Crystal Image through
Imaging Innovation

PIXELPLUS

GUI

PI5008K

User Guide

Rev 0.3

Last Update: 2018.11.16

6th Floor, 105, Gwanggyo-ro, Yeongtong-gu,
Suwong-si, Gyeonggi-do, 16229, Korea
Tel : +82-31-888-5300, FAX : +82-31-888-5399

Contents

1. Overview

Before describing how to customize GUI, an overview for DU(Display Unit) block will be introduced to help you understand the hardware limitation of PI5008K. In this guide, the limitation is explained briefly. Please refer to DU part of PI5008 data sheet

1.1. Structure

DU(Display Unit) block provides overlay channels and controls final video output.

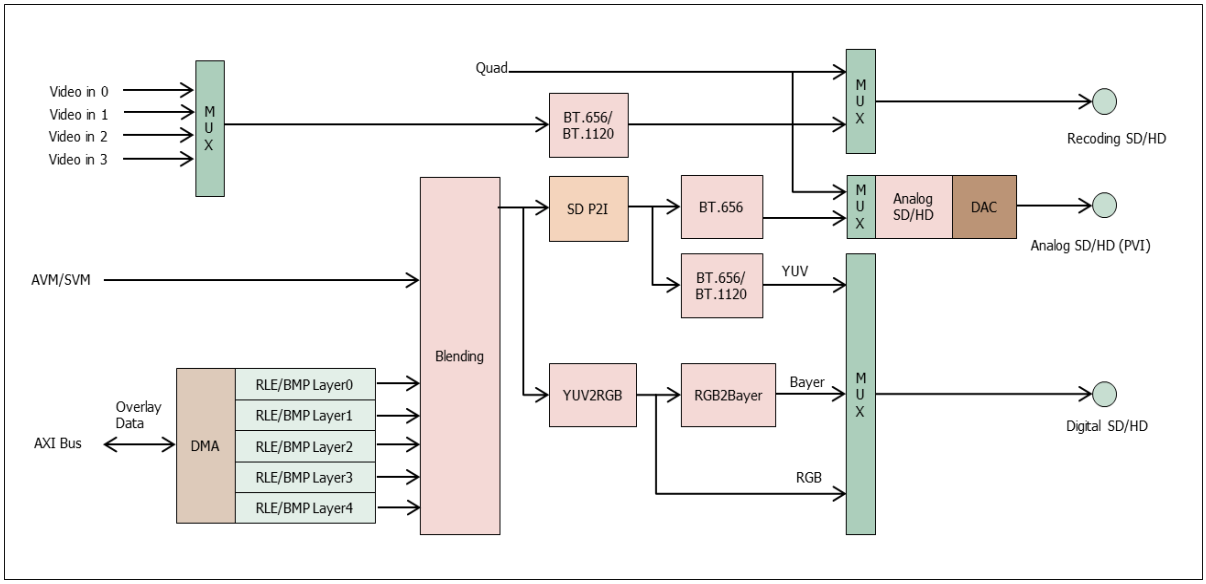


Fig1. DU Block Diagram

1.2. Specification

1.2.1. OSD (On Screen Display)

- 5 Overlay Layers. 4 areas per layer
  Layers can be overlapped but areas cannot be overlapped.
- 1/8/16/24/32 bit per Overlay Data Format & RLE

Overlay Data format	Description
RLE	Run-Length Encoding : [15:8]:number of pixel which has same index, [7:0]index of Color LUT Color LUT: [31:24]: Alpha, [23:16]: R, [15:8]: G, [7:0]: B

	- Color LUT with 256 entries
1-bit	1 if there is a data in 1 bit, else 0 (designed for Edge or VPU) Color LUT: [31:24]: Alpha, [23:16]: R, [15:8]: G, [7:0]: B - Index 0 : Background, Index1 : Foreground Color
8-bit(INDEX Mode)	1 Mode: [0] Color LUT Index Color LUT: [31:24]: Alpha, [23:16]: R, [15:8]: G, [7:0]: B - 256 개 Color LUT
16-bit	4:4:4:4 Mode: [15:12]: R, [11:8]: G, [7:4]: B, [3:0]: Alpha 5:6:5 Mode: [15:11]: R, [10:5]: G, [4:0]: B
24-bit	8:8:8 Mode : [23:16]:R, [15:8]:G, [7:0]:R
32-bit	8:8:8:8 Mode: [31:24]: Alpha, [23:16]: R, [15:8]: G, [7:0]: B Maximum horizontal size of the area which uses this mode will be limited to 1/2 of display(layer) size. For example, max horizontal size of the area will be limited to 1280/2=640 if display size is HD.

- RLE or BMP mode

	RLE	INDEX (8bit)	RGB565 (16bit)	RGBA4444 (16bit)	RGB888 (24bit)	ARGB8888 (32bit)	1BIT
Layer 0	O	O	O	O	X	X	O
Layer 1	O	O	O	O	X	X	O
Layer 2	O	O	O	O	X	X	O
Layer 3	O	O	O	O	O	O	O
Layer 4	O	O	O	O	O	O	O

1.2.2. Video Output

- Analog: Multi-Standard Analog HD, SD(NTSC/PAL)
  - DAC: 10-bit@148.5MHz
- Digital: YUV/RGB/Bayer
  - BT.656 8bit, BT.1120 8/16bit
  - RGB 888
  - Bayer 8/10bit

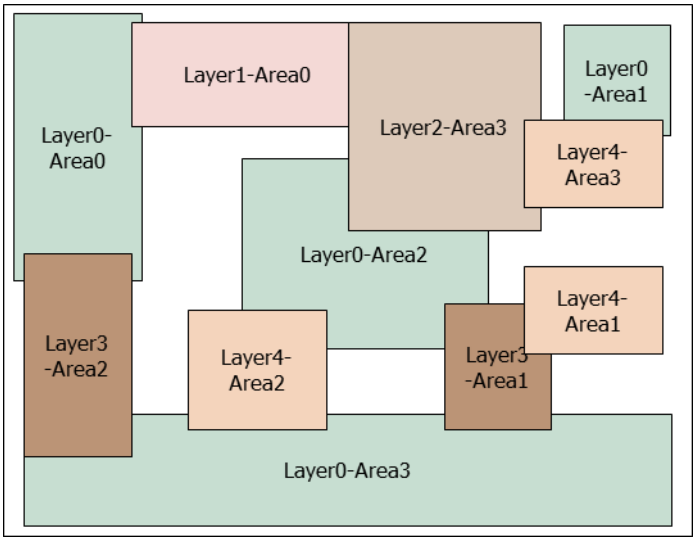


Fig2. Layer/Area Allocation Example

1.2.3. Layer Order

By default, layer order is configured as below. The layer order can be adjusted except video input.

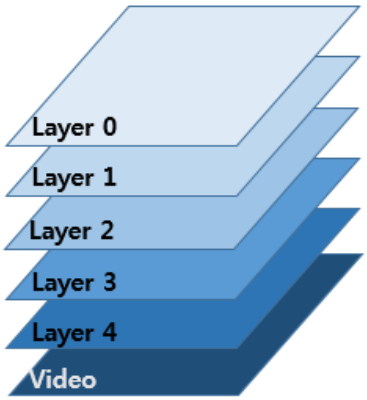


Fig3. Default Layer Order

2. How to create resource binary

This section shows how to convert RGB888 BMP image to RLE format, and then how to convert RLE image into the binary image which can be used by PI5008K. Same procedure will be applied for UI, Car, and Parking Guide Line.

2.1. Procedure

image1.bmp image2.bmp	→	LUT.bin image1_RLE.bin image2_RLE.bin	→	resource.bin
BMP Image Generation		RLE & LUT binary Generation		Generating UI/Car/PGL resource binary which includes RLE&LUT binary and header

Please refer to 5.2 ui_rscimg_720p.bin for the structure of resource binary file which includes header which is the result of final procedure.

2.1.1. UI

(1) Make BMP image.

source\applications\Display\1280x720\UI\Booting\Layer1\Booting_CI.bmp

… …

(2) Make image list to be converted into RLE.

source\applications\Display\1280x720\UI\Booting\Layer1_LIST_Booting_Layer1.txt

(3) Run batch file to convert into LUT & RLE.

source\applications\Display\1280x720\UI\Booting\Layer1_RLE_Booting_Layer1.bat

(4) Check whether below files are generated.

source\applications\Display\1280x720\UI\Booting\Layer1\ _LUT_Booting_Layer1.bin

source\applications\Display\1280x720\UI\Booting\Layer1\ _LUT_Booting_Layer1.txt

source\applications\Display\1280x720\UI\Booting\Layer1\ Booting_CI_RLE.bin

… …

(5) Make list to make UI binary. Please refer to 5.1 _rscimg_merger_ui_list.txt for list

structure.

source\applications\Display\1280x720_rscimg_merger_ui_list.txt

(6) Make UI binary which include LUT, RLE and headers using batch file.

source\applications\Display\1280x720_rscimg_merger_ui.bat

(7) Check whether UI binary is generated correctly. Please refer to c structure of 5. Appendix for the structure of binary file.

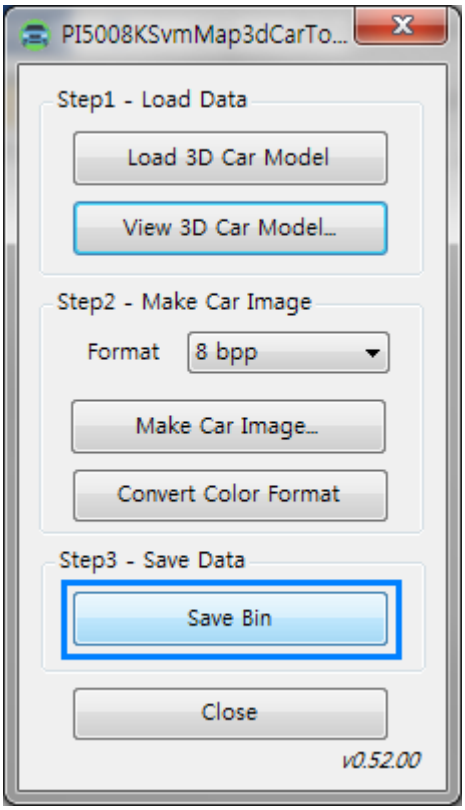
source\applications\Display\1280x720\ui_rscimg_720p.bin

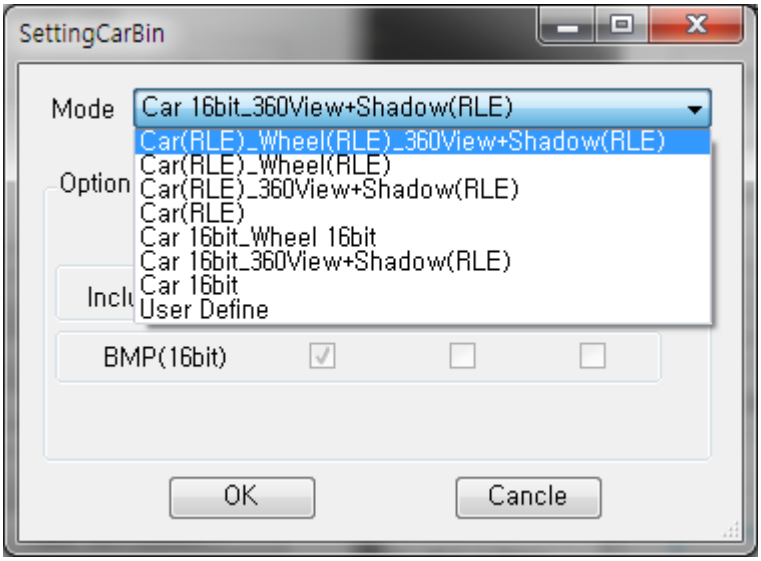
\source\applications\Display\ui_rsclist.h

source\Image\ui_rscimg_720p.bin

2.1.2. CAR Image

3D Car Tool (PI5008KSvmMap3dCarTool.exe) is used to produce binary file which includes car images. If Save Bin button is pressed, dialog box will be displayed to choose images. After choosing image, press OK to make image file. To display car image properly, options for SDK has to be sync with 3D car tool. After setting option for SDK, rebuild it and download car image file and main binary. Please refer to 3DCarTool guide for usage of the tool.

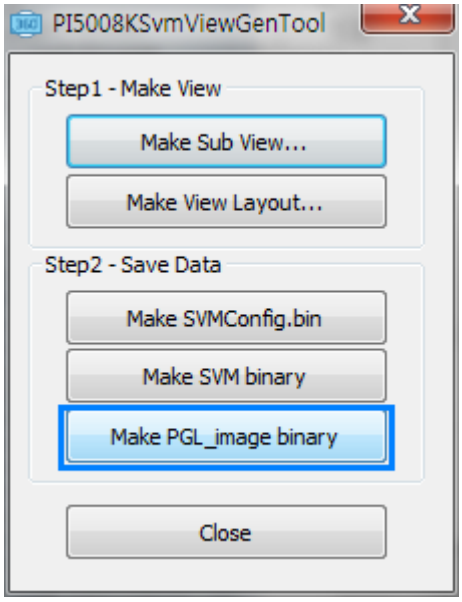




Description	"prj_config_scenario.h" Option Config
Car(RLE)+Wheel(RLE)+360View&Shadow(RLE)	//#define USE_16BIT_CAR #define USE_CAR_WHEEL #define CACHE_VIEW_USE
Car(RLE)+Wheel(RLE)	//#define USE_16BIT_CAR #define USE_CAR_WHEEL //#define CACHE_VIEW_USE
Car(RLE)+360View&Shadow(RLE)	//#define USE_16BIT_CAR //#define USE_CAR_WHEEL #define CACHE_VIEW_USE
Car(RLE)	//#define USE_16BIT_CAR //#define USE_CAR_WHEEL //#define CACHE_VIEW_USE
Car(16bit)+Wheel(16bit)	#define USE_16BIT_CAR #define USE_CAR_WHEEL //#define CACHE_VIEW_USE
Car(16bit)+360View&Shadow(RLE)	#define USE_16BIT_CAR //#define USE_CAR_WHEEL #define CACHE_VIEW_USE
Car(16bit)	#define USE_16BIT_CAR //#define USE_CAR_WHEEL //#define CACHE_VIEW_USE

2.1.3. PGL

View Generation (PI5008KSvmViewGenTool.exe) is used to produce binary file which includes parking guide line images. Press Make PGL_image binary button to produce image file. You can find new PGL line after downloading it to PI5008K. Please refer to View Generation Tool guide for usage of the tool.



3. How to modify High Layer

This section is for a customer who wants change only UI skin. A specific image can be modified or added to a scene. It is possible to make the limited modification for reference GUI of Pixelplus.

3.1. Modify image

3.1.1. Sequence

(1) Before modifying an image, you need to understand how to design graphics image and which limitation exists in layer structure by referring PI5008K_GUI_Design_Guideline_vx.x.pptx.

(2) Modify the target image.

(3) Convert image into RLE format.

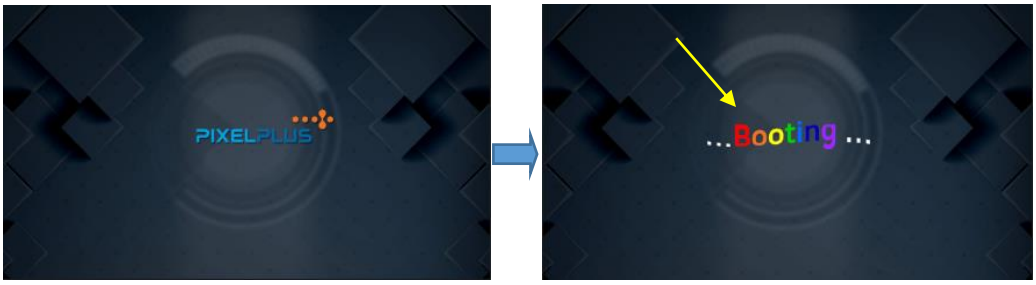
(4) Make UI resource image which includes the modified image.

(5) Make the binary image to download to flash memory.

(6) After downloading the binary image, check whether the image is modified correctly.

3.1.2. Example

(1) This example shows how to modify the booting image.

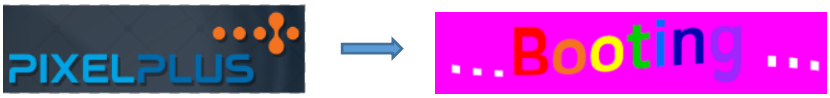


(2) Image files to display the original booting image are as follows.

- source\applications\Display\1280x720\UI\Common\Layer4\Background.bmp

- source\applications\Display\1280x720\UI\Booting\Layer1\Booting_CI.bmp

(3) Modify Booting_CI.bmp as follows.



(4) After running 'source\applications\Display\1280x720\UI\Booting\Layer1_RLE_Booting_Layer1.bat', check whether two files are updated.

- source\applications\Dislay\1280x720\UI\Booting\Layer1_LUT_Booting_Layer1.bin
- source\applications\Display\1280x720\UI\Booting\Layer1\Booting_CI_RLE.bin

(5) Modify 'source\applications\Display\1280x720_rscimg_merger_ui_list.txt' txt to change coordinates and size according to the modified image.
Please refer to 5.1 _rscimg_merger_ui_list.txt for the file structure.

- Before modification

```
UI/Booting/Layer1/_LUT_Booting_Layer1.bin          255 0xFFFFFFFF
UI/Booting/Layer1/Booting_CI_RLE.bin               0   0xFFFFFFFF  500 265 375 108
```

- After modification

```
UI/Booting/Layer1/_LUT_Booting_Layer1.bin          255 0xFFFFFFFF
UI/Booting/Layer1/Booting_CI_RLE.bin               0   0xFFFFFFFF  410 295 455 85
```

(6) Run 'source\applications\Display\1280x720\ _rscimg_merger_ui.bat' and check below two files are updated.

- source\applications\Display\1280x720\ui_rscimg_720p.bin
- Image\ ui_rscimg_720p.bin

(7) Download binary image and check the modified image.

3.2. Add image

3.2.1. Sequence

(1) Before adding an image to a scene, you need to understand how to design graphics image and layer structure. You also need to understand which limitation exists in DU block.

(2) Convert image to be added into RLE compression format.

(3) Make UI resource image which includes the added image.

(4) Add information of added image to the image list of the target scene..

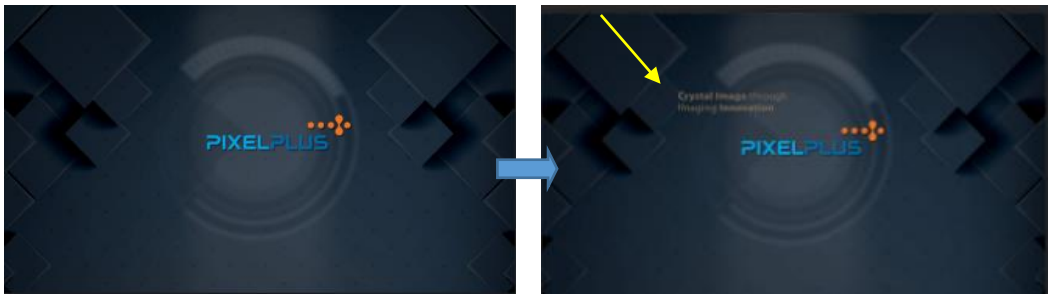
(5) Add code to display added image to the function which displays the target scene.

(6) Rebuild image.

(7) Make the binary image and download it to flash memory

3.2.2. Example

(1) This example shows how to add image to booting image.



(2) Image files for original booting image is as follows.

- source\applications\Display\1280x720\UI\Common\Layer4\Background.bmp

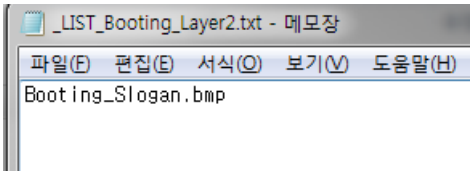
- source\applications\Display\1280x720\UI\Booting\Layer1\Booting_CI.bmp

(3) Now, add 'Booting_Slogan.bmp' to layer-2.

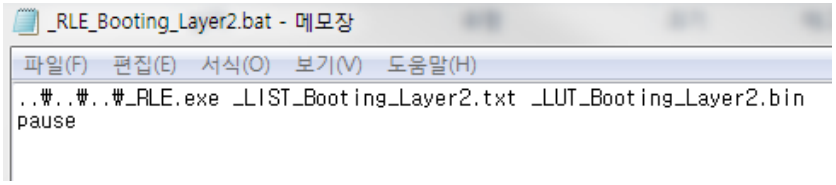
Make layer-2 folder and copy image file to the folder and make text and batch file as below.

source\applications\Display\1280x720\UI\Booting\Layer2\Booting_Slogan.bmp

source\applications\Display\1280x720\UI\Booting\Layer2\ _LIST_Booting_Layer2.txt



source\applications\Display\1280x720\UI\Booting\Layer2\ _RLE_Booting_Layer2.bat



(4) Run

'source\applications\Display\1280x720\UI\Booting\Layer2_RLE_Booting_Layer2.bat' and check below two files are created. If you want to add an image to layer which is already used, please make RLE based on LUT for the layer by adding image file information to text file of the layer

- source\applications\Dislay\1280x720\UI\Booting\Layer1_LUT_Booting_Layer2.bin
- source\applications\Display\1280x720\UI\Booting\Layer2\ Booting_Slogan_RLE.bin

(5) Add LUT and RLE information to

'source\applications\Display\1280x720_rscimg_merger_ui_list.txt'

Please refer to 5.1 _rscimg_merger_ui_list.txt for file structure.

```
UI/Booting/Layer1/_LUT_Booting_Layer1.bin          255 0xFFFFFFFF
UI/Booting/Layer1/Booting_CI_RLE.bin               0   0xFFFFFFFF  500 265 375 108
UI/Booting/Layer2/_LUT_Booting_Layer2.bin          255 0xFFFFFFFF
UI/Booting/Layer2/Booting_Slogan_RLE.bin           0   0xFFFFFFFF  336 193 294 67
```

(6) Run source\applications\Display\1280x720\ _rscimg_merger_ui.bat and check below 3

files are updated.

- source\applications\Display\1280x720\ui_rscimg_720p.bin
- source\applications\Display\ UI_rsclist.h
- Image\ ui_rscimg_720p.bin

(7) Add image and LUT information to the image list to be loaded for target scene.

source\tasks\Display\api_display.c

```
PP_RSC_LUT_S bootingImgLUT[] =          {
            {       0xFFFFFFFF,             0xFFFFFFFF          }
            ,{      eRSC_MODE_UI,           e_LUT_Booting_Layer1    }
            ,{      eRSC_MODE_UI,           e_LUT_Booting_Layer2    }
            ,{      0xFFFFFFFF,             0xFFFFFFFF          }
            ,{      eRSC_MODE_UI,           e_LUT_Common_Layer4     }
};
PP_RSC_UI_IMG_S bootingImg[] =          {
            {       eLayer1,    eArea0,     eBooting_CI_RLE,        PP_NULL                 },
            {       eLayer2,    eArea0,     eBooting_Slogan_RLE,    PP_NULL                 },
            {       eLayer4,    eArea0,     eBackground_RLE,        PP_NULL                 },
            {       eLayer_MAX,             eArea_MAX,              0xFFFFFFFF,     PP_NULL }
};
```

(8) Output an added image

Code to display added image has to be added to API

source\tasks\Display\api_display.c

```
PP_RESULT_E PPAPI_DISPLAY_BOOTING_CI (PP_VOID)
{
        … …

        img = PPAPI_DISPLAY_GetImg(eRSC_MODE_UI, eBooting_Slogan_RLE);
        if(!img)
                return eERROR_FAILURE;

        rect.u16X = img->info->x;
        rect.u16Y = img->info->y;
        rect.u16Width = img->info->width;
        rect.u16Height = img->info->height;
```

```
        stride = img->info->width;
        result = PPDRV_DU_OSD_SetAreaConfig(img->layer, img->area, img->buf, NULL, img->info->size, rect, stride, img->info->format);
        if(result == eSUCCESS)
                PPDRV_DU_OSD_EnableArea(img->layer, img->area, PP_TRUE);
        else
                PPDRV_DU_OSD_EnableArea(img->layer, img->area, PP_FALSE);

        return result;
}
```

(9) Rebuild SDK and download main binary to flash memory and check the added image after rebooting.

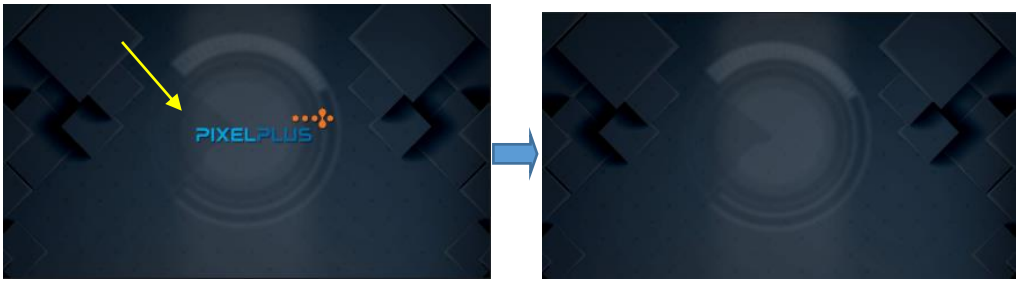
3.3. Remove image

3.3.1. Sequence

(1) Generate UI resource image which includes image to be removed.
(2) Remove target image from image list of a scene의 image list
(3) Delete code to display image to be removed.
(4) Rebuild image.
(5) Make binary image and download to it to flash memory.

3.3.2. Example

(1) This example shows how to remove an image from booting image.



(2) Image files to display original booting image is as follows.
- source\applications\Display\1280x720\UI\Common\Layer4\Background.bmp

- source\applications\Display\1280x720\UI\Booting\Layer1\Booting_CI.bmp

(3) Now, delete Booting_CI.bmp.

(4) Delete RLE and LUT of target image from source\applications\Display\1280x720_rscimg_merger_ui_list.txt. If there is an image which RLE refers LUT, LUT has not to be removed.

Please refer to 5.1 _rscimg_merger_ui_list.txt of 5. Appendix for structure.

```
UI/Booting/Layer1/_LUT_Booting_Layer1.bin        255        0xFFFFFFFF
UI/Booting/Layer1/Booting_CI_RLE.bin             0          0xFFFFFFFF        500 265 375 108
```

(5) Run source\applications\Display\1280x720\ _rscimg_merger_ui.bat and check whether below 3 files are updated.

- source\applications\Display\1280x720\ui_rscimg_720p.bin
- source\applications\Display\ UI_rsclist.h
- Image\ ui_rscimg_720p.bin

(6) Remove target image information from image list which will be loaded for the scene.

source\tasks\Display\api_display.c

```
PP_RSC_LUT_S bootingImgLUT[] =          {
            {       0xFFFFFFFF,                 0xFFFFFFFF      }
           ,{       eRSC_MODE_UI,     e_LUT_Booting_Layer1      }
           ,{       0xFFFFFFFF,                 0xFFFFFFFF      }
           ,{       0xFFFFFFFF,                 0xFFFFFFFF      }
           ,{       0xFFFFFFFF,                 0xFFFFFFFF      }
```

```
        ,{      eRSC_MODE_UI,   e_LUT_Common_Layer4     }
};
PP_RSC_UI_IMG_S bootingImg[] =      {
        {       eLayer1,  eArea0,   eBooting_CI_RLE,   PP_NULL          },
        {       eLayer4,  eArea0,   eBackground_RLE,   PP_NULL          },
        {       eLayer_MAX,         eArea_MAX,         0xFFFFFFFF,          PP_NULL          }
};
```

(7) Remove code to display the target image of API.

source\tasks\Display\api_display.c

```
PP_RESULT_E PPAPI_DISPLAY_BOOTING_CI (PP_VOID)
{
        img = PPAPI_DISPLAY_GetImg(eRSC_MODE_UI, eBooting_CI_RLE);
        if(!img)
                return eERROR_FAILURE;

        rect.u16X = img->info->x;
        rect.u16Y = img->info->y;
        rect.u16Width = img->info->width;
        rect.u16Height = img->info->height;
        stride = img->info->width;
        result = PPDRV_DU_OSD_SetAreaConfig(img->layer, img->area, img->buf, NULL, img->info->size, rect, stride, img->info->format);
        if(result == eSUCCESS)
                PPDRV_DU_OSD_EnableArea(img->layer, img->area, PP_TRUE);
        else
                PPDRV_DU_OSD_EnableArea(img->layer, img->area, PP_FALSE);

        return result;
}
```

(8) Rebuild SDK and download main binary to flash memory and check the image after rebooting.

3.4. Procedure to display an image

(1) Load image for target scene in advance.

PPAPI_DISPLAY_LoadImage()

(2) Set color look up table for each layer of target scene.

PPAPI_DISPlAY_UpdateLUT()

(3) Display an image to the target position.

PPAPI_DISPLAY_XXX () or PPAPI_DISPLAY_XXX_On()

(4) Do not display an image.

PPAPI_DISPLAY_DisableAll() or PPAPI_DISPLAY_XXX_Off()

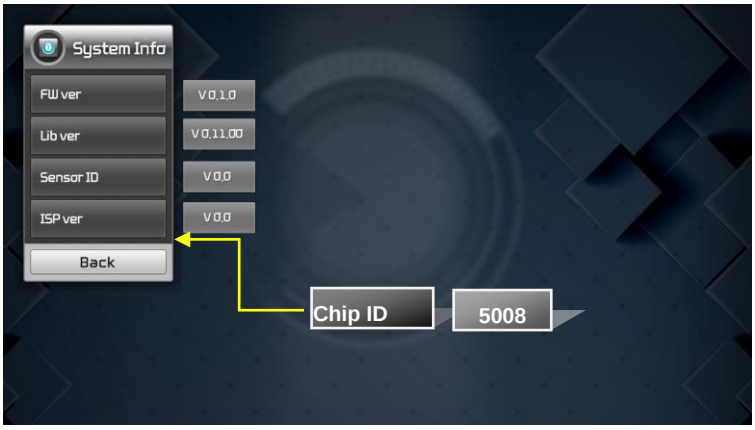
3.5. Add Menu

3.5.1. Sequence

(1) Add an item to menu list enumeration
(2) Modify menu structure for a scene menu to initialize submenu and setting value.
(3) Add code to implement the function of added item
(4) Add image for the added item.

3.5.2. Example

(1) There are five items in Menu > System Info menu: FW version / Lib version / Sensor ID / ISP version. This example shows how to add Chip ID item.



(2) Add eSystemInfo_ChipID which is the index of Chip ID item to menu list enumeration.

source\applications\Apps\apps.h

```
typedef enum ppSYSTEM_INFO_E {
        eSystemInfo_FWVer,
        eSystemInfo_LibVer,
        eSystemInfo_SencorID,
        eSystemInfo_ISPVer,
        eSystemInfo_ChipID,
        eSystemInfo_Back,

        eSystemInfo_Max
} PP_SYSTEM_INFO_E;
```

(3) Add an element for Chip ID to the System info menu structure and delete the default value to maintain the number of arrays. (?)

source\applications\Apps\apps.c

```
PP_SCENE_ELEM_S systemInfo = {
        eScene_SystemInfo,
        &Apps_UI_SystemInfo,
        eSystemInfo_Back,
        eSystemInfo_Max,
                { eSystemInfo_FWVer, 0, NULL_PTR },
                { eSystemInfo_LibVer, 0, NULL_PTR },
                { eSystemInfo_SencorID, 0, NULL_PTR },
                { eSystemInfo_ISPVer, 0, NULL_PTR },
                { eSystemInfo_ChipID, 0, NULL_PTR },
                { eSystemInfo_Back, 0xFFFFFFFF, NULL_PTR },
                { eSystemInfo_Max, 0xFFFFFFFF, NULL_PTR },
                { eSystemInfo_Max, 0xFFFFFFFF, NULL_PTR },
                { eSystemInfo_Max, 0xFFFFFFFF, NULL_PTR },
                { eSystemInfo_Max, 0xFFFFFFFF, NULL_PTR },
        }
};
```

(4) Add code to display sub menu for Chip ID item

source\tasks\Display\api_display.c

```
PP_RESULT_E PPAPI_DISPLAY_SubMenuList_SystemInfo (…)
{
        … …
        switch(elem->id)
        {
                case eSystemInfo_FWVer:
                        id = eFwVer_RLE;        break;
                case eSystemInfo_LibVer:
                        id = eLibVer_RLE;       break;
                case eSystemInfo_ISPVer:
                        id = eIspVer_RLE;       break;
                case eSystemInfo_SencorID:
                        id = eSensorId_RLE;     break;
                case eSystemInfo_ChipID:
                        id = eChipId_RLE;       break;

        }
        … …
}
```

(5) Add image for Chip ID.

Please refer to 3.2 Add image..

3.6. Remove Menu item

3.6.1. Sequence

(1) Delete target item in menu list enumeration

(2) Delete target element from menu structure for a scene menu

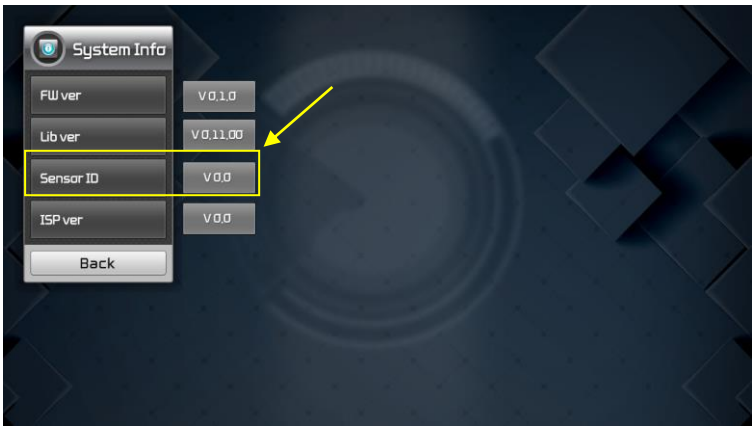
(3) Delete code for the item

(4) Remove image.

3.6.2. Example

There are five items in Menu > System Info menu. FW version / Lib version / Sensor ID / ISP version

This example shows how to delete Sensor ID item.



(1) Remove eSystemInfo_SencorID(Index for Sensor ID) from menu list enumeration for System Info.

source\applications\Apps\apps.h

```
typedef enum ppSYSTEM_INFO_E {
        eSystemInfo_FWVer,
        eSystemInfo_LibVer,
        eSystemInfo_SencorID,
        eSystemInfo_ISPVer,
        eSystemInfo_Back,

        eSystemInfo_Max
} PP_SYSTEM_INFO_E;
```

(2) Remove array of target item from the structure for System Info menu and add the default value to maintain the number of arrays.

source\applications\Apps\apps.c

```
PP_SCENE_ELEM_S systemInfo = {
        eScene_SystemInfo,
        &Apps_UI_SystemInfo,
        eSystemInfo_Back,
        eSystemInfo_Max,
                { eSystemInfo_FWVer, 0, NULL_PTR },
                { eSystemInfo_LibVer, 0, NULL_PTR },
                { eSystemInfo_SencorID, 0, NULL_PTR },
                { eSystemInfo_ISPVer, 0, NULL_PTR },
```

```
                { eSystemInfo_Back, 0xFFFFFFFF, NULL_PTR },
                { eSystemInfo_Max, 0xFFFFFFFF, NULL_PTR },
                { eSystemInfo_Max, 0xFFFFFFFF, NULL_PTR },
                { eSystemInfo_Max, 0xFFFFFFFF, NULL_PTR },
                { eSystemInfo_Max, 0xFFFFFFFF, NULL_PTR },
                { eSystemInfo_Max, 0xFFFFFFFF, NULL_PTR },
                { eSystemInfo_Max, 0xFFFFFFFF, NULL_PTR },
        }
};
```

(3) Delete code to output submenu for Sensor ID item.

source\tasks\Display\api_display.c

```
PP_RESULT_E PPAPI_DISPLAY_SubMenuList_SystemInfo (…)
{
        ……
        switch(elem->id)
        {
                case eSystemInfo_FWVer:
                        id = eFwVer_RLE;        break;
                case eSystemInfo_LibVer:
                        id = eLibVer_RLE;       break;
                case eSystemInfo_ISPVer:
                        id = eIspVer_RLE;       break;
                case eSystemInfo_SencorID:
                        id = eSensorId_RLE;     break;
        }
        … omission …
}
```

(4) Remove image for Sensor ID.

Please refer to 3.3 Remove image.

4. How to modify Low Layer

This is a guide for customers who want to design their own GUI. Although customers can develop their own GUI as they like, they have to use display driver directly and hence understand its function in depth

4.1. Sequence

(1) Design UI scenario and screen layout.

(2) Design graphic image according to UI scenario.

(3) UI & display structure for scenario and graphic image.

(4) Make image resource and download it to flash memory.

(5) Initialize DU block.

(6) Implement required functions for UI scenario.

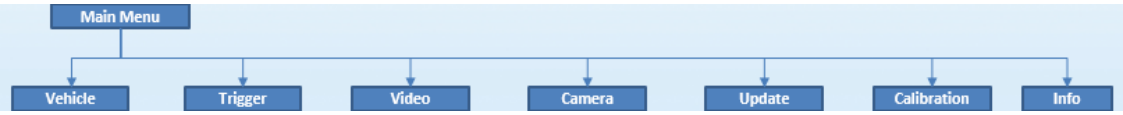
(7) Implement display part of each function for UI scenario.

4.2. Example

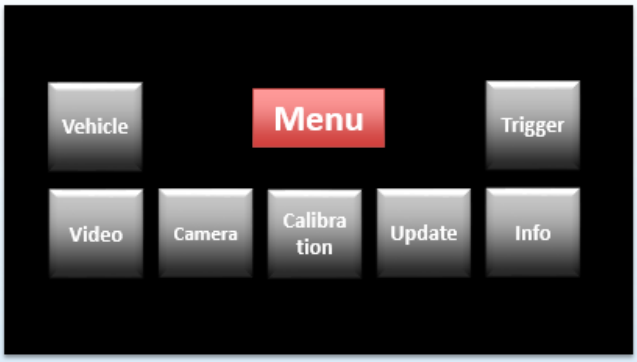
Pixelplus GUI is used as a reference.

(1) Design UI scenario and screen layout.

- Flow Chart



- Scenario

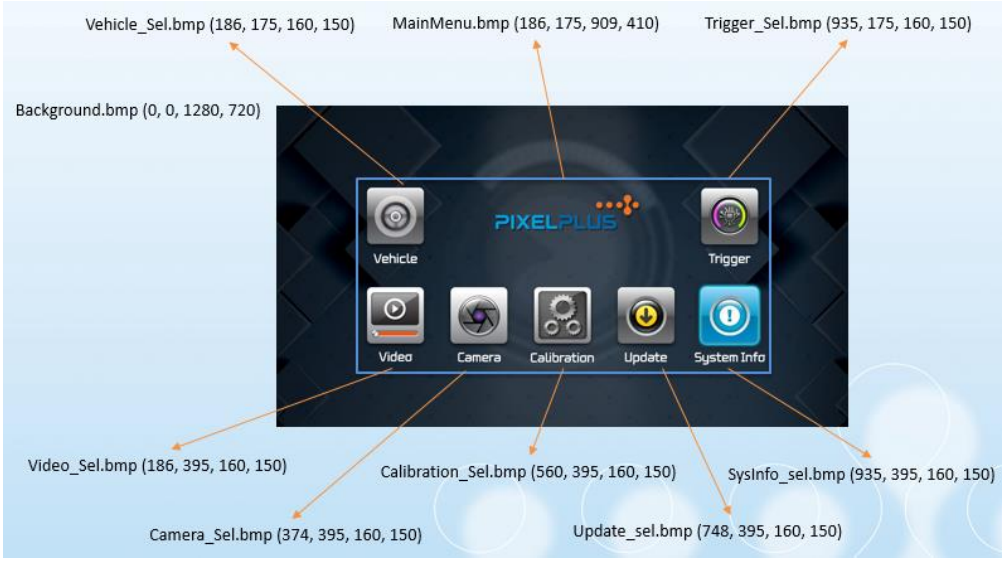


Main Menu	Sub Menu	Key	Range
Select Item	Vehicle / Trigger / Video / Camera / Update / Calibration / Info	← / →	7
To Live View		menu	

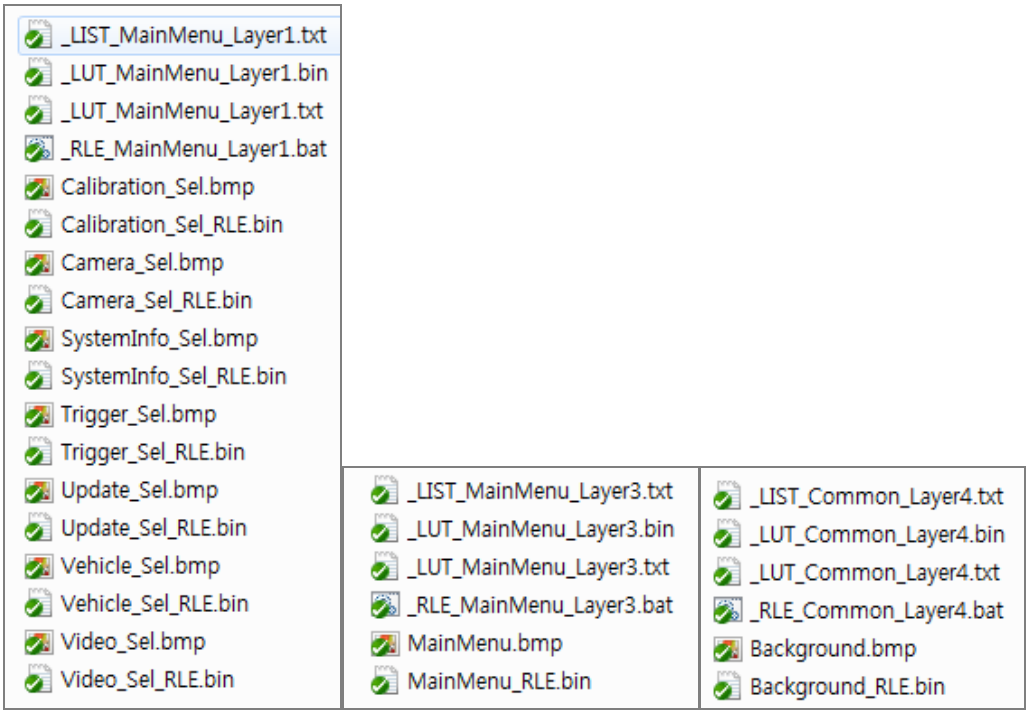
(2) Design graphic image according to UI scenario.

(3) Design UI & display structure according to UI scenario and graphic design.

Scene	API	OSD list		Layer	Area
Main Menu	PPAPI_DISPLAY_POPUP_On	Popup		Layer 0	Area 0
	PPAPI_DISPLAY_MenuItem	Select List Item	vehicle / trigger / video / camera / calibration / update / system info	Layer 1	Area 0
	PPAPI_DISPLAY_MenuList	Menu List		Layer 3	Area 0
	PPAPI_DISPLAY_Background_On	Background		Layer 4	Area 0



(4) Make bmp images according to display structure and convert them into Color LUT and RLE Binary.

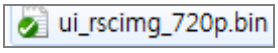


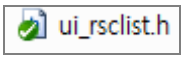
(5) Generate resource binary to be written to flash memory.

A. Create _rscimg_merger_ui_list.txt file which includes resource list to make resource binary. Please refer to 5. Appendix for more detail.

```
UI/Common/Layer4/_LUT_Common_Layer4.bin                255
UI/Common/Layer4/Background_RLE.bin                    0   255 0 0 1280 720
UI/MainMenu/Layer1/_LUT_MainMenu_Layer1.bin            255
UI/MainMenu/Layer1/Vehicle_Sel_RLE.bin                 0   255 186 175 160 150
UI/MainMenu/Layer1/Trigger_Sel_RLE.bin                 0   255 935 175 160 150
UI/MainMenu/Layer1/Video_Sel_RLE.bin                   0   255 186 395 160 150
UI/MainMenu/Layer1/Camera_Sel_RLE.bin                  0   255 374 395 160 150
UI/MainMenu/Layer1/Calibration_Sel_RLE.bin             0   255 560 395 160 150
UI/MainMenu/Layer1/Update_Sel_RLE.bin                  0   255 748 395 160 150
UI/MainMenu/Layer1/SystemInfo_Sel_RLE.bin              0   255 935 395 160 150
UI/MainMenu/Layer3/_LUT_MainMenu_Layer3.bin            255
UI/MainMenu/Layer3/MainMenu_RLE.bin                    0   255 186 175 909 410
```

B. Resource enumeration header file will also be generated when resource binary is made..





(6) Make flash image including ui_rscimg_720p.bin using ‘Image\PicassoDownloadTool_ver0.5_TestVersion.exe’ tool and download it to flash memory.

(7) Initialize DU block.
Please refer to PPAPI_DISPlAY_Initialize() of api_display.c. DU has to be initialized after SVM block is initialized

A. Decide Video Path.
PPDRV_DU_SetVideoPath()

B. Set Mixer Path.
PPDRV_DU_OSD_SetMixerPath()

C. Set Mixer Size according to resolution.
PPDRV_DU_OSD_SetLayerSize()

D. Set format of each Layer.
PPDRV_DU_OSD_SetLayerFormat()

E. Set color of each Layer.
PPDRV_DU_OSD_SetLayerColor()

F. Activate DU block.
PPDRV_DU_OSD_RunMixer()

G. Set and output BTO.

PPDRV_DU_BTO_SetXXX()

(8) Implement U

Implement the action of each event according to UI scenario.

Please refer to vTaskUI() of app_ui.c.

```
switch(queueItem.u32Cmd)
{
    case CMD_UI_KEY_UP:
    case CMD_UI_KEY_DOWN:
    case CMD_UI_KEY_LEFT:
    case CMD_UI_KEY_RIGHT:
    case CMD_UI_KEY_MENU:
        Apps_UI_MainMenu();
        break;
}
```

(9) Load resource from flash memory.

Please refer to 5. Appendix for flash memory map.

Please refer to PPAPI_DISPLAY_LoadHeader() and PPAPI_DISPLAY_LoadImage() of api_display.c for loading and parsing example.

There are two APIs to load resource

PPAPI_FLASH_Read()

PPAPI_FLASH_ReadQDMA()

There is a condition to use these APIs

A. Address and size of dram buffer has to be 16 byte aligned

B. Flash address has to be 256byte aligned.

(10) Implement display functions according to UI Scenario.

Please refer to PPAPI_DISPLAY_Background_On() of api_display.c.

A. Set Color LUT

PPDRV_DU_OSD_SetColorLut()

B. Set Area

PPDRV_DU_OSD_SetAreaConfig()

C. Display Area.

PPDRV_DU_OSD_EnableArea()

```
PP_VOID Apps_UI_MainMenu (PP_U32 IN event)
{
    PP_RECT_S rect;
    PP_U32 stride;
    PP_RESULT_E result;

    switch(event)
    {
        case CMD_UI_KEY_CENTER:
            PPDRV_DU_OSD_SetColorLut(eLayer4, (PP_U32 *)&lut);
            rect.u16X = x;
            rect.u16Y = y;
            rect.u16Width = w;
            rect.u16Height = h;
            stride = w;
            result = PPDRV_DU_OSD_SetAreaConfig(eLayer4, eArea0, imgBuf, NULL, size, rect, stride, eFORMAT_RLE);
            if(result == eSUCCESS)
                PPDRV_DU_OSD_EnableArea(eLayer4, eArea0, PP_TRUE);
            else
                PPDRV_DU_OSD_EnableArea(eLayer4, eArea0, PP_FALSE);
            break;
    }
}
```

5. Appendix

5.1. _rscimg_merger_ui_list.txt structure

[Path]	[Format] [Filed]	[x] [y] [width] [height]

[Format]

255	Color LUT
0	RLE
1	INDEX
2	RGB565
3	RGB888
4	ARGB8888
5	RGBA4444

[Filed]

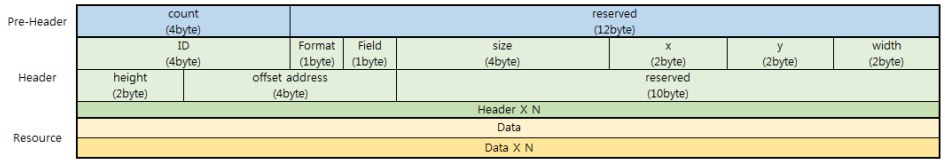
0xFFFFFFFF	Invalid
0	Even
1	Odd

- Example

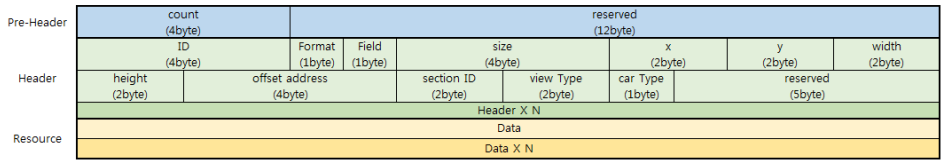
UI/Common/Layer4/_LUT_Common_Layer4.bin	255	0xFFFFFFFF	
UI/Common/Layer4/Background_RLE.bin	0	0xFFFFFFFF	0 0 1280 720
UI/Common/Layer4/Outline_RLE.bin	0	0xFFFFFFFF	0 0 1280 720

5.2. ui_rscimg_720p.bin/Car_Image.bin/PGL_Image.bin structure

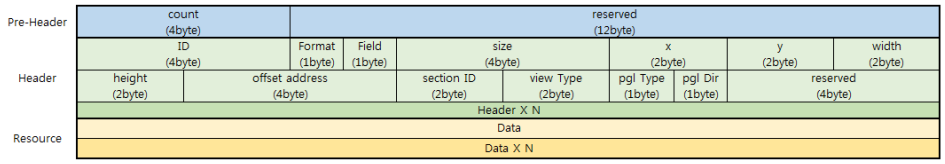
UI



CAR



PGL



Format	
0	RLE
1	INDEX
2	RGB565
3	RGB888
4	ARGB8888
5	RGBA4444
6	1BIT
255	LUT

Field	
0	even
1	odd
255	none

Car Type	
0	car
1	shadow
2	wheel
3	door

PGL Type	
0	static PGL
1	dynamic PGL

PGL Direction	
0	bw
1	fw

6. Revision History

Version	Date	Description
v 0.1	2018.04.06	Draft
v 0.11	2018.06.04	Renewal & Update
v 0.2	2018.07.31	Update
v 0.3	2018.11.16	Update