NT96660 Sensor Porting

Table of Content

1 Introduction

新增一个 sensor driver，需要修改或新增相关参数及档案主要在下列档案中(以 IMX078CQK 为例)，此份文件主要会说明 sensor driver 的部分(.\DrvExt\Sensor\CMOS_ IMX078CQK)，要如何填写 sensor 相关参数。

1. .\DrvExt\Sensor\CMOS_IMX078CQK (新增整份文件夹)
   (1) IMX078CQK_param_Int.h
   (2) IMX078CQK_param.c
   (3) IMX078CQK.c
2. . \DrvExt\DrvExt_src\ModelExt\FIREFLY (若有要另外拉一个新的 model 才需要新增)
   (1) DxCamera_Sensor.c (修改)
   (2) PinmuxCfg.c (修改)
   (3) 此部分请参考文件 *NT96660 Sensor Driver Application Note* 做修改
3. . \Include\DrvExt\SENSOR (修改)
   (1) SensorDrv.h
4. .\LibExt\LibExt_src\
   (1) .\LibExt\LibExt_src\DevCamIPL\IPL_IMX078CQK_EVB_FF (新增整份文件夹)
   (2) .\LibExt\LibExt_src\PluginPhoto\AE_IMX078CQK_EVB_FF (新增整份文件夹)
   (3) .\LibExt\LibExt_src\PluginPhoto\AWB_IMX078CQK_EVB_FF (新增整份文件夹)
   (4) 此部分请参考文件 *NT96660 Add a New Extend_IPL* 做修改
5. .\Project\DemoKit
   (1) MakeConfig.txt (修改)
   (2) ModelConfig_FIREFLY.txt (修改)

2 Driver Ext

2.1 Sensor driver – LVDS (IMX078CQK)

档案位置：.\DrvExt\Sensor\CMOS_IMX078CQK

1. IMX078CQK_param_Int.h
   (1) 换算 sensor gain 相关参数，根据 sensor 特性决定，在下列 function 当中做换算：
   - 档案：IMX078CQK.c
   - function：GetGainSetting_IMX078CQK() & SetAGC_ISOBase_IMX078CQK()

```
#define ISOBASE          50
#define AGC_ISOBASE      1000
```

   - ISOBASE：ISO 基准值，通常设为 50，或是 sensor 最低可以接受的 ISO 值。
   - AGC_ISOBASE：校正基准值，根据 sensor 特性决定。

   (2) Sensor mode 数量，根据需要接几个 sensor mode 给定 define

```
#define MODE_1        SENSOR_MODE_1
#define MODE_2        SENSOR_MODE_2
#define MODE_3        SENSOR_MODE_3
#define MODE_4        SENSOR_MODE_4
#define MODE_5        SENSOR_MODE_5
#define MODE_6        SENSOR_MODE_6
#define MODE_7        SENSOR_MODE_7
#define MODE_8        SENSOR_MODE_8
#define MODE_9        SENSOR_MODE_9
#define MODE_10        SENSOR_MODE_10
#define MODE_MAX                10
```

2. IMX078CQK_param.c
   (1) Sensor LVDS sync signal (12bit 为例)

```
typedef struct
{
    UINT32 Num;              ///< sync code total number
    UINT32 Code[7];          ///< sync code
} SENSOR_LVDS_SYNC_CODE_PTN;
```

```
static SENSOR_LVDS_SYNC_CODE_PTN SyncCode_IMX078CQK_12Bit =
{
    3,
    {0xFFF, 0x000, 0x000, 0x000, 0x000, 0x000, 0x000},
};
```

   - Num：sync code 有几个 word，最多可以有 7 个。

- Code[7]：同步讯号，填前面共同的部分即可。
- 数据参考来源：Sensor spec

Sync code details (hexadecimal notation) 12-bit output

		1st word	2nd word	3rd word	4th word
Blanking line	Start sync code (SAV)	FFFh	000h	000h	AB0h
	End sync code (EAV)				B60h
Except blanking line	Start sync code (SAV)				800h
	End sync code (EAV)				9D0h

(2) Sensor LVDS ctrl signal (12bit 为例)

LVDS 用来判断 VD/HD 的 start 及 end。

```
typedef struct
{
    UINT32 CtrlHD;          ///< lvds ctrl words
    UINT32 MaskHD;          ///< lvds mask
    UINT32 CtrlVD;          ///< lvds ctrl words
    UINT32 MaskVD;          ///< lvds mask
} SENSOR_LVDS_CTRLPTN;
```

```
static SENSOR_LVDS_CTRLPTN RecogCode_IMX078CQK_12Bit [LVDS_DATLANE_ID_MAX]=
{
    {0x800, 0xFFF, 0xAB0, 0xFFF},
    {0x800, 0xFFF, 0xAB0, 0xFFF},
    {0x800, 0xFFF, 0xAB0, 0xFFF},
    {0x800, 0xFFF, 0xAB0, 0xFFF},
    {0x800, 0xFFF, 0xAB0, 0xFFF},
    {0x800, 0xFFF, 0xAB0, 0xFFF},
    {0x800, 0xFFF, 0xAB0, 0xFFF},
    {0x800, 0xFFF, 0xAB0, 0xFFF},
    {0x800, 0xFFF, 0xAB0, 0xFFF},
    {0x800, 0xFFF, 0xAB0, 0xFFF},
};
```

- CtrlHD：Except blanking line / Start sync code / 4th word
- MaskHD：1st word
- CtrlVD：Blanking line / Start sync code / 4th word
- MaskVD：1st word
- 数据参考来源：Sensor spec

Sync code details (hexadecimal notation) 12-bit output

		1st word	2nd word	3rd word	4th word
Blanking line	Start sync code (SAV)	FFFh	000h	000h	AB0h
	End sync code (EAV)				B60h
Except blanking line	Start sync code (SAV)				800h
	End sync code (EAV)				9D0h

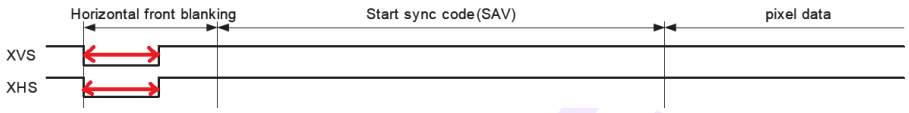
(3) Sensor LVDS information (mode 0 为例)

```
typedef struct
{
    SENSOR_SIGNAL XHS;                  ///< HD length for sony sensor only.
    SENSOR_SIGNAL XVS;                  ///< VD length for sony sensor only.
    UINT32 Width;                       ///< data output width
    UINT32 Height;                      ///< data output height
    UINT32 LaneNum;                     ///< data lanes number
    UINT32 BitDepth;                    ///< data bits
    UINT32 DataAlign;                   ///< data MSB/LSB
    UINT32 OutputPixelOrder[10];        ///< output data pixel order
} SENSOR_LVDS;
```

```
static SENSOR_LVDS LVDS_IMX078CQK[MODE_MAX + 1] =
{
    {//mode 0
        {8, 1170, 0, 0},
        {8, 3080, 0, 0},
        4168,
        3060,
        LVDS_DATLANE_8,
        LVDS_PIXDEPTH_12BIT,
        LVDS_DATAIN_BIT_ORDER_MSB,
        {0, 1, 2, 4, 5, 7, 8, 9, SEN_IGNORE, SEN_IGNORE},
    },
```

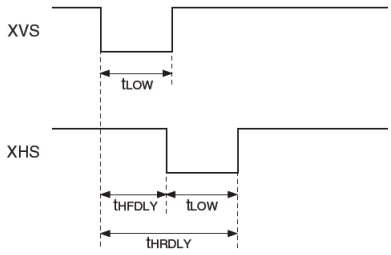
- XHS/ XVS：

  Sync：同步讯号 pulse 宽度。



数据参考来源：Sensor spec

XVS Low level pulse width	tLOW	4/fINCK	—	12/fINCK	ns
XHS Low level pulse width	tLOW	4/fINCK	—	12/fINCK	ns



Period：一个 XHS 有几个 input clock (MCLK) / 一个 XVS 有几个 XHS，增加 XVS period 可以达到降低 frame rate 的效果

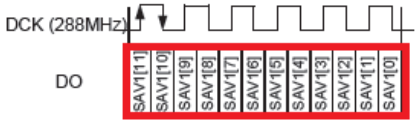
1170[INCK]/1[XHS] = 1170

3080[XHS]/1[XVS] = 3080

- Width/ Height：LVDS crop 的宽度及高度，也就是 sensor 吐出来所有 data 的宽度及高度。
- LaneNum：sensor 用几 lane 来出 data，也就是同时有几个 channel 在出 data。
- BitDepth：sensor 出的 data bit 数。
- DataAlign：data lane 的排序方式。

  MSB：data 的 bit 由大排到小。

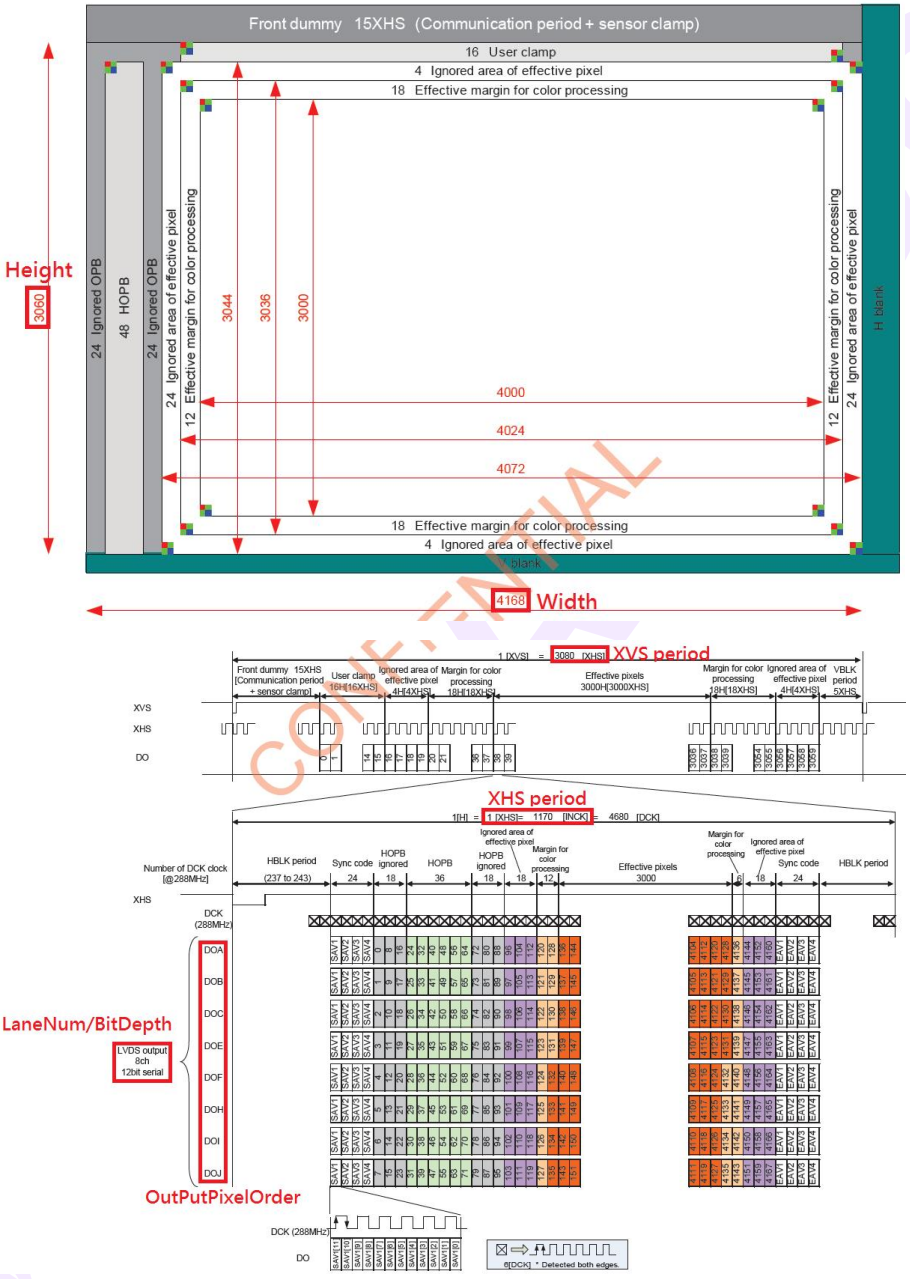
  LSB：data 的 bit 由小排到大。



- OutputPixelOrder[10] ：sensor 用哪几个 channel 来出 data，对照表如下

DOA	DOB	DOC	DOD	DOE	DOF	DOG	DOH	DOI	DOJ
0	1	2	3	4	5	6	7	8	9

- 数据参考来源：Sensor spec



[DCK]：data clock

(4) Sensor signal information (for AE)

GetExpoSetting_IMX078CQK()：用来换算曝光时间和设定给 sensor 曝光参数的关系性。 P.S. sony sensor 直接用 LVDS 讯号来计算，所以也可以采用 LVDS 当中 XHS/XVS 的 period 做换算

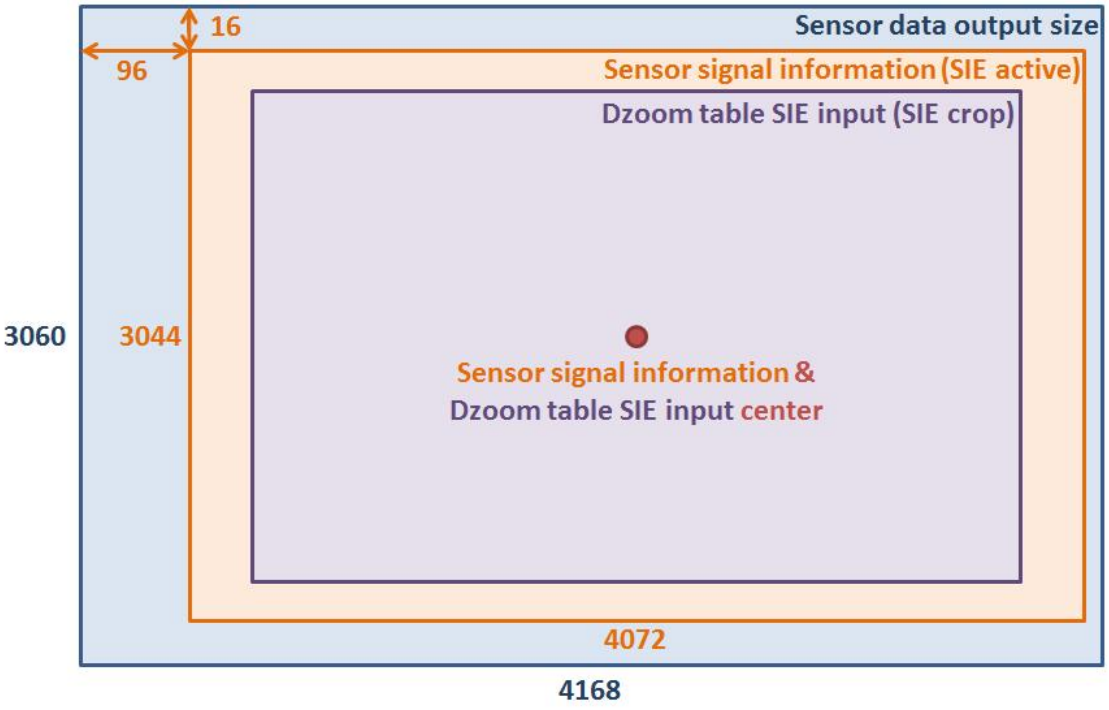
```
typedef struct
{
    UINT32 Sync;
    UINT32 Period;
    UINT32 DataStart;
    UINT32 DataSize;
} SENSOR_SIGNAL;
```

```
static SENSOR_SIGNAL HD_SEN_IMX078CQK[MODE_MAX + 1] =
{
    {0,    0, 0, 0},
    {0, 1170, 0, 0}, //mode 0
    {0,  462, 0, 0}, //mode 2
    {0,  572, 0, 0}, //mode 4
    {0,  462, 0, 0}, //mode 2 p30
    {0,  550, 0, 0}, //mode 1 4000*3000
    {0,  550, 0, 0}, //mode 1 4000*3000
    {0,  550, 0, 0}, //mode 1 4000*3000
    {0,  550, 0, 0}, //New mode 1 4000*2000 60fps
    {0,  550, 0, 0}, //New mode 1 4000*2000 60fps -> 30fps
    {0,  550, 0, 0}, // SENSOR_MODE_6 scale down for video mode
};

static SENSOR_SIGNAL VD_SEN_IMX078CQK[MODE_MAX + 1] =
{
    {0,    0, 0, 0},
    {0, 3080, 0, 0}, //mode 0
    {0, 2600, 0, 0}, //mode 2
    {0, 1050, 0, 0}, //mode 4
    {0, 2600, 0, 0}, //mode 2 p30
    {0, 4368, 0, 0}, //mode 1 4000*3000
    {0, 4368, 0, 0}, //mode 1 4000*3000
    {0, 5460, 0, 0}, //mode 1 4000*3000
    {0, 2184, 0, 0}, //New mode 1 4000*2000 60fps
    {0, 2184, 0, 0}, //New mode 1 4000*2000 60fps -> 30fps
    {0, 4368, 0, 0}, // SENSOR_MODE_6 scale down for video mode
};
```

(5) Sensor signal information (for SIE active window)

```
static SENSOR_SIGNAL HD_TRANS_IMX078CQK[MODE_MAX + 1] =
{
    {0, 0, 0, 0},
    {0, 0, 96, 4168 - 96}, //mode 0
    {0, 0, 48, 2084 - 48}, //mode 2
    {0, 0, 32, 1388 - 32}, //mode 4
    {0, 0, 48, 2084 - 48}, //mode 2 p30
    {0, 0, 96, 4168 - 96}, //mode 1 4000*3000
    {0, 0, 96, 4168 - 96}, //mode 1 4000*3000
    {0, 0, 96, 4168 - 96}, //mode 1 4000*3000
    {0, 0, 96, 4168 - 96}, //New mode 1 4000*2000 60fps
    {0, 0, 96, 4168 - 96}, //New mode 1 4000*2000 60fps -> 30fps
    {0, 0, 96, 4168 - 96}, // SENSOR_MODE_6 scale down for video mode
};

static SENSOR_SIGNAL VD_TRANS_IMX078CQK[MODE_MAX + 1] =
{
    {0, 0, 0, 0},
    {0, 0, 16,3060 - 16}, //mode 0
    {0, 0, 4,1150 - 6}, //mode 2
    {0, 0, 4,1020 - 4}, //mode 4
    {0, 0, 4,1150 - 6}, //mode 2 p30
    {0, 0, 16,3060 - 16},//mode 1 4000*3000
    {0, 0, 16,3060 - 16},//mode 1 4000*3000
    {0, 0, 16,3060 - 16},//mode 1 4000*3000
    {0, 0, 16,2060 - 16},//New mode 1 4000*2000 60fps
    {0, 0, 16,2060 - 16},//New mode 1 4000*2000 60fps -> 30fps
    {0, 0, 16,3060 - 16},// SENSOR_MODE_6 scale down for video mode
};
```

用来决定 SIE active 的 data start 及 data size，框到的部分必须要有 LVDS 讯号，SIE crop window (dzoom table)中心点会根据 active window 中心点来决定，必须确保最后 crop window 没有坏图。以 mode 0 为例，示意图如下：



(6) Sensor mode information

```
typedef struct
{
    SENSOR_MODE_TYPE ModeType;      ///< sensor mode type(HDR or .....)
    UINT32 SIEFreq;                 ///< SIE clock frequence Hz
    UINT32 MCLKFreq;                ///< MCLK frequence Hz
    SENSOR_STPIX StPix;             ///< Sensor start pixel
    SENSOR_FMT Fmt;                 ///< Sensor data type
    SENSOR_IMG_RATIO Ratio;         ///< Sensor ratio information
    SENSOR_OB OB;                   ///< Sensor OB
    UINT32 FrameRate;               ///< frame rate X 10
    UINT32 Pclk;                    ///< pixel clock Hz
    UINT32 biningRatio;             ///< binning ratio X 100
    UINT32 StrLnT;                  ///< length from VD start to 1st active line(including OB),
    UINT32 EndLnT;                  ///< length from VD start to last active line(including OB),
    UINT32 TransDelyT;              ///< length from exposure end to start of data transmission,

    SENSOR_SEL_IMG_ID *SelImgId;    ///< sensor select frame information
    SENSOR_SIGNAL *TransHD;         ///< transfer HD signal
    SENSOR_SIGNAL *TransVD;         ///< transfer VD signal
    SENSOR_SIGNAL *Trans2HD;        ///< transfer HD signal (for HDR Sensor frame 2)
    SENSOR_SIGNAL *Trans2VD;        ///< transfer VD signal (for HDR Sensor frame 2)
    SENSOR_SIGNAL *SenHD;           ///< Sensor HD signal
    SENSOR_SIGNAL *SenVD;           ///< Sensor VD signal

    SENSOR_LVDS *LVDS;              ///< lvds information

    SENSOR_DVI *DVI;                ///< dvi information
} SENSOR_MODE_INFO;
```

```
static SENSOR_MODE_INFO Mode_IMX078CQK[MODE_MAX + 1] =
{
    { //mode 0
        SENSOR_MODE_LINEAR,
        288000000,
        72000000,
        SENSOR_STPIX_R,
        SENSOR_FMT_POGRESSIVE,
        {SENSOR_RATIO_4_3, 1000, 1000},
        {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}},
        200,
        72000000,
        100,
        621,
        49641,
        0,
        NULL,
        &HD_TRANS_IMX078CQK[1],
        &VD_TRANS_IMX078CQK[1],
        NULL,
        NULL,
        &HD_SEN_IMX078CQK[1],
        &VD_SEN_IMX078CQK[1],
        &LVDS_IMX078CQK[1],
        NULL
    },
```

- ModeType：sensor mode type

  SENSOR_MODE_LINEAR：一般模式

  SENSOR_MODE_BUILTIN_HDR：sensor 出来的影像是经过 HDR 效果，只有一张影像 (EX：AR0230)。

  SENSOR_MODE_STAGGER_HDR：sensor 会出两张不同曝光时间的影像，提供给 IPL 做合成 (EX：OV4689)。

  SENSOR_MODE_CCIR：CCIR sensor

  SENSOR_MODE_CCIR_INTERLACE：CCIR interlace sensor

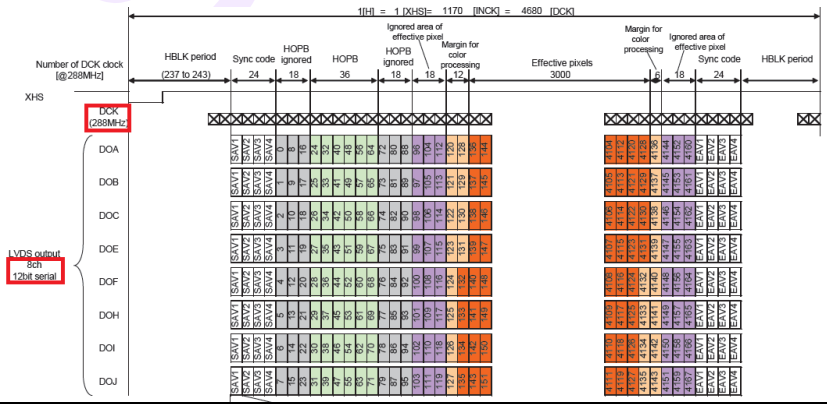
- SIEFreq：SIE clock 频率

  SIE 收的速度必须大于等于 LVDS 送的速度，LVDS 送的速度计算方式如下：

  LvdsClk= (data clock) x (number of data lane) / (mumber of bits per pixel)

  = (288M)*8/12 = 192M

  如果 Data_Lane_Number 为奇数，Data_Lane_Number = Data_Lane_Number + 1



SIE frequency 最大限制为 297M，每个 sensor mode 可以不一样，但是必须可以从 SIE Source 除得出来，以 SIE Source=480M 为例，SIE frequency 必须为 240M /160M /120M /96M /.....， SIE Source 在 IMX078CQK.c 当中设定，位置如下：

```
static ER Init_IMX078CQK(SENSOR_ID Id, SENSOR_INIT_OBJ *InitObj)
{
.....
    SensorIF_GetSIECtrlObj(Id)->SIE_CLK_chgClockSource(SIE_CLKSRC_PLL5);
.....
}
```

PLL2：SIE Source= 设定频率*(12M)/131078

PLL5：SIE Source= 设定频率*(12M)/131078

EX：0x300000*(12M)/131072 = 288M

PLL_CLKSRC_480：SIE Source= 480M

PLL_CLKSRC_240：SIE Source= 240M

设定频率在 DxCamera_Sensor.c 当中给定，位置如下：

```
static void SenPowerOn(void)
{
.....
    pll_setPLL(PLL_ID_5, 0x300000); //PLL5
.....
}
```

- MCLKFreq：sensor input clk

Sensor spec 会写 sensor 需要多少频率的 input clock，Input clock 除 2 等同于 frame rate 降一半。

◆ Input clock frequency 72 MHz

但是必须可以从 MCLK Source 除得出来，以 MCLK Source =480M 为例，MCLK frequency 必须为 240M /160M /120M /96M /.....，MCLK source 在 DxCamera_Sensor.c 当中设定，位置如下：

```
static void SenPowerOn(void)
{
.....
    pll_selectClkSrc(PLL_CLK_SIEMCLK, PLL_CLKSRC_PLL5);
.....
}
```

PLL2：MCLK Source = 设定频率*(12M)/131078

PLL5：MCLK Source = 设定频率*(12M)/131078

PLL_CLKSRC_480：MCLK Source = 480M

PLL_CLKSRC_240：MCLK Source = 240M

- StPix：

  Sensor 出影像的时候，start pixel 是 R/Gr/Gb/B 哪一种，NT96660 统一为 SENSOR_STPIX_R，若有 start pixel 不对，导致影像颜色有问题，请调整 start position。

- Fmt：sensor data format。目前只有 SENSOR_FMT_POGRESSIVE。

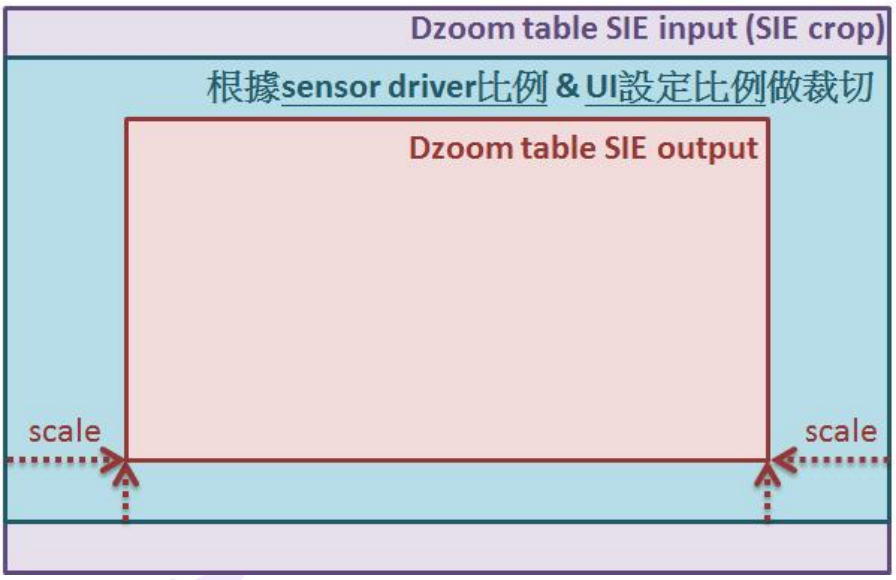
- Ratio：

  RatioHV(SENSOR_RATIO_4_3/ 3_2/ 16_9/ 1_1)：通常可以根据 sensor active 大小决定比例；在 IPL 当中，会根据下列流程决定是否要裁切以符合比例。

  i. 判断 sensor driver 设定比例与 UI 设定比例是否相同，若不相同，根据 dzoom table 第一段(sie crop/sie input)所得到的大小做裁切

  ii. 会同时裁切 dzoom table 第二段(sie output)及第三段(IPL input)的大小。

  iii. 再将裁切后的影像，根据 sie output 大小做 scale down。



RatioH/RatioV：根据 sub-sampling 的方式或 binning 的方式来计算，ECS/DIS 等会采用此参数，根据不同 sensor mode 大小做参数的换算，计算方式如下:

mode	RatioH	RatioV
0	1000 / 1 = 1000	1000 / 1 = 1000
1	1000 / 1 = 1000	1000 / 1 = 1000
2, 2A	1000 / 2 = 500	1000 / 2 = 500
3, 8	1000 / 3 = 333	1000 / 3 = 333

4, 5	1000 / 3 = 333	1000 / 9 = 111
6	1000 / 3 = 333	1000 / 19 = 53
7	1000 / 2 = 500	1000 / 2 = 500

Relationship between Arithmetic Processing and the Number of Output Bits in Each Readout Drive Mode

Readout mode No.	A/D conversion resolution	Horizontal pixel processing	Vertical pixel processing	Total number of binning pixels	Internal arithmetic processing	Number of output bits
0	12 bits	—	—	—	—	10 bits + 2 bits *1
1	10 bits	—	—	—	—	10 bits
2, 2A	10 bits	2 binning	2 binning	4 pixels	9/16, 3/16, 1/16 (weighted binning *2)	10 bits
3, 8	10 bits	3 binning	2/3 subsampling binning	6 pixels	1/6	10 bits
4, 5	10 bits	3 binning	2/9 subsampling binning	6 pixels	1/6	10 bits
6	10 bits	3 binning	2/19 subsampling binning	6 pixels	1/6	10 bits
7	10 bits	2/4 subsampling	2 binning	2 pixels	3/4, 1/4 (weighted binning *2)	10 bits

- OB：要拿来计算 OB 值的区域，只有在自动计算 OB 方式下才会采用，目前都是减固定的 OB 值，所以都设为 0。
- FrameRate (*10)：影像 frame rate (取整数)

Readout mode No.	NTSC compatible drive			
	XHS period (INCK) *1	H period (number of XHS pulses) *2	V period (number of XHS pulses)	Frame frequency [frame/s]
0	1170	1	3080	19.98
1	550	1	3120	41.96
2	462	2	2600	59.94
3	390	3	3080	59.94
4	572	1	1050	119.88
5	420	2	715	239.76
6	420	8	5720	29.97
7	385	2	390	479.52
7A	390	2	231	799.20
7B	396	2	182	999.00
8	572	1	630	199.80
9	462	2	2600	59.94

- Pclk：sensor output clk，用在同步用 VD/HD，目前为 AE 使用。
- biningRatio (*100)：

Relationship between Arithmetic Processing and Number of Output Bits in Each Readout Drive Mode

Readout mode No.	A/D conversion resolution	Horizontal pixel processing	Vertical pixel processing	Total number of added pixels	Internal arithmetic processing	Number of output bits
0	12 bits	—	—	—	—	10 bits + 2 bits *1
1	10 bits	—	—	—	—	10 bits
2	10 bits	2 addition	2 addition	4 pixels	1/4	10 bits + 2 bits *2
3	9 bits	3 addition	3 addition	9 pixels	2/9	10 bits + 2 bits *2
4	10 bits	3 addition	1/3 elimination	3 pixels	1/3	10 bits + 2 bits *2
5	9 bits	3 addition	2/9 elimination addition	6 pixels	2/6	10 bits + 2 bits *2
6	9 bits	3 addition	2/9 elimination addition	6 pixels	2/6	10 bits + 2 bits *2
7	9 bits	3 addition	2/17 elimination addition	6 pixels	2/6	10 bits + 2 bits *2
7A	9 bits	3 addition	2/17 elimination addition	6 pixels	2/6	10 bits + 2 bits *2
7B	9 bits	3 addition	2/17 elimination addition	6 pixels	2/6	10 bits + 2 bits *2
8	10 bits	3 addition	1/5 elimination	3 pixels	1/3	10 bits + 2 bits *2
9	10 bits	2/4 elimination	2 addition	2 pixels	1/2	10 bits + 2 bits *2

Binning ratio = (Total number of added pixels) * (internal arithmetic processing)

mode	binning
0,1,2,4,8,9	1
3,5,6,7,7A,7B	2

- StrLnT/ EndLnT/ TransDelyT：RSC 相关参数，请参考 RSC 文件。
- SelImgId：sensor select frame information，CCD sensor field 相关参数，目前没有在用。
- TransHD/ TransVD：Sensor signal information (for SIE active window) 。
- Trans2HD/ Trans2VD：SENSOR_MODE_STAGGER_HDR 给第二张 frame 使用。
- SenHD/ SenVD：Sensor signal information (for AE) 。
- LVDS：Sensor LVDS information。
- DVI：Sensor DVI structure，CCIR sensor 相关参数，(EX：NT99141)

(7) Sensor information

IPL 当中，会统一用这个 struct 取得 sensor 相关信息。

```
typedef struct
{
    SENSOR_TYPE SenType;          ///< sensor type
    SENSOR_SIGNAL_TYPE SigType;   ///< sensor mask or slave
    SENSOR_DATA_TYPE DataType;    ///< transfer type
    UINT32 CellWidth;             ///< cell width mm * 1000
    UINT32 CellHeight;            ///< cell height mm * 1000
    UINT32 Width1X;               ///< width (full scan pixel size)
    UINT32 Height1X;              ///< height(full scan pixel size)
    UINT32 SyncTiming;            ///< sync timing for Exposure time & gain(VD)
    SENSOR_CMD_TYPE CmdType;      ///< protocol type
    SENSOR_FPS_TYPE FPSType;      ///< control frame rate type
    SENSOR_MODE_INFO *Mode;       ///< sensor current mode
} SENSOR_INFO;
```

```
static SENSOR_INFO g_pIMX078CQK_BASE_INFO =
{
        SENSOR_TYPE_CMOS,
        SENSOR_SIGNAL_SLAVE,
        SENSOR_DATA_LVDS,
        6460,
        4743,
        4000,
        3000,
           1,
        SENSOR_CMD_UNKNOWN,
        SENSOR_FPS_DEPEND_ON_EXPT,
        NULL,
};
```

- SenType：目前都是 CMOS sensor。
- SigType：决定 sensor 是 slave 还是 master。
  SENSOR_SIGNAL_MASTER：sensor 提供 HD/VD 给 DSP。
  SENSOR_SIGNAL_SLAVE：DSP 提供 HD/VD 给 sensor。
  参考来源：sensor spec，以下图为例，对 sensor 来说 XHS/XVS 是 input，所以这个 sensor 是 slave。

H10	XHS	I	D	Horizontal sync signal input

J10	XVS	I	D	Vertical sync signal input

- DataType：sensor 传递讯号的方式，目前有 PARALLEL/LVDS/MIPI。
- CellWidth/ CellHeight：RSC 相关参数，请参考 RSC 文件。
- Width1X/ Height1X：sensor 提供的有效 pixel 数量。

◆Number of recommended recording pixels
- Type 1/2.3 approx. 16.35 M pixels use : 4608 (H) × 3456 (V) approx. 15.93 M pixels aspect ratio 4:3

- SyncTiming：AE 相关参数，请参考 AE 文件。
- CmdType：sensor 传递 command 的方式，目前有 Vx1/SIF/I2C/IO。
- FPSType：RSC 相关参数，请参考 RSC 文件。
- Mode：Sensor mode information。

(8) Sensor register

数据参考来源：直接问 sensor 厂是否能提供 / 从 sensor spec 上查询

- 基本 register：

3. Register Setting for Each Readout Drive Mode

The register setting for each readout drive mode available with this sensor is shown in the table below.

Address	Bit assignment	Register name	Readout mode No.*2											
			0	1	2	3	4	5	6	7	7A	7B	8	9
0001h	[3:0]	STBLVDS	1h	0h	3h	3h	3h	3h	5h	3h	3h	3h	3h	3h
	[6:4]	CHSEL	1h	0h	3h	3h	3h	3h	5h	3h	3h	3h	3h	3h
	[7]		0h											
0002h	[7:0]	SVR	According to exposure time						Multiple of 8 – 1	According to exposure time				
0003h	[7:0]													
000Dh	[3:0]	MDSEL	0h	1h	2h	3h	4h	5h	5h	6h	Bh	Dh	7h	2h
	[6:4]		0h											
	[7]	MDVREV	0h: vertical direction normal/1h:inverted											
000Eh	[0]	SMD	0h*1	0h*1	0h	0h	0h	0h	0h	0h	0h	0h	0h	0h
	[7:1]		0h											
0017h	[3:0]	RDHMD	0h	0h	0h	0h	0h	0h	3h	0h	0h	0h	0h	0h
	[7:4]	GSVR	*3		(invalid)									
0019h	[1:0]	ADBIT	3h	1h	1h	0h	1h	0h	0h	0h	0h	0h	1h	1h
	[6:2]		0h											
	[7]	PSMONEN	0h	0h	0h	0h	0h	0h	1h	0h	0h	0h	0h	0h
0069h	[0]	MDDCMT	0h	0h	0h	0h	0h	0h	0h	0h	0h	0h	0h	1h
	[7:1]		0h											

- 有些 register 可以控制 AE 相关参数：

0002h	[7:0]	0000h	Next frame after communication end *2	SVR	Specifies the exposure shutdown vertical period
0003h	[7:0]				
0004h	[7:0]	0000h	*2	SPL	Specifies the exposure start vertical period
0005h	[7:0]				
0006h	[7:0]	0000h	*2	SHR	Specifies the exposure start horizontal period
0007h	[7:0]				
0008h	[7:0]	000h	*1	PGC	Analog gain setting
0009h	[2:0]				

0010h	[1:0]	0h	*1	DGAIN	Digital gain setting 0h: 0 dB gain setting value 1h: +6 dB gain setting value 2h: +12 dB gain setting value 3h: +18 dB gain setting value
	[3:2]	0h			—
	[7:4]	8h			—

- 有些 register 可以看这个 sensor 本身是否能做 flip/mirror 的功能：

	[7]	0h	*1	MDVREV	0h: Vertical direction normal readout 1h: Vertical direction inversion readout

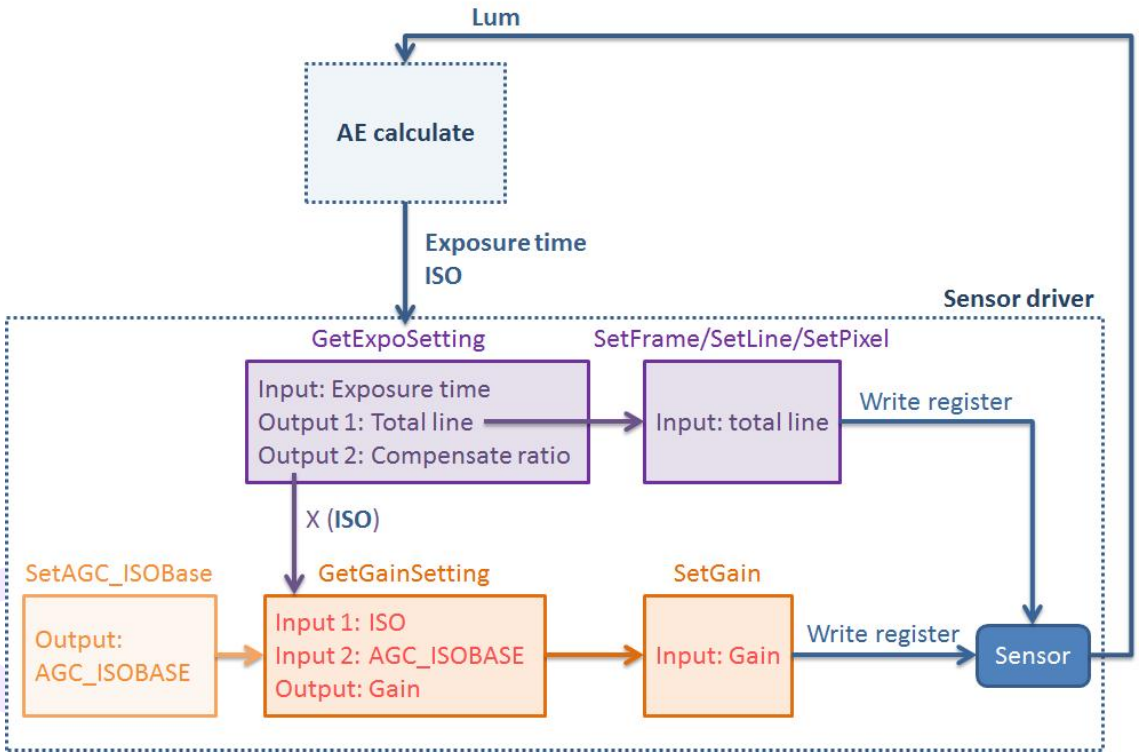
3. IMX078CQK.c

(1) 主要为 sensor API (sensor 统一控制接口)，API 功能请参考文件 *NT96660 Sensor*

Driver Application Note 所描述。

(2) 需要注意 AE 相关部分，要根据不同 sensor 做修改，function 如下：

- GetExpoSetting_IMX078CQK()：
  ➔AE 所算出来的曝光时间
  ➔ 透过(Sensor signal information for AE) 的 HD period 信息，换算出总共需要曝几条 line (有些 sensor 可以 surport pixel)
  ➔ 换算 line 的过程，会有无法整除的问题，必须再换回曝光时间计算其误差
  ➔ 误差换算为 compensate ratio，会再利用 gain 做补偿
- SetFrame_IMX078CQK()/SetLine_IMX078CQK()/SetPixel_IMX078CQK()：
  ➔ 将 GetExpoSetting()计算出来的曝光条数，换算后设定给 sensor register
- SetAGC_ISOBase_IMX078CQK()：
  ➔ 当机器有做 ISO 校正，透过 function 给定校正值，并且在设 gain 之前针对 ISO 校正，否则都是 initial 的固定值。
- GetGainSetting_IMX078CQK()：
  ➔ 将 GetExpoSetting 得到的 compensate ratio，乘上 AE 算出来的 ISO 值
  ➔ 利用 SetAGC_ISOBase_IMX078CQK 参数将 ISO 值做校正
  ➔ 根据 sensor 特性，将 ISO 换算为 analog gain/digital gain
- SetGain_IMX078CQK()：
  ➔ 将 GetGainSetting_IMX078CQK 所得到的 gain 值设给 sensor register
- 运作原理如下：



- 參考來源：Sensor spec

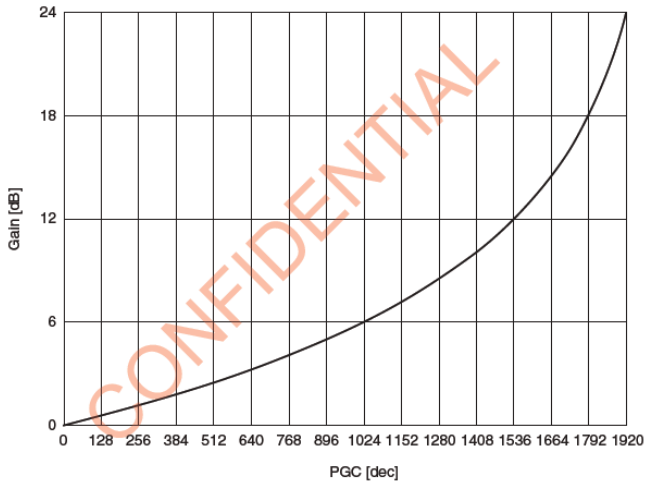
➔ GetGainSetting_IMX078CQK

PGC Setting

Register value	Function
0h to 780h (0d to 1920d)	Analog gain setting

Relational Formula

Gain [dB] = –20log{(2048 – PGC [10:0])/2048}



Relationship between Register Setting Value and Set Gain Value

➔ SetLine_IMX078CQK()

Register	Register value		Function
SHR	7 to {(SVR + 1) number of XHS pulses per frame – 4}	Readout mode No.0, 1 and 5 All-pixel scan mode (12 bits/10 bits) Vertical 2/9 elimination addition mode (4ch/1ch output)	Specifies the exposure start horizontal period
	8 to {(SVR + 1) number of XHS pulses per frame – 4}	Readout mode No.2 and 9 Horizontal/vertical 2/2-line addition readout mode Horizontal 2/4 elimination readout mode (16:9 cropping)	
	12 to {(SVR + 1) number of XHS pulses per frame – 7}	Readout mode No.7,7A and 7B Vertical 2/17 elimination mode	
	0 to {(SVR + 1) number of XHS pulses per frame – 2}	Global reset shutter (SMD = 1)	
	7 to {(SVR + 1) number of XHS pulses per frame/4 – 4}	Readout mode No.6 Vertical 2/9 elimination addition mode (low power consumption drive)	
	8 to {(SVR + 1) number of XHS pulses per frame – 5}	Readout mode No.3 Horizontal/vertical 3/3-line addition readout mode	
	5 to {(SVR + 1) number of XHS pulses per frame – 4}	Readout mode No.4 Vertical 1/3 elimination mode	
	8 to {(SVR + 1) number of XHS pulses per frame – 4}	Readout mode No.8 Vertical 1/5 elimination mode	

➔ GetExpoSetting_IMX078CQK

Readout mode No.	0	1	2	3	4	5	6	7	7A	7B	8	9
Number of clocks per internal offset period	312	141	141	126	141	126	126	126	126	126	141	141

(3) Chip ID

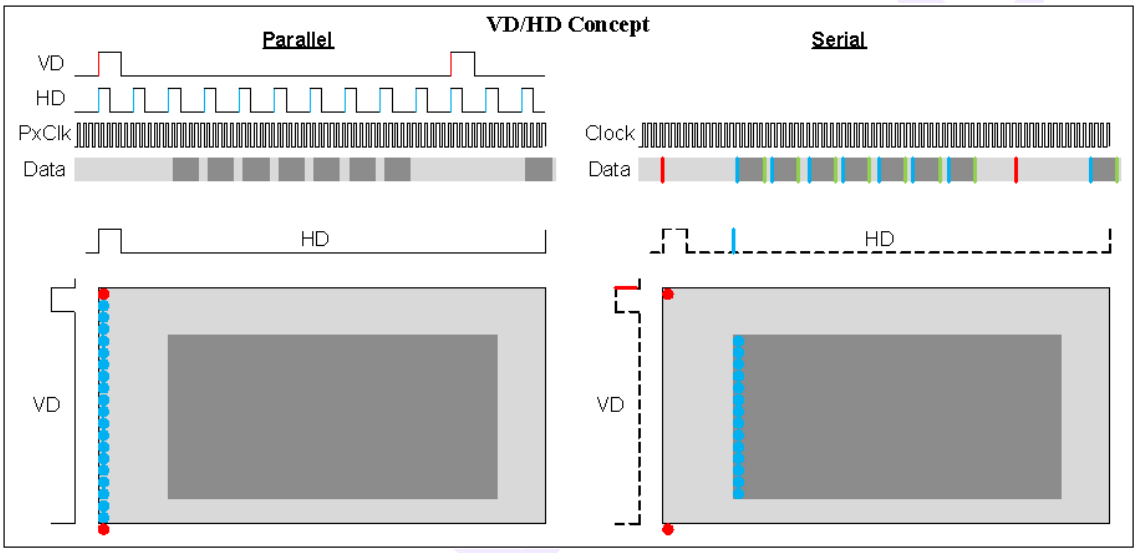
```
#define CHIPID     0x81
```

Transmit the Chip ID (fixed value: 81h) in the first byte.

3 Appendix

1. LVDS 使用方式：

2. sensor 传递讯号方式：



Parallel MIPI/LVDS

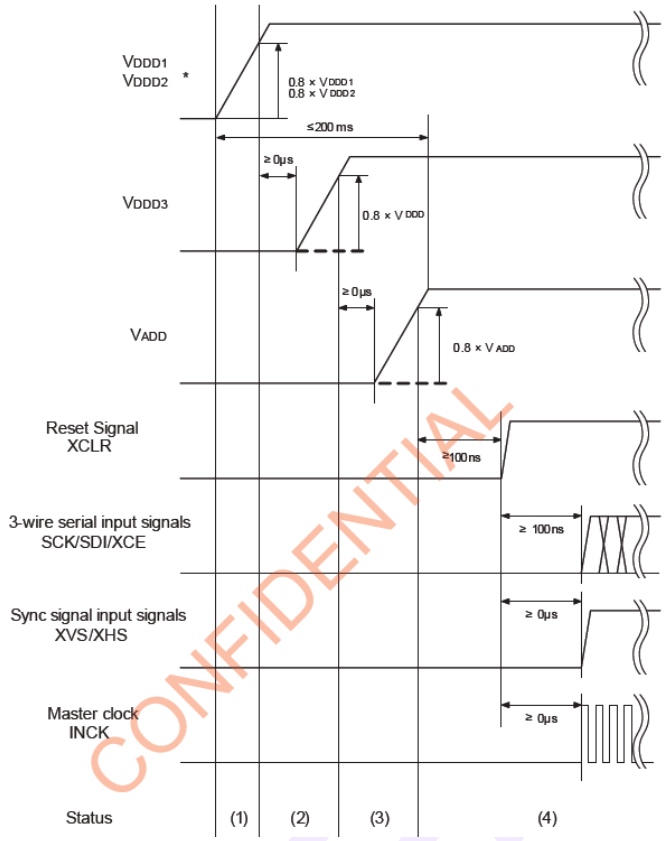
3. Sensor power on sequency

   档案及 function：DxCamera_Sensor.c / SenPowerOn()

   Sensor spec 会写 power on 的顺序，要请 HW 提供相关 pin 脚。

1. Power-on Sequence

All power supplies should finish rising within 200 ms.



4 Revision History

Revision	Date	Author	Changes
0.1	2016/03/01	Silvia Wu	First draft version. (only LVDS sensor)