 - bit[0~7] : Multi-HROI<br>- bit[8~15] : Multi-VROI                               |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と“出力例” | Multi-HROI1, Multi-HROI2, Multi-VROI1を有効にする場合<br>設定値 : 0x00000103 - bit[0~1]が1(ON), bit[8]が1 |

#### Multi-HROI (マルチROIの横方向設定)

|            |   |
|------------|---|
| アドレス       | 0xf0004440 + (idxHROI * 4)<br><br>Multi-HROIの番号 = 0~7 = idxHROI<br>bit[16~31] : マルチROIの横画素数<br>bit[0~15] : マルチROIを開始するX座標 |
| 長さ (DWord) | 1   |
| アクセスタイプ    | RW  |
| 設定内容と“出力例” | Multi-HROI1<br><br>X座標 = 64 = 0x0040<br>横画素数 = 256 = 0x0100<br>書き込む設定値 : 0x01000040                                       |

#### Multi-VROI (マルチROIの縦方向設定)

|            |   |
|------------|---|
| アドレス       | 0xf0004460 + (idxVROI * 4)<br><br>Multi-VROIの番号 = 0~7 = idxVROI<br>bit[16~31] : マルチROIの縦画素数<br>bit[0~15] : マルチROIを開始するY座標 |
| 長さ (DWord) | 1   |
| アクセスタイプ    | RW  |
| 設定内容と“出力例” | Multi-VROI1<br><br>Y座標 = 64 = 0x0040<br>縦画素数 = 256 = 0x0100<br>書き込む設定値 : 0x01000040                                       |

### 3.5.9. ビニングとサブサンプル

#### 注意

この機能はモノクロカメラのみ利用可能です。

HXCカメラはビニングとサブサンプルをサポートしています。これらの設定を行う場合、アドレス0xf0004020にあるビニング制御レジスタを以下の設定内容に従って変更します。

| 値          | データ長  | 制御するbit位置 | 利用可能な値       | 内容                       |
|------------|-------|-----------|--------------|--------------------------|
| BinningX   | 8 bit | bit[0~7]  | 0x00 or 0x01 | X方向のビニングもしくはサブサンプルの有効/無効 |
| BinningY   | 8 bit | bit[8~15] | 0x00 or 0x01 | Y方向のビニングもしくはサブサンプルの有効/無効 |
| ビニング補正     | 1 bit | bit[16]   | 0x0 or 0x1   | ビニング補正の有効化               |
| SubsampleX | 1 bit | bit[17]   | 0x0 or 0x1   | X方向のビニングの代わりにサブサンプルを有効化  |
| SubsampleY | 1 bit | bit[18]   | 0x0 or 0x1   | Y方向のビニングの代わりにサブサンプルを有効化  |

#### 注意

0x00000000をビニング制御レジスタに書き込んだ場合、ビニングもサブサンプルも完全に無効化されます。

尚、全ての設定組み合わせが利用可能ではない事に注意して下さい。

例えば、センサーはY方向のビニングをサポートしていませんが、Y方向のサブサンプルはサポートしています。従って、Y方向のビニングの設定のみを実行しても適用されません。ビニングとサブサンプルの有効な組み合わせに関してはユーザーガイドをご確認下さい。

|            |  |
|------------|--|
| アドレス       | 0xf0004020   |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と“出力例” | ビニングとサブサンプルの無効化<br>書き込む設定値 : 0x00000000<br><br>X方向のビニングとビニング補正の有効化<br>書き込む設定値 : 0x00010001<br><br>X方向とY方向のサブサンプル、およびビニング補正の有効化<br>書き込む設定値 : 0x00050101 |

### 3.5.10. 画像反転

#### 注意

この機能はモノクロカメラのみ利用可能です。

全ての読み出しモードとピクセルフォーマットにおいて、画像を縦横それぞれ反転する事ができます。これには反転レジスタ0xf000406cのbit[0]とbit[1]を変更して、横方向や縦方向の反転を操作します。

|            |   |
|------------|---|
| アドレス       | 0xf000406c  |
| 長さ (DWord) | 1   |
| アクセスタイプ    | RW  |
| 設定内容と"出力例" | 0x00000000 - 反転無効化<br>0x00000001 - 水平方向反転<br>0x00000002 - 垂直方向反転<br>0x00000003 - 水平方向と垂直方向の反転 |

### 3.5.11. センサーデジタイザータップ

HXCに使用されているセンサーは複数の読み出しモードをサポートしており、2,4,8,16タップ転送が利用可能です。これに関してピクセルフォーマットによる制限などより詳しく知りたい場合はユーザーガイドをご確認下さい。

|            |   |
|------------|---|
| アドレス       | 0xf0004028  |
| 長さ (DWord) | 1   |
| アクセスタイプ    | RW  |
| 設定内容と"出力例" | 0x00000000 - 2Tap<br>0x00000001 - 4Tap<br>0x00000002 - 8Tap<br>0x00000004 - 16Tap |

### 3.5.12. テストパターン

この機能は取得される画像に特定のパターン模様が上書きで適用されますが、センサーデジタイザータップ (3.3.11) の現在値に依存します。縦方向と横方向がそれぞれ選択可能です。

|            |  |
|------------|--|
| アドレス       | 0xf0004064   |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と"出力例" | 0x00000000 - Off<br>0x00000001 - 横方向<br>0x00000002 - 縦方向 |

### 3.6. ピクセルフォーマット

カメラのピクセルフォーマットはピクセルフォーマットレジスタを変更して操作します。  
より詳しい情報はユーザーガイドをご確認下さい。

|            |   |
|------------|---|
| アドレス       | 0xf0004040  |
| 長さ (DWord) | 1   |
| アクセスタイプ    | RW  |
| 設定内容と“出力例” | <u>モノクロカメラの場合</u><br>0x01080001 - モノクロ 8bit<br>0x01100003 - モノクロ 10bit<br>0x01100005 - モノクロ 12bit<br><br><u>カラーカメラの場合</u><br>0x0108000A - ベイヤーGB 8bit<br>0x0110000E - ベイヤーGB 10bit<br>0x01100012 - ベイヤーGB 12bit |

### 3.7. ゲインとオフセット

#### 3.7.1. ゲイン

カメラのデジタルゲインはゲインレジスタで操作します。値は16bitの小数点で指定し、4bitの整数値と12bitの少数値で構成されています。  
値はゲインレジスタの下位16bitに格納されます。

#### 注意

他のbitはゲインレジスタの他の機能に影響を与えます。  
決して下位16bit以外のbitは操作しないで下さい。

|            |   |
|------------|---|
| アドレス       | 0xf0004050                                  |
| 長さ (DWord) | 1   |
| アクセスタイプ    | RW  |
| 設定内容と“出力例” | 0x00011953 - ゲイン1.583 = 6483[0x1953] / 4096 |

#### 3.7.2. オフセット

カメラのオフセットはオフセットレジスタで操作します。

|            |                              |
|------------|------------------------------|
| アドレス       | 0xf000404c                   |
| 長さ (DWord) | 1                            |
| アクセスタイプ    | RW                           |
| 設定内容と“出力例” | 0x00000043 - オフセット 67 = 0x43 |



### 3.7.3. カラーゲイン

カメラのカラー補正値はカラーゲインレジスタで操作します。値は16bitの少数点で指定し、4bitの整数値と12bitの少数値で構成されます。

|                   |   |
|-------------------|---|
| <b>アドレス</b>       | 0xf0004054 赤 (R)<br>0xf0004058 緑 (RG)<br>0xf000405c 緑 (BG)<br>0xf0004060 青 (B)  |
| <b>長さ (DWord)</b> | 1   |
| <b>アクセスタイプ</b>    | RW  |
| <b>設定内容と"出力例"</b> | 0xf0004054 - 0x00002000 = 8192 / 4096 -> 2.0<br>0xf0004058 - 0x00001400 = 5120 / 4096 -> 1.25<br>0xf000405c - 0x00001200 = 4608 / 4096 -> 1.125<br>0xf0004060 - 0x00001800 = 6144 / 4096 -> 1.5 |

## 3.8. LUT

### 3.8.1. 基本仕様

LUTは置換テーブルを表す追加のメモリ領域です。ここではこのLUTを有効化するか無効化するかを制御します。LUTを有効化する場合レジスタ0xf0004000のbit[19]を1に変更します。0にする事で無効化します。

|            |  |
|------------|--|
| アドレス       | 0xf0004000 - bit[19]のみを操作  |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と“出力例” | 0x0000020d (bit[19] = 0) - LUT無効化<br>0x0008020d (bit[19] = 1) - LUT有効化 |

#### 注意

このレジスタには、他にも重要なカメラ設定の値（シーケンサーやLUT等）を含みます。  
LUTの無効化と有効化を行うために必ず現在値を確認し、bit[19]の値以外は変更しないで下さい。

置換テーブルはアドレス0xf0010000から始まり、一つのテーブル毎に12bitの輝度値が設定されており、各アドレス毎に置換用の値が格納されます。

例えばアドレス0xf0010000は輝度値0の置換テーブルを示し、アドレス0xf0010004は輝度値1の置換テーブルを示します。

これは以下の式で表します。

$$\text{アドレス値} = 0xf0010000 + 4 * \text{輝度値}$$

|            |   |
|------------|---|
| アドレス       | 0xf0000000 ~ 0xf0013ffc   |
| 長さ (DWord) | 1<br>(アドレス域には4096個のDwordsの変更値を含みます)   |
| アクセスタイプ    | RW  |
| 設定内容と“出力例” | アドレス : 0xf0010000<br>データ : 0x00000003 -> 輝度値0を輝度値3に置換<br><br>アドレス : 0xf0010004<br>データ : 0x00000002 -> 輝度値1を輝度値2に置換<br>データ : 0x00000FFF -> 輝度値1を輝度値4095に置換 |

#### 注意

カメラは12bitで動作するので最も低い12bitの値だけが参照されます。  
従って、有効な値の範囲は0x000~0xFFFFまでです。

### 3.8.2. ガンマ補正

カメラはLUTの値を自動的にガンマ値の値に設定する機能を提供します。  
これによりガンマ1.0とは異なるガンマ値がレジスタに書き込まれます。  
ガンマ値レジスタは8bitの小数点で指定し、1bitの整数値と7bitの少数値で構成されています。  
ガンマ値は0.5から2.0の範囲で設定可能です。

#### 注意

ガンマ補正はLUTが有効な時にのみ利用可能です。

|            |   |
|------------|---|
| アドレス       | 0xf0004068  |
| 長さ (DWord) | 1   |
| アクセスタイプ    | RW  |
| 設定内容と"出力例" | 0x00000080 = 128 / 128 -> 1.0 (ガンマ補正無効)<br>0x000000c0 = 192 / 128 -> 1.5<br>0x00000040 = 64 / 128 -> 0.5<br>0x00000057 = 87 / 128 -> 0.68 |

### 3.9. 欠陥画素補正

どのような製造元であってもいくつかの問題のある画素（欠陥画素）がセンサー上には存在します。これらの画素の座標が欠陥画素リストには格納されています。

リストはBaumerカメラの製造時に作成され、工場出荷設定にカメラへ格納されています。ただし、このリストは編集が可能です。

リストに掲載された欠陥画素は補正により目立たなくなります。  
この手法を欠陥画素補正と呼び、以下の方法で実行されます。

- 補正を行う前に欠陥画素に隣接する左右の2画素のデータを読み出します
- それら2画素の平均値を算出します
- 最後に欠陥画素の輝度値に先ほど算出した値を代入します

欠陥画素補正を有効化するにはアドレス0xf0004000のbit[11]を1に設定します。

#### 有効化と無効化

|            |  |
|------------|--|
| アドレス       | 0xf0004000 - bit[11]のみを操作                  |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と"出力例" | 有効化<br>0x02000A0D<br><br>無効化<br>0x0200020D |

## 欠陥画素リストインデックス

|            |   |
|------------|---|
| アドレス       | 0xf0020004 + (Index * 4)  |
| 長さ (DWord) | 1   |
| アクセスタイプ    | RW  |
| 設定内容と“出力例” | 欠陥画素補正の有効化/無効化 - bit[31]<br>欠陥画素補正のX座標 - bit[0~10]<br>欠陥画素補正のY座標 - bit[12~23]<br><br>補正1箇所目 : Index = 0、X = 10、Y = 10、有効<br>アドレス : 0xf0020004 + (0*4) = 0xf0020004<br>設定値 : 0x8000A00A<br><br>補正2箇所目 : Index = 1、X = 2047、Y = 2047、無効<br>アドレス : 0xf0020004 + (1*4) = 0xf0020008<br>設定値 : 0x007FF7FF |

## 3.10. シーケンサー

### 3.10.1. 概要

BaumerHXCカメラシリーズは異なるパラメーターセットで連続した画像を取得するシーケンサー機能を有しています。その詳細に関してはユーザーガイドを確認して下さい。

シーケンサーで使用するパラメーターセットには露光時間 (ExposureTime)、ゲイン (Gain)、デジタルIO (出力) などを利用できます。設定は3つの異なるグループに分けられており、設定レジスタがそれぞれ連なってメモリ領域に格納されています。

シーケンサーを使用するためにはカメラのマスター領域で機能を有効化します。シーケンサーが有効化されると、機能により画像の取得と読み出し処理が制御可能になります。露光時間を変更する機能などをそれ以降は操作しないで下さい。

シーケンサーの有効化にはアドレス0xf0004000のbit[24]を1に設定します。0にする事で無効化します。

|            |  |
|------------|--|
| アドレス       | 0xf0004000 - bit[24]のみを操作  |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と“出力例” | 0x0200020d (bit[24] = 0) - シーケンサー無効化<br>0x0300020d (bit[24] = 1) - シーケンサー有効化 |

### 注意

このレジスタには、他にも重要なカメラ設定の値 (シーケンサーやLUT等) を含みます。シーケンサーの有効化と無効化を行うために必ず現在値を確認し、bit[24]の値以外は変更しないで下さい。

### 3.10.2. 設定領域

#### 基本設定領域

シーケンサーのメインの設定領域は0xf0009000です。

ここには以下のようにいくつかの異なる設定が格納されています。

|                  |  |
|------------------|--|
| <b>Bit[0]</b>    | <ul style="list-style-type: none"> <li>・スタートビット</li> </ul> <p>ここではシーケンサーの開始を制御します。<br/>bit[0]が0ならシーケンサーは停止し、内部ステータスはリセットされます。0xf0004000のbit[24]が有効化されている必要があります。</p>   |
| <b>Bit[4~7]</b>  | <p>モードビット</p> <p>シーケンサーの動作モードを4bitで定義します。<br/>以下のモードが利用可能です。</p> <p>0001 (0x1) モード：トリガー毎に1枚（連続撮影）<br/>各トリガ毎にシーケンサーはZ枚の画像を取得します。<br/>Z枚の画像は1回のトリガに応じてフリーランで取得されます。全ての撮影サイクルが終了しても再び始めのサイクルに戻り撮影を続けます。</p> <p>0010 (0x2) モード：トリガー毎に1枚（1サイクル）<br/>各トリガ毎にシーケンサーはZ枚の画像を取得します。<br/>Z枚の画像は1回のトリガに応じてフリーランで取得されます。全ての撮影サイクルが終了するとシーケンサーは停止します。</p> <p>0011 (0x3) モード：フリーラン（連続撮影）<br/>シーケンサーはトリガを検知せずフリーランで撮影を続けます。</p> <p>0101 (0x5) モード：トリガで開始するフリーラン（連続撮影）<br/>検知したトリガに応じてシーケンサーはフリーランの撮影を開始します。全てのサイクルが終わった後シーケンサーは次の撮影のためトリガを再度待機します。</p> <p>0110 (0x6) モード：トリガで開始するフリーラン（1サイクル）<br/>検知したトリガに応じてシーケンサーはフリーランの撮影を開始します。全てのサイクルが終わった後シーケンサーは停止します。</p> |
| <b>Bit[8~14]</b> | <ul style="list-style-type: none"> <li>・パラメーターセットの数</li> </ul> <p>この値はシーケンサーで使用するパラメーターセットの数を定義します。値は0から始まるため1つのセットのみの場合値は0x00にします。セットの数が2の場合0x01で、3の場合は0x02です。</p>  |

|                   |  |
|-------------------|--|
| <b>アドレス</b>       | 0xf0009000   |
| <b>長さ (DWord)</b> | 1  |
| <b>アクセスタイプ</b>    | RW   |
| <b>設定内容と"出力例"</b> | <p>0x00000000 - シーケンサー停止</p> <p>0x00000021 - シーケンサー開始<br/>モード：トリガー毎に1枚（1サイクル）<br/>パラメーターセット数：1</p> <p>0x00000131 - シーケンサー開始<br/>モード：フリーラン（連続撮影）<br/>パラメーターセット数：2</p> |

### ループ制御レジスタ (M)

このレジスタは全てのセットを何回ループするかを指定できます。  
Mの値は1~4294967295まで指定可能です。

|            |  |
|------------|--|
| アドレス       | 0xf0009004                               |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW                                       |
| 設定内容と"出力例" | 0x00000001 - M = 1<br>0x00000004 - M = 4 |

### トリガー毎のフレーム枚数 (Z)

このレジスタは一つのトリガーイベントで何枚続けて画像を取得するかを指定できます。  
Zの値は1~4294967295まで指定可能です。

|            |  |
|------------|--|
| アドレス       | 0xf0009008                                       |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と"出力例" | 0x00000001 - 1トリガー毎に1枚<br>0x00000007 - 1トリガー毎に7枚 |

### 注意

フリーランモードでシーケンサーを動作させている場合、Zの値は無視されます。

### 3.10.3. ステータス

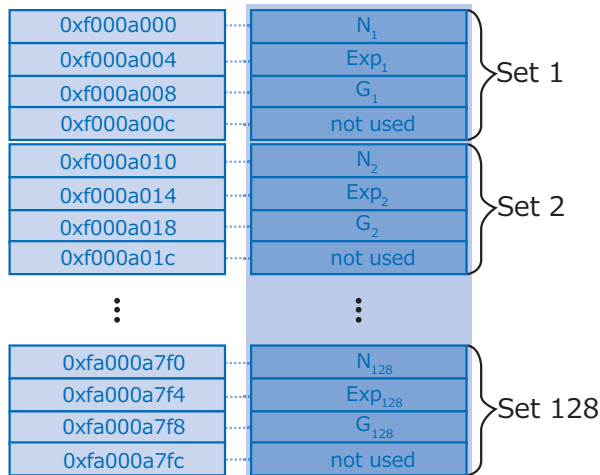
シーケンサーの現在地を取得する場合があります。もしかかもしれません。  
ステータスレジスタの0xf0009010から現在値を読み出す事が可能です。  
読み出した値は以下の構造になっています。

|            |  |
|------------|--|
| bit [0]    | ・シーケンサーの動作状態<br>このビットが1ならシーケンサーは動作中です。このbitはシーケンサーが停止するか所定のサイクルを終了すると自動的に0になります。   |
| bit [8~14] | ・現在のパラメーターセット番号<br>このレジスタには現在のパラメーターセット番号を含みます。<br>ここではカウントが0から始まります。例えば、0x01の結果が返ってきた場合パラメーターセットは2番を使用しています。<br>0x02の場合は3番を使用しています。 |

|            |  |
|------------|--|
| アドレス       | 0xf0009010   |
| 長さ (DWord) | 1  |
| アクセスタイプ    | RW   |
| 設定内容と"出力例" | 0x00009000 - シーケンサーは停止中です<br>0x00000201 - シーケンサーは動作中で、現在3番のパラメーターセットを使用しています |

### 3.10.4. パラメーターセット

パラメーターセットは連続したメモリ領域に格納されています。各パラメーターは4つに分かれたDWordで構成されています。メモリ領域の構造は以下の通りです。



各パラメーターは以下の情報を定義しています。

| 番号 | 機能          | 内容  |
|----|-------------|---|
| 0  | N           | このパラメーターセットで何枚のフレームを取得するかを32bit値で指定します。範囲は1~4294967295の値です。フリーランかトリガーモードかによってZの設定の影響を受けます。  |
| 1  | 露光時間 (Exp)  | このパラメーターセットでの露光時間を32bit値で指定します。有効な範囲はカメラに依存します。ユーザーガイドをご確認下さい。  |
| 2  | ゲイン (G) とIO | <p>Bit [0~15] このセットでのゲイン値を指定します。この値は4bitの整数値と12bitの少数値で構成され、ゲインの設定と同じ構造です。</p> <p>Bit [16~18] このセットで使用する3つの異なるシーケンサー出力の値を指定します。これらを出力ラインのソースとして使用する場合、SequencerOutput0~2の設定をラインソースに指定します。</p> |
| 3  | 未使用         | -   |

|                   |                         |   |
|-------------------|-------------------------|---|
| <b>アドレス</b>       | 0xf000a000 ~ 0xf000a7fc |   |
| <b>長さ (DWord)</b> | 4 (最大512 : 128セット分)     |   |
| <b>アクセスタイプ</b>    | RW                      |   |
| <b>設定内容と“出力例”</b> | パラメーターセット1              |   |
|                   | 0xf000a000              | = 0x00000001 N = 1<br>1フレーム取得                               |
|                   | 0xf000a004              | = 0x00000fa0 露光時間 = 4000 us                                 |
|                   | 0xf000a008              | = 0x00011000 ゲイン = 1.0 / IO = “001”<br>(SequencerOutput0=1) |
|                   | 0xf000a00c              | = 未使用 -   |
|                   | パラメーターセット1              |   |
|                   | 0xf000a010              | = 0x00000002 N = 3<br>3フレーム取得                               |
|                   | 0xf000a014              | = 0x00001770 露光時間 = 6000 us                                 |
|                   | 0xf000a018              | = 0x00021800 ゲイン = 1.5 / IO = “010”<br>(SequencerOutput1=1) |
|                   | 0xf000a01c              | = 未使用 -   |
|                   | . . .                   |   |



## 3.11. IO制御、トリガーとタイマー

### 3.11.1 IO制御

BaumerHXCカメラは複数の入力と出力のIO制御ラインが利用可能です。各ラインではいくつかの設定項目があります。出力ライン用の信号は固定化されておらず変更が可能です。詳細な内容はユーザーガイドをご確認下さい。

下記の表は利用可能なラインとライン番号です。最初の入カラインはライン0から割り当てられており、最初の出カラインはライン7から割り当てられています。

| ライン番号  | ライン                  |
|--------|----------------------|
| 0 から 2 | 外部入カライン              |
| 3 から 6 | フレームグラバからの入カライン (CC) |
| 7 から 9 | 外部出カライン              |

ライン設定はアドレス0x80002000からつらなる連続したメモリ領域に格納されています。各ラインには5つのDword値による設定が存在します。各ラインは以下のメモリ領域に割り当てられています。

|                         |                 |                      |
|-------------------------|-----------------|----------------------|
| 0x80002000 ~ 0x80002010 | Line 0          |                      |
| 0x80002014 ~ 0x80002024 | Line 1          | 外部入カライン              |
| 0x80002028 ~ 0x80002038 | Line 2          |                      |
| 0x8000203c ~ 0x8000204c | Frame Grabber 0 | フレームグラバからの入カライン (CC) |
| 0x80002050 ~ 0x80002060 | Frame Grabber 1 |                      |
| 0x80002064 ~ 0x80002074 | Frame Grabber 2 |                      |
| 0x80002078 ~ 0x80002088 | Frame Grabber 3 |                      |
| 0x8000208c ~ 0x8000209c | Line 7          |                      |
| 0x800020a0 ~ 0x800020b0 | Line 8          | 外部出カライン              |
| 0x800020b4 ~ 0x800020c4 | Line 9          |                      |

以下の表は設定エリアの内容が格納されています。各値の意味はラインによって異なります。例では最初の入カライン (Line 0) と最初の出カライン (Line 7) を表記しています。

| アドレス               | 入カライン   | 出カライン                  |
|--------------------|---|------------------------|
| Line0 : 0x80002000 | 反転設定 : 入力の場合検知するレベルが反転し、出力の場合には出力レベルが反転します。bit[0]が0でOff、1でOnです。 | ラインのソース信号設定 : 上位32bit値 |
| Line7 : 0x8000208c |   |                        |
| Line0 : 0x80002004 | bit[0~31] : デバウンサーLowの監視時間 (us単位)                               | ラインのソース信号設定 : 下位32bit値 |
| Line7 : 0x80002090 |   |                        |
| Line0 : 0x80002008 | bit[0~31] : デバウンサーHighの監視時間 (us単位)                              |                        |
| Line7 : 0x80002094 |   |                        |
| Line0 : 0x8000200c |   |                        |
| Line7 : 0x80002098 |   |                        |
| Line0 : 0x80002010 |   |                        |
| Line7 : 0x8000209c |   |                        |

カメラはいくつかの内部信号を外部出力ラインに割り当てる事で、トリガー信号のソースやタイマーイベントとして利用できます。これらの信号を“ラインソース”と呼びます。

以下の表では利用可能なラインソースとどのように指定するかを記載しています。

各ラインソースは64bitの設定値を持ちそれぞれ32bitの2つの領域に分離しています。出力ラインやトリガーやタイマーイベントとしてラインソースを割り当てる場合、上位パートと下位パートの両方の値をソースレジスタに設定しなければなりません。

| 名前 (ラインソース)        | 設定値 (64bit) |            | 利用可能な<br>ラインソース設定 |      |      |
|--------------------|-------------|------------|-------------------|------|------|
|                    | 上位32bit     | 下位32bit    | IO<br>ライン         | トリガー | タイマー |
| Off                | 0x00000000  | 0x00000000 | ■                 | ■    | ■    |
| Line0              | 0x00000001  | 0x00000000 | ■                 | ■    | ■    |
| Line1              | 0x00000002  | 0x00000000 | ■                 | ■    | ■    |
| Line2              | 0x00000004  | 0x00000000 | ■                 | ■    | ■    |
| FrameGrabber Line0 | 0x00000008  | 0x00000000 | ■                 | ■    | □    |
| FrameGrabber Line1 | 0x00000010  | 0x00000000 | ■                 | ■    | □    |
| FrameGrabber Line2 | 0x00000020  | 0x00000000 | ■                 | ■    | □    |
| FrameGrabber Line3 | 0x00000040  | 0x00000000 | ■                 | ■    | □    |
| Frameactive        | 0x00000000  | 0x00000003 | ■                 | □    | □    |
| TriggerReady       | 0x00000000  | 0x00000004 | ■                 | □    | □    |
| TriggerOverlapped  | 0x00000000  | 0x00000008 | ■                 | □    | □    |
| TriggerSkipped     | 0x00000000  | 0x00000010 | ■                 | □    | ■    |
| ExposureActive     | 0x00000000  | 0x00000001 | ■                 | □    | □    |
| TransferActive     | 0x00000080  | 0x00000000 | ■                 | □    | □    |
| Useroutput0        | 0x00020000  | 0x00000000 | ■                 | □    | □    |
| Useroutput1        | 0x00040000  | 0x00000000 | ■                 | □    | □    |
| Useroutput2        | 0x00080000  | 0x00000000 | ■                 | □    | □    |
| Timer1Active       | 0x00100000  | 0x00000000 | ■                 | □    | □    |
| SequencerOutput0   | 0x00000000  | 0x00020000 | ■                 | □    | □    |
| SequencerOutput1   | 0x00000000  | 0x00040000 | ■                 | □    | □    |
| SequencerOutput2   | 0x00000000  | 0x00080000 | ■                 | □    | □    |
| SW-Trigger         | 0x00002000  | 0x00000000 | □                 | ■    | ■    |
| Exposure Start     | 0x00800000  | 0x00000000 | □                 | □    | ■    |
| Exposure End       | 0x01000000  | 0x00000000 | □                 | □    | ■    |
| Frame Start        | 0x02000000  | 0x00000000 | □                 | □    | ■    |
| Frame End          | 0x04000000  | 0x00000000 | □                 | □    | ■    |

例えば、ExposureActiveを外部出力ラインのLine7に割り当てる場合、0x80002090に0x00000000を設定し、0x80002094に0x00000001を設定します。

## ラインステータス

この機能は全てのラインのステータスを一度に取得可能です。

|         |               |
|---------|---------------|
| bit [0] | Line 0 のステータス |
| bit [1] | Line 1 のステータス |
| bit [2] | Line 2 のステータス |
| bit [3] | Line 3 のステータス |
| bit [4] | Line 4 のステータス |
| bit [5] | Line 5 のステータス |
| bit [6] | Line 6 のステータス |
| bit [7] | Line 7 のステータス |
| bit [8] | Line 8 のステータス |
| bit [9] | Line 9 のステータス |

|                   |   |
|-------------------|---|
| <b>アドレス</b>       | 0x80001408  |
| <b>長さ (DWord)</b> | 1   |
| <b>アクセスタイプ</b>    | R   |
| <b>設定内容と"出力例"</b> | 0x0000001a = 00011010 (2進数)<br>Line1 = On<br>Line3 = On<br>Line4 = On |

## ユーザー出力

IO制御は最大3つまでユーザーが任意に指定可能な出力制御を提供します。

ユーザー出力を特定の出力ラインに割り当てる場合、指定のラインのラインソースレジスタに設定値を与えなければなりません。

例えば、UserOutput0を出力ラインLine9に割り当てる場合、0x800020b8に0x00020000を指定し、0x800020bcに0x00000000を指定します。

0x8000140cのユーザー出力レジスタでは3つ全てのユーザー出力の現在値が下位3bitに格納されています。

| <b>0x8000140c<br/>のbit値</b> | <b>名前</b>   |
|-----------------------------|-------------|
| bit [0]                     | UserOutput0 |
| bit [1]                     | UserOutput1 |
| bit [2]                     | UserOutput  |

|                   |   |
|-------------------|---|
| <b>アドレス</b>       | 0x8000140c  |
| <b>長さ (DWord)</b> | 1   |
| <b>アクセスタイプ</b>    | RW  |
| <b>設定内容と"出力例"</b> | 00000003 = (0/1/1)<br>UserOutput2 = Off<br>UserOutput1 = On<br>UserOutput0 = On |

### 3.11.2. トリガー

BaumerHXCカメラはフリーラン以外に、3つの異なるトリガーモードをそれぞれ使用することが可能です。トリガーモードを使用するにはアドレス0xf0004034の設定レジスタをはじめに有効にしなければなりません。詳細な情報はユーザーガイドのトリガーモードをご確認下さい。

|                   |  |
|-------------------|--|
| <b>アドレス</b>       | 0xf0004034   |
| <b>長さ (DWord)</b> | 1  |
| <b>アクセスタイプ</b>    | RW   |
| <b>設定内容と"出力例"</b> | 0xf0004034 = 0x00 - フリーラン<br>0xf0004034 = 0x04 - トリガーモード<br>0xf0004034 = 0x0c - トリガーモード<br>(パルス長露光制御)<br>0xf0004034 = 0x14 - トリガーモード<br>(エッジ間露光制御) |

カメラのトリガーイベントに使用するラインソースは自由に設定可能で、0x80001000のレジスタにラインソースの上位32bitの値を指定し、0x80001004のレジスタに下位32bitの値を指定します。ラインソース設定に使用する値はページ42の表をご確認下さい。

|                   |  |
|-------------------|--|
| <b>アドレス</b>       | 0x80001000 - ラインソースの上位32bit の値<br>0x80001004 - ラインソースの下位32bit の値               |
| <b>長さ (DWord)</b> | 1  |
| <b>アクセスタイプ</b>    | RW   |
| <b>設定内容と"出力例"</b> | Line0をトリガー制御のソースに使用する場合：<br>0x80001000 = 0x00000001<br>0x80001004 = 0x00000000 |

#### 注意

トリガーソースを指定する前に必ずトリガーモードを有効にして下さい。

#### トリガー遅延

入力されたトリガーイベントは、トリガー遅延レジスタによって任意の遅延を与える事が可能です。遅延時間はus単位です。詳細な情報はユーザーガイドのトリガー遅延をご確認下さい。

|                   |                          |
|-------------------|--------------------------|
| <b>アドレス</b>       | 0x80001008               |
| <b>長さ (DWord)</b> | 1                        |
| <b>アクセスタイプ</b>    | RW                       |
| <b>設定内容と"出力例"</b> | 0x000007d0 - 遅延時間2000 us |

## トリガー極性

トリガーモードでは、カメラは立上りエッジ (RisingEdge) と立下がりエッジ (FallingEdge) に反応します。どちらに反応するかはレジスタ0x8000100cで設定します。

|            |   |
|------------|---|
| アドレス       | 0x8000100c  |
| 長さ (DWord) | 1   |
| アクセスタイプ    | RW  |
| 設定内容と"出力例" | 0x00000000 - Falling Edge<br>0x00000001 - Rising Edge |

## ソフトウェアトリガー

### 注意

ソフトウェアトリガーを使用する場合ソフトウェアトリガーのソースを最初に設定して下さい。

ソフトウェアトリガーは特別なソースのトリガー方法で、PCで実行しているソフトウェアから画像取得を制御できます。ソフトウェアトリガーを使用する場合、はじめに0x80001000のレジスタに0x00002000を指定し、0x80001004のレジスタに0x00000000を指定して、ラインソースをソフトウェアトリガーに設定しなければなりません。

|            |                         |
|------------|-------------------------|
| アドレス       | 0x80001100              |
| 長さ (DWord) | 1                       |
| アクセスタイプ    | RW                      |
| 設定内容と"出力例" | 何らかの値を設定するとトリガーが発行されます。 |

## 3.11.3. タイマー制御

BaumerHXCカメラは検知したイベントに応じて指定幅のパルスを発行するタイマー機能を利用可能です。開始イベントやパルス幅、パルス遅延などが自由に設定可能です。ページ42の表では開始イベントに指定可能なラインソースが記載されています。

タイマー設定はアドレス0x80003400から始まる連続したメモリ領域に格納されています。設定内容は以下の通りです。

| アドレス       | 内容   |
|------------|--|
| 0x80003400 | タイマーイベント設定：<br>0x00000000 - タイマー機能OFF<br>0x00000001 - Rising Edge, Falling Edgeの両方に反応<br>0x00000002 - Rising Edgeに反応<br>0x00000003 - Falling Edgeに反応 |
| 0x80003404 | タイマーソース設定：<br>ラインソースの上位32bit値  |
| 0x80003408 | タイマーソース設定：<br>ラインソースの下位32bit値  |
| 0x80003410 | 遅延時間 (us単位)  |
| 0x80003414 | パルス幅 (us単位)  |

生成したパルスを使用するには出力ラインのソースを指定するレジスタでTimer1Acticeのラインソースを割り当てなければなりません。

## 3.12. カメラステータスと情報

### 3.12.1. センサー温度

HXCカメラに使用されているセンサーにはセンサー温度を計測する機能があります。現在値を取得するには温度レジスタw0読みだして下さい。値は16bitの小数で出力され、bit[15]に符号データが、bit[8~14]に整数値が、bit[0~7]に少数値が格納されています。

#### 注意

温度は摂氏 (°C) で提供されます。(ただし、キャリブレーションされていません)

|            |   |
|------------|---|
| アドレス       | 0xf0004100  |
| 長さ (DWord) | 1   |
| アクセスタイプ    | R   |
| 設定内容と“出力例” | 0x00003780 - bit[0~14] 0x3780 = 14208/256 -> 55.5°C |

### 3.12.2. フレームカウンター

フレームカウントレジスタでカメラが出力したフレーム数を取得できます。このレジスタは0を書き込むことで値をリセットできます。

|            |                   |
|------------|-------------------|
| アドレス       | 0xf0004080        |
| 長さ (DWord) | 1                 |
| アクセスタイプ    | RW                |
| 設定内容と“出力例” | 0x00000112 - 274枚 |

### 3.12.3. 現在の画素数

画像サイズレジスタに画像数の現在値が格納されています。ROIによる変更だけではなくサブサンプルやビニングによる画素数の変更にも連動します。

このレジスタは以下の16bit整数値を含みます。

|        |            |
|--------|------------|
| X方向画素数 | bit[0~15]  |
| Y方向画素数 | bit[16~31] |

|            |  |
|------------|--|
| アドレス       | 0xf0004088                                     |
| 長さ (DWord) | 1  |
| アクセスタイプ    | R  |
| 設定内容と“出力例” | 0x08000600 - 0x0800 -> Y:2048、0x0600 -> X:1536 |

### 3.12.4. 読み出し時間

センサーの読み出し時間はセンサー読み出し速度やピクセルフォーマット、ROIなどに依存します。読み出し時間レジスタではカメラの現在の読み出し時間をμs単位で提供します。

|            |                      |
|------------|----------------------|
| アドレス       | 0xf0004400           |
| 長さ (DWord) | 1                    |
| アクセスタイプ    | R                    |
| 設定内容と“出力例” | 0x080019b4 - 6580 μs |





■ Baumer Optronik GmbH  
Badstrasse 30  
DE-01454 Radeberg, Germany  
Tell : +49 (0)3528 4386 0  
Fax : +49 (0)3528 4386 86  
Mail : sales@baumeroptronic.com  
URL : <http://www.baumer.com/>

■ 株式会社アルゴ  
〒564-0063 大阪府吹田市江坂町1-13-48  
インタープラネット江坂ビル9F  
Tell : 06-6339-3366  
Fax : 06-6339-3365  
Mail : argo@argocorp.com  
URL : <http://www.argocorp.com>