Display

PI5008K

API

Rev 0.5

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 Type

```
typedef void                                  PP_VOID;

typedef char                                  PP_CHAR;

typedef unsigned char                         PP_U8;

typedef short                                 PP_S16;

typedef unsigned short                        PP_U16;

typedef int                                   PP_S32;

typedef unsigned int                          PP_U32;

typedef enum {PP_FALSE, PP_TRUE}              PP_BOOL;


#define null                                  PP_NULL

#define (-1)                                  PP_INVALID

#define (-1)                                  PP_FAILURE

#define (0L)                                  PP_SUCCESS
```

2 Enumeration

2.1 PP_SCENE_E

[Syntax]

```
typedef enum ppSCENE_E {
            eScene_Booting,
            eScene_LiveView,
            eScene_MainMenu,
            eScene_VehicleSetting,
            eScene_TriggerSetting,
            eScene_VideoSetting,
            eScene_CameraSetting,
            eScene_CameraPreview,
            eScene_CameraCapture,
            eScene_CalibSetting,
            eScene_ManualCalib,
            eScene_MovePoint,
            eScene_ViewGen,
            eScene_UpdateSetting,
            eScene_SystemInfo,
            eScene_Dialog,

            eScene_Max
} PP_SCENE_E;
```

[Description]

Scene enumeration

[Member]

Member	Description
eScene_Booting	Booting scene
eScene_LiveView	Live view scene
eScene_MainMenu	Main menu scene
eScene_VehicleSetting	Vehicle setting scene
eScene_TriggerSetting	Trigger setting scene

eScene_VideoSetting	Video setting scene
eScene_CameraSetting	Camera setting scene
eScene_CameraPreview	Front/Right/Left/Rear/Quad Preview menu scene of camera setting
eScene_CameraCapture	Capture menu scene of camera setting
eScene_CalibSetting	Auto/Manual calibration menu scene 및 Auto calibration scene
eScene_ManualCalib	Manual calibration scene
eScene_MovePoint	Move point scene of Manual calibration
eScene_ViewGen	View generation scene of Manual calibration
eScene_UpdateSetting	Update setting scene
eScene_SystemInfo	System info scene
eScene_Dialog	Dialog scene

2.2 PP_VIEW_MODE_E

[Syntax]

```
typedef enum ppVIEW_MODE_E {
        eViewMode_TopCam,
        eViewMode_Top3D,

        eViewMode_Max
} PP_VIEW_MODE_E;
```

[Description]

View Mode enumeration of Live View Scene

[Member]

Member	Description
eViewMode_TopCam	2D Top View + Cam View
eViewMode_Top3D	2D Top View + 3D View

2.3 PP_VIEW_MODE_TOPCAM_E

[Syntax]

```
typedef enum ppVIEW_MODE_TOPCAM_E {
        eViewMode_TopCam_Front,
        eViewMode_TopCam_Left,
        eViewMode_TopCam_Right,
        eViewMode_TopCam_Rear,
        eViewMode_TopCam_WideFront,
        eViewMode_TopCam_WideRear,

        eViewMode_TopCam_Max
} VIEW_MODE_TOPCAM_E;
```

[Description]

Cam View Mode enumeration of 2D Top View + Cam View Mode

[Member]

Member	Description
eViewMode_TopCam_Front	Front cam view
eViewMode_TopCam_Left	Left cam view
eViewMode_TopCam_Right	Right cam view
eViewMode_TopCam_Rear	Rear cam view
eViewMode_TopCam_WideFront	Wide front cam view
eViewMode_TopCam_WideRear	Wide rear cam view

2.4 PP_VIEW_MODE_TOP3D_E

[Syntax]

```
typedef enum ppVIEW_MODE_TOP3D_E {
        eViewMode_Top3D_Swing_0,
        eViewMode_Top3D_Swing_1,
        eViewMode_Top3D_Swing_2,
        eViewMode_Top3D_Swing_3,
        eViewMode_Top3D_Swing_4,
        eViewMode_Top3D_Swing_5,
        eViewMode_Top3D_Swing_6,
        eViewMode_Top3D_Swing_7,

        eViewMode_Top3D_Max
} PP_VIEW_MODE_TOP3D_E;
```

[Description]

3D View Mode enumeration of 2D Top View + 3D View Mode

[Member]

Member	Description
eViewMode_Top3D_Swing_0	swing 0 3d view mode
eViewMode_Top3D_Swing_1	swing 1 3d view mode
eViewMode_Top3D_Swing_2	swing 2 3d view mode
eViewMode_Top3D_Swing_3	swing 3 3d view mode
eViewMode_Top3D_Swing_4	swing 4 3d view mode
eViewMode_Top3D_Swing_5	swing 5 3d view mode
eViewMode_Top3D_Swing_6	swing 6 3d view mode
eViewMode_Top3D_Swing_7	swing 7 3d view mode

2.5 PP_POPUP_E

[Syntax]

```
typedef enum ppPOPUP_E {
        ePopUp_DoNotOff,                // "Do not turn off the power"
        ePopUp_SaveDone,                // "Save Done"
        ePopUp_NoSaveDone,              // "Save Failed"
        ePopUp_NoInput,                 // "No Camera Input"
        ePopUp_NoSDCard,                // "No SD Card"
        ePopUp_ShutDown,                // "System Shutdown"
        ePopUp_Calibration,             // "Calibration in Progress..."
        ePopUp_ViewGen,                 // "View Generation in Progress..."
        ePopUp_Done,                    // "Done!"
        ePopUp_Failed,                  // "Failed!"
        ePopUp_NoUpdate,                // "FW Update Filed!"
        ePopUp_Max
} PP_POPUP_E;
```

[Description]

팝업 메시지

[Member]

Member	Description
ePopUp_DoNotOff	“Do not turn off the power”
ePopUp_SaveDone	“Save done”
ePopUp_NoSaveDone	“Save Failed”
ePopUp_NoInput	“No Camera Input”
ePopUp_NoSDCard	“No SD Card”
ePopUp_ShutDown	“System Shutdown”
ePopUp_Calibration	“Calibration in process”
ePopUp_ViewGen	“View generation in process”
ePopUp_Done	“Done”
ePopUp_Failed	“Failed!”
ePopUp_NoUpdate	“FW Update Failed!”

2.6 PP_DIALOG_BOX_E

[Syntax]

```
typedef enum ppDIALOG_BOX_E {
            eDialog_ManualCalib,            // "Calibrate manually?"
            eDialog_SaveView,               // "Save view?"
            eDialog_CalibFailRetry,         // "Calibration failed. Retry?"
            eDialog_RunMenu,                // "Run the selected menu?"
            eDialog_Update,                 // "Update FW?"

            eDialog_Max
} PP_DIALOG_BOX_E;
```

[Description]

Dialog box message

[Member]

Member	Description
eDialog_ManualCalib	“Calibrate manually?”
eDialog_SaveView	“Save view?”
eDialog_CalibFailRetry	“Calibration failed. Retry?”

eDialog_RunMenu	“Run the selected menu?”
eDialog_Update	“Update FW?”

2.7 PP_DIALOG_BTN_E

[Syntax]

```
typedef enum ppDIALOG_BTN_E {
            eDialog_Btn_Ok,
            eDialog_Btn_Cancel,

            eDialog_Btn_Max
} PP_DIALOG_BTN_E;
```

[Description]

Button of Dialog box

[Member]

Member	Description
eDialog_Btn_Ok	“OK” button
eDialog_Btn_Cancel	“CANCEL” button

2.8 PP_RSC_MODE_E

[Syntax]

```
typedef enum ppRSC_MODE_E {
            eRSC_MODE_CAR,
            eRSC_MODE_UI,
            eRSC_MODE_PGL,

            eRSC_MODE_MAX
} PP_RSC_MODE_E;
```

[Description]

Resource Mode

[Member]

Member	Description
eRSC_MODE_CAR	Car Image
eRSC_MODE_UI	UI Image
eRSC_MODE_PGL	PGL Image

2.9 PP_RSC_CAR_TYPE_E

[Syntax]

```
typedef enum ppRSC_CAR_TYPE_E {
        eRSC_CAR_TYPE_CAR,
        eRSC_CAR_TYPE_SHADOW,
        eRSC_CAR_TYPE_WHEEL,
        eRSC_CAR_TYPE_DOOR,

        eRSC_CAR_TYPE_MAX
} PP_RSC_CAR_TYPE_E;
```

[Description]

Car resource type enumeration in flash

[Member]

Member	Description
eRSC_CAR_TYPE_CAR	Car image
eRSC_CAR_TYPE_SHADOW	Shadow image
eRSC_CAR_TYPE_WHEEL	Wheel image
eRSC_CAR_TYPE_DOOR	Door image

2.10 PP_RSC_PGL_TYPE_E

[Syntax]

```
typedef enum ppRSC_PGL_TYPE_E {
        eRSC_PGL_TYPE_STATIC,
        eRSC_PGL_TYPE_DYNAMIC,

        eRSC_PGL_TYPE_MAX
} PP_RSC_PGL_TYPE_E;
```

[Description]

PGL resource type enumeration in flash

[Member]

Member	Description
eRSC_PGL_TYPE_STATIC	Static PGL type
eRSC_PGL_TYPE_DYNAMIC	Dynamic PGL type

2.11 PP_RSC_PGL_DIR_E

[Syntax]

```
typedef enum ppRSC_PGL_DIR_E {
        eRSC_PGL_DIR_BACKWARD,
        eRSC_PGL_DIR_FORWARD,

        eRSC_PGL_DIR_MAX
} PP_RSC_PGL_DIR_E;
```

[Description]

Car resource direction enumeration in flash

[Member]

Member	Description
eRSC_PGL_DIR_BACKWARD	backward direction pgl
eRSC_PGL_DIR_FORWARD	forward direction pgl

3 Structure

3.1 PP_POS_S

[Syntax]

```
    typedef struct ppPOS_S
    {
        PP_U16   u16X;
        PP_U16   u16Y;
} PP_POS_S;
```

[Description]

Structure for coordinate

[Member]

Member	Description
u16X	x coordinate
u16Y	y coordinate

3.2 PP_SCENE_SUB_ELEM_S

[Syntax]

```
    typedef struct ppSCENE_SUB_ELEM_S {
        PP_U32          id;
        PP_U32          valueNum;
        PP_U32          *value;
} PP_SCENE_SUB_ELEM_S;
```

[Description]

Structure for menu information

[Member]

Member	Description
id	sub menu id
valueNum	Max value of sub menu
*value	Value of sub menu

3.3 PP_RSC_INFO_S

[Syntax]

```
typedef struct ppRSC_INFO_S {
            PP_U32 id;
            PP_U8 format;
            PP_U32 field;
            PP_U32 size;
            PP_U16 x;
            PP_U16 y;
            PP_U16 width;
            PP_U16 height;
            PP_U32 offset;
            PP_U16 sectionID;
            PP_U16 viewType;
            PP_U8 type;         // car or pgl
            PP_U8 dir;          // pgl dir
            PP_U32 *buf;
} PP_RSC_INFO_S;
```

[Description]

Resource Info

This structure is used to hold information of header for display resource.

UI

<table>
<tr><td>Pre-Header</td><td colspan="2">count
(4byte)</td><td colspan="8">reserved
(12byte)</td></tr>
<tr><td rowspan="3">Header</td><td colspan="2">ID
(4byte)</td><td>Format
(1byte)</td><td>Field
(1byte)</td><td colspan="2">size
(4byte)</td><td colspan="2">x
(2byte)</td><td>y
(2byte)</td><td>width
(2byte)</td></tr>
<tr><td>height
(2byte)</td><td colspan="3">offset address
(4byte)</td><td colspan="6">reserved
(10byte)</td></tr>
<tr><td colspan="10">Header X N</td></tr>
<tr><td rowspan="2">Resource</td><td colspan="10">Data</td></tr>
<tr><td colspan="10">Data X N</td></tr>
</table>

CAR

<table>
<tr><td>Pre-Header</td><td colspan="2">count
(4byte)</td><td colspan="8">reserved
(12byte)</td></tr>
<tr><td rowspan="3">Header</td><td colspan="2">ID
(4byte)</td><td>Format
(1byte)</td><td>Field
(1byte)</td><td colspan="2">size
(4byte)</td><td colspan="2">x
(2byte)</td><td>y
(2byte)</td><td>width
(2byte)</td></tr>
<tr><td>height
(2byte)</td><td colspan="3">offset address
(4byte)</td><td>section ID
(2byte)</td><td>view Type
(2byte)</td><td>car Type
(1byte)</td><td colspan="3">reserved
(5byte)</td></tr>
<tr><td colspan="10">Header X N</td></tr>
<tr><td rowspan="2">Resource</td><td colspan="10">Data</td></tr>
<tr><td colspan="10">Data X N</td></tr>
</table>

PGL

<table>
<tr><td>Pre-Header</td><td colspan="2">count
(4byte)</td><td colspan="8">reserved
(12byte)</td></tr>
<tr><td rowspan="3">Header</td><td colspan="2">ID
(4byte)</td><td>Format
(1byte)</td><td>Field
(1byte)</td><td colspan="2">size
(4byte)</td><td colspan="2">x
(2byte)</td><td>y
(2byte)</td><td>width
(2byte)</td></tr>
<tr><td>height
(2byte)</td><td colspan="3">offset address
(4byte)</td><td>section ID
(2byte)</td><td>view Type
(2byte)</td><td>pgl Type
(1byte)</td><td>pgl Dir
(1byte)</td><td colspan="2">reserved
(4byte)</td></tr>
<tr><td colspan="10">Header X N</td></tr>
<tr><td rowspan="2">Resource</td><td colspan="10">Data</td></tr>
<tr><td colspan="10">Data X N</td></tr>
</table>

[Member]

Member	Description
id	id
format	format
field	field (even / odd)
size	resource size
x	x coordinate
y	y coordinate
width	width
height	height
offset	Offset where resource is located
sectionID	section ID (by SVM view)
viewType	view Type (by SVM view)
type	(car case) car type (car / shadow / wheel / door) (pgl case) pgl Type (static / dynamic)
dir	Parking Guide Line Direction (forward / backward)
buf	resource data

3.4 PP_RSC_HEADER_S

[Syntax]

```
typedef struct ppRSC_HEADER_S {
            PP_U32                      num;
            PP_U32                      reserved[3];
            PP_RSC_INFO_S               *info;
} PP_RSC_HEADER_S;
```

[Description]

Resource header

This structure is used to hold information of header and pre-header for display resource.

UI

<table>
<tr><td>Pre-Header</td><td colspan="2">count
(4byte)</td><td colspan="8">reserved
(12byte)</td></tr>
<tr><td rowspan="3">Header</td><td colspan="2">ID
(4byte)</td><td>Format
(1byte)</td><td>Field
(1byte)</td><td colspan="2">size
(4byte)</td><td colspan="2">x
(2byte)</td><td>y
(2byte)</td><td>width
(2byte)</td></tr>
<tr><td>height
(2byte)</td><td colspan="3">offset address
(4byte)</td><td colspan="6">reserved
(10byte)</td></tr>
<tr><td colspan="10">Header X N</td></tr>
<tr><td rowspan="2">Resource</td><td colspan="10">Data</td></tr>
<tr><td colspan="10">Data X N</td></tr>
</table>

CAR

<table>
<tr><td>Pre-Header</td><td colspan="2">count
(4byte)</td><td colspan="8">reserved
(12byte)</td></tr>
<tr><td rowspan="3">Header</td><td colspan="2">ID
(4byte)</td><td>Format
(1byte)</td><td>Field
(1byte)</td><td colspan="2">size
(4byte)</td><td colspan="2">x
(2byte)</td><td>y
(2byte)</td><td>width
(2byte)</td></tr>
<tr><td>height
(2byte)</td><td colspan="3">offset address
(4byte)</td><td>section ID
(2byte)</td><td>view Type
(2byte)</td><td>car Type
(1byte)</td><td colspan="3">reserved
(5byte)</td></tr>
<tr><td colspan="10">Header X N</td></tr>
<tr><td rowspan="2">Resource</td><td colspan="10">Data</td></tr>
<tr><td colspan="10">Data X N</td></tr>
</table>

PGL

<table>
<tr><td>Pre-Header</td><td colspan="2">count
(4byte)</td><td colspan="8">reserved
(12byte)</td></tr>
<tr><td rowspan="3">Header</td><td colspan="2">ID
(4byte)</td><td>Format
(1byte)</td><td>Field
(1byte)</td><td colspan="2">size
(4byte)</td><td colspan="2">x
(2byte)</td><td>y
(2byte)</td><td>width
(2byte)</td></tr>
<tr><td>height
(2byte)</td><td colspan="3">offset address
(4byte)</td><td>section ID
(2byte)</td><td>view Type
(2byte)</td><td>pgl Type
(1byte)</td><td>pgl Dir
(1byte)</td><td colspan="2">reserved
(4byte)</td></tr>
<tr><td colspan="10">Header X N</td></tr>
<tr><td rowspan="2">Resource</td><td colspan="10">Data</td></tr>
<tr><td colspan="10">Data X N</td></tr>
</table>

[Member]

Member	Description
num	Number of resource
reserved	Reserved
info	resource info (exists as many as num)

3.5 PP_RSC_LUT_S

[Syntax]

```
typedef struct ppRSC_LUT_S {
            PP_RSC_MODE_E           mode;
            PP_U32                  lut_id;
} PP_RSC_LUT_S;
```

[Description]

This structure is used to define Color LUT of RLE for each layer

[Member]

Member	Description
mode	UI / CAR / PGL mode
lut_id	LUT id

3.6 PP_RSC_UI_IMG_S

[Syntax]

```
typedef struct ppRSC_UI_IMG_S {
            PP_DU_LAYER_E           layer;
            PP_DU_AREA_E            area;
            PP_U32                  id;
            PP_RSC_INFO_S           *info;
} PP_RSC_UI_IMG_S;
```

[Description]

Structure specific for UI Display

[Member]

Member	Description
layer	layer to be displayed
area	area to be displayed
id	Resource id
info	Resource info

3.7 PP_RSC_IMG_S

[Syntax]

```
typedef struct ppRSC_IMG_S {
            PP_DU_LAYER_E           layer;
            PP_DU_AREA_E            area;
} PP_RSC_IMG_S;
```

[Description]

Structure for overall display

[Member]

Member	Description
layer	layer to be displayed
area	area to be displayed

3.8 PP_CAR_SECTION_S

[Syntax]

```
typedef struct ppCAR_SECTION_S {
            PP_RSC_INFO_S           *car;
            PP_RSC_INFO_S           *shadow;
            PP_U32                  wheelNum;
            PP_RSC_INFO_S           *wheel[CAR_WHEEL_MAX];
} PP_CAR_SECTION_S;
```

[Description]

Structure to manage car image of section view

[Member]

Member	Description
car	Pointer of car image info of target section
shadow	Pointer of shadow image info of target section
wheelNum	number of car wheel image of target section
wheel	Pointer of car wheel image info of target section

3.9 PP_PGL_SECTION_S

[Syntax]

```
typedef struct ppPGL_SECTION_S {
            PP_RSC_INFO_S           *bwStatic;
            PP_U32                  bwDynamicNum;
            PP_RSC_INFO_S           *bwDynamic[71];

            PP_RSC_INFO_S           *fwStatic;
            PP_U32                  fwDynamicNum;
            PP_RSC_INFO_S           *fwDynamic[71];
} PP_PGL_SECTION_S;
```

[Description]

Structure to manage PGL image for a section view

[Member]

Member	Description
bwStatic	Pointer of backward static pgl image
bwDynamicNum	Number of backward dynamic pgl image
bwDynamic	Pointer of backward dynamic pgl image
fwStatic	Pointer of forward static pgl image
fwDynamicNum	Number of forward dynamic pgl image
fwDynamic	Pointer of forward dynamic plg image

4 Variable

4.1 gCarRsc

[Syntax]

```
PP_RSC_HEADER_S gCarRsc
```

[Description]

Car Mode Image resource

4.2 gUiRsc

[Syntax]

```
PP_RSC_HEADER_S gUiRsc
```

[Description]

UI Mode Image resource

4.3 gPglRsc

[Syntax]

```
PP_RSC_HEADER_S gPglRsc;
```

[Description]

PGL Mode Image resource

4.4 gRscLut

[Syntax]

```
PP_RSC_LUT_S *gRscLut;
```

[Description]

Pointer variable for color LUT

4.5 gUiImg

[Syntax]

```
PP_RSC_UI_IMG_S         *gUiImg;
```

[Description]

Pointer variable for UI Image

4.6 gCarSection

[Syntax]

```
PP_CAR_SECTION_S      *gCarSection;
```

[Description]

Management variable of car image of a section

4.7 gPglSection

[Syntax]

```
PP_PGL_SECTION_S      *gPglSection;
```

[Description]

Section 에 의한 Pgl Image 의 Management variable

4.8 wheelAngle

[Syntax]

```
PP_U32 wheelAngle;
```

[Description]

wheel image index for wheel animation

Range : 0~3

4.9 gCarWheelBase

[Syntax]

```
PP_CAR_SECTION_S *gCarWheelBase;
```

[Description]

car image pointer for Car wheel animation

4.10 gPgl2dBase

[Syntax]

```
PP_RSC_INFO_S *gPgl2dBase;
```

[Description]

Pointer of PGL image of Top2D for PGL animation

4.11 gPglRearCamBase

[Syntax]

```
PP_RSC_INFO_S *gPglRearCamBase;
```

[Description]

Pointer of PGL image for rearcam for PGL animation

4.12 gCarDoor

[Syntax]

```
PP_RSC_INFO_S *gCarDoor[eCar_OpenDoor_Max]
```

[Description]

Car Door info pointer

4.13 gExistFlashRsc

[Syntax]

```
PP_BOOL gExistFlashRsc
```

[Description]

Variable to decide whether image resource exists in Flash

5 Function

5.1 PPAPI_DISPLAY_Initialize

[Syntax]

```
PP_RESULT_E PPAPI_DISPlAY_Initialize (PP_VOID);
```

[Description]

This API is used to Initialize display function. It should be called before using any display API.

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	Initialization success
eERROR_FAILURE	Initialization fail

5.2 PPAPI_DISPLAY_LoadHeader

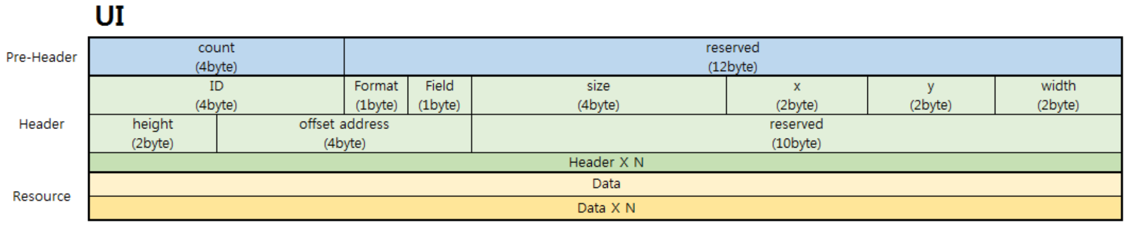
[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_LoadHeader (PP_VOID);
```

[Description]

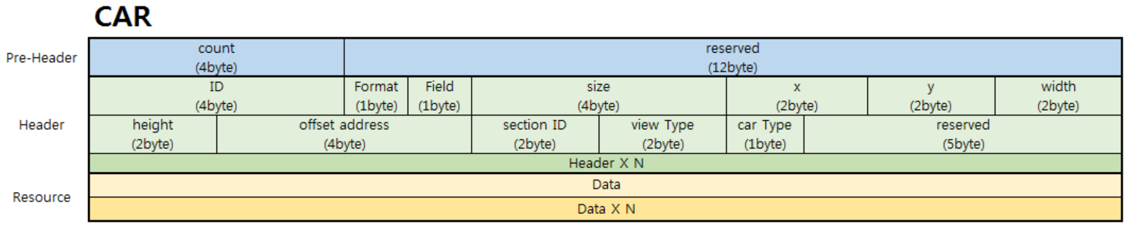
Structure of Display Binary is as below. After loading all Pre-Header, Header information, the information will be parsed.

Only some resources related to booting image and color LUT are loaded from flash and parsed according to structure.



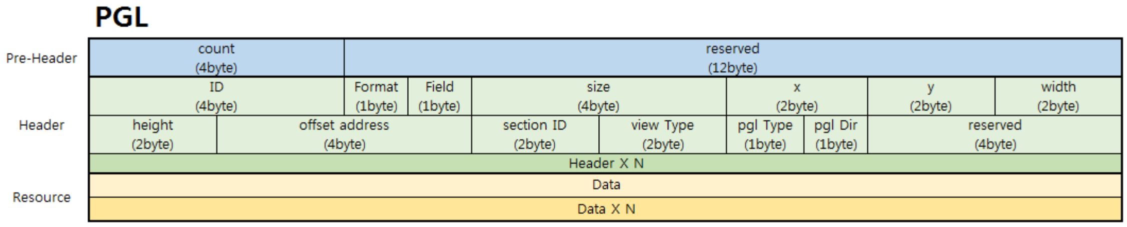
UI

Pre-Header	count (4byte)	reserved (12byte)					
Header	ID (4byte)	Format (1byte)	Field (1byte)	size (4byte)	x (2byte)	y (2byte)	width (2byte)
	height (2byte)	offset address (4byte)	reserved (10byte)				
	Header X N						
Resource	Data						
	Data X N						



CAR

Pre-Header	count (4byte)	reserved (12byte)					
Header	ID (4byte)	Format (1byte)	Field (1byte)	size (4byte)	x (2byte)	y (2byte)	width (2byte)
	height (2byte)	offset address (4byte)	section ID (2byte)	view Type (2byte)	car Type (1byte)	reserved (5byte)	
	Header X N						
Resource	Data						
	Data X N						



PGL

Pre-Header	count (4byte)	reserved (12byte)						
Header	ID (4byte)	Format (1byte)	Field (1byte)	size (4byte)	x (2byte)	y (2byte)	width (2byte)	
	height (2byte)	offset address (4byte)	section ID (2byte)	view Type (2byte)	pgl Type (1byte)	pgl Dir (1byte)	reserved (4byte)	
	Header X N							
Resource	Data							
	Data X N							

This API can be used without initializing display but it can be used only after Flash initialize(FLASHAPI_initialize(), FLASHAPI_read_header()). This API needs to be called only once after booting.

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Fail

5.3 PPAPI_DISPLAY_LoadUiImage

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_LoadUiImage (PP_U32 IN idx)
```

[Description]

UI image will be loaded from flash and parsed according to the structure.

To run this API, PPAPI_DISPLAY_LoadHeader() has to be run in advance.

[Parameter]

Member	Description
idx	resource index to loading (PP_UI_RSCLIST_E)

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Fail

5.4 PPAPI_DISPLAY_LoadCarImage

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_LoadCarImage (PP_U32 IN idx)
```

[Description]

Car image will be loaded from flash and parsed according to the structure.

To run this API, PPAPI_DISPLAY_LoadHeader() has to be run in advance.

[Parameter]

Member	Description
idx	resource index to loading (PP_CAR_RSCLIST_E)

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Fail

5.5 PPAPI_DISPLAY_LoadPglImage

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_LoadPglImage (PP_U32 IN idx)
```

[Description]

Pgl image will be loaded from flash and parsed according to the structure.

To run this API, PPAPI_DISPLAY_LoadHeader() has to be run in advance.

[Parameter]

Member	Description
idx	resource index to loading (PP_PGL_RSCLIST_E)

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Fail

5.6 PPAPI_DISPLAY_PrintLoadImage

[Syntax]

```
PP_VOID PPAPI_DISPLAY_PrintLoadImage (PP_VOID)
```

[Description]

This API is used to print result that UI / CAR / PGL resource image be loaded in flash and parsing structure. This is for debugging.

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
PP_VOID	

5.7 PPAPI_DISPLAY_LoadWheel

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_LoadWheel (PP_VOID);
```

[Description]

This API is used to load 16bit Car Wheel Image.

PPAPI_DISPLAY_LoadHeader() has to be executed in advance.

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	성공
eERROR_FAILURE	실패

5.8 PPAPI_DISPLAY_ParsingImage

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_ParsingImage (PP_SCENE_E IN scene)
```

[Description]

This API is used to parse image and LUT for target scene when scene is changed.

[Parameter]

Member	Description
scene	scene

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Fail

5.9 PPAPI_DISPLAY_GeUiImg

[Syntax]

```
STATIC PP_RSC_UI_IMG_S *PPAPI_DISPLAY_GetUiImg (PP_U32 IN id)
```

[Description]

This API returns display info for target id of UI Image list which is used for target scene.

[Parameter]

Member	Description
id	image id

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Fail

5.10 PPAPI_DISPlAY_UpdateLUT

[Syntax]

```
PP_RESULT_E PPAPI_DISPlAY_UpdateLUT (PP_VOID);
```

[Description]

This API is used to update and set color LUT for target scene.

This API is called after PPAPI_DISPLAY_LoadImag() when scene is changed.

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Fail

5.11 PPAPI_DISPLAY_DisableAll

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_DisableAll (PP_VOID);
```

[Description]

Disable all display out except Pop-Up, Progress Bar and Background

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.12 PPAPI_DISPLAY_Background_On

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_Background_On (PP_VOID);
```

[Description]

This API is used to display background image.

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.13 PPAPI_DISPLAY_Background_Off

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_Background_Off (PP_VOID);
```

[Description]

This API is used to hide background image.

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	Success

eERROR_FAILURE	Failure

5.14 PPAPI_DISPLAY_BOOTING_CI

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_BOOTING_CI (PP_VOID);
```

[Description]

This API is used to display CI during booting

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.15 PPAPI_DISPLAY_VIEW_Outline_On

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_VIEW_Outline_On (PP_VOID);
```

[Description]

This API is used for Live View Scene.

This API draws borders between views for Live View Mode.

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Fail

5.16 PPAPI_DISPLAY_VIEW_Outline_Off

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_VIEW_Outline_Off (PP_VOID);
```

[Description]

This API is used for Live View Scene.

This API is used to remove borders between views for Live View Mode.

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Fail

5.17 PPAPI_DISPLAY_VIEW_UpdateWheel

[Syntax]

```
PP_VOID PPAPI_DISPLAY_VIEW_UpdateWheel (PP_BOOL isRun);
```

[Description]

This API is used to update Car Wheel

This API can make wheel rotation effect by updating car image whenever it is called.

[Parameter]

Member	Description
isRun	Whether to update wheel

[Return]

Member	Description
PP_VOID	

5.18 PPAPI_DISPLAY_MenuList

[Syntax]

```
RESULT PPAPI_DISPLAY_MenuList (PP_SCENE_E IN scene);
```

[Description]

This API is used to display all items for a scene.

[Parameter]

Member	Description
scene	Selected menu scene

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.19 PPAPI_DISPLAY_MenuItem

[Syntax]

```
RESULT PPAPI_DISPLAY_MenuItem (PP_SCENE_E IN scene, PP_U32 IN idx,
PP_BOOL IN isSel);
```

[Description]

This API is used to display selected item of a scene with or without highlight.

[Parameter]

Member	Description
scene	setting menu of current scene
idx	Index of selected item of current menu
isSel	Whether to highlight the selected item

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.20 PPAPI_DISPLAY_SubMenuList

[Syntax]

```
RESULT PPAPI_DISPLAY_SubMenuList (PP_SCENE_E IN scene, PP_SCENE_SUB_ELEM_S* IN elem, PP_U32 idx);
```

[Description]

This API is used to display current setting value of all items of a scene. This API also highlight setting value of current selected item.

[Parameter]

Member	Description
scene	Current menu
elem	Holds setting value of an item of a scene
idx	Index of selected item of a scene

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.21 PPAPI_DISPLAY_CALIB_Point

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_CALIB_Point (PP_POS_S* IN pnt, PP_U32 IN num);
```

[Description]

This API is used to display feature point of Manual Calibration. Feature points are displayed in gray color. Maximum number of feature point to be displayed is 8.

.

[Parameter]

Member	Description
pnt	Array of coordinate of feature point
num	Feature point number. Max value of num is 8.

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.22 PPAPI_DISPLAY_CALIB_SelPoint

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_CALIB_SelPoint (PP_POS_S* IN pnt, PP_U32 IN
num, PP_U32 IN idx);
```

[Description]

This API is used to display feature points of manual calibration.

Maximum number of feature point is 8.

Feature point with focus will be highlighted in yellow color

[Parameter]

Member	Description
pnt	Feature Point
num	Feature point number to be displayed
idx	Index of feature point to be highlighted

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.23 PPAPI_DISPLAY_CALIB_MovePoint

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_CALIB_MovePoint (PP_POS_S* IN pnt, PP_U32 IN
num, PP_U32 IN idx);
```

[Description]

This API is used to display feature points of manual calibration.

Maximum number of feature point is 8.

Feature point to be moved will be highlighted in red color.

[Parameter]

Member	Description
pnt	Feature Point
num	Feature point number to be displayed
idx	Index of feature point to be highlighted

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.24 PPAPI_DISPLAY_POPUP_On

[Syntax]

```
RESULT PPAPI_DISPLAY_POPUP_On (PP_POPUP_E IN msg);
```

[Description]

Display popup message

[Parameter]

Member	Description
msg	Pop up message type

5.25 PPAPI_DISPLAY_POPUP_Off

[Syntax]

```
RESULT PPAPI_DISPLAY_POPUP_Off (PP_VOID);
```

[Description]

Do not display popup message.

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.26 PPAPI_DISPLAY_PROGRESSBAR_On

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_PROGRESSBAR_On (PP_U32 IN level, PP_U32 IN flicker);
```

[Description]

Display progress bar.

[Parameter]

Member	Description
level	Progress level (range : 0 ~ 10)
flicker	There are two kinds of progress bar. To make animation effect, these two progress bar is displayed by turns. “flicker” is used to select which progress bar will be displayed. During displaying progressive bar, this API will be called with different value periodically. (Range : 0 ~ 1)

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.27 PPAPI_DISPLAY_PROGRESSBAR_Off

[Syntax]

```
RESULT PPAPI_DISPLAY_PROGRESSBAR_Off (PP_VOID);
```

[Description]

Do not display progress bar.

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.28 PPAPI_DISPLAY_DIALOG_On

[Syntax]

PP_RESULT_E PPAPI_DISPLAY_DIALOG_On (PP_DIALOG_BOX_E IN box, PP_DIALOG_BTN_E IN btn);

[Description]

Display dialog box.

[Parameter]

Member	Description
box	Type of dialog box
btn	Button to be highlighted

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.29 PPAPI_DISPLAY_DIALOG_Off

[Syntax]

RESULT PPAPI_DISPLAY_DIALOG_Off (PP_VOID);

[Description]

Dialog box will be disappeared.

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Fail

5.30 PPAPI_DISPLAY_VIEW_Car_On

[Syntax]

PP_RESULT_E PPAPI_DISPLAY_VIEW_Car_On (PP_U32 IN sectionID)	PP_RESULT_E P

[Description]

This API is used to display 2D Car/3D Car image for target Section ID.

[Parameter]

Member	Description
sectionID	section ID

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.31 PPAPI_DISPLAY_VIEW_Car_Off

[Syntax]

PP_VOID PPAPI_DISPLAY_VIEW_Car_Off(PP_VOID)

[Description]

This API is used to not display All Car image

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
PP_VOID	

5.32 PPAPI_DISPLAY_VIEW_SetCarDoor

[Syntax]

```
PP_VOID PPAPI_DISPLAY_VIEW_SetCarDoor (PP_CAR_DOOR_E IN open)
```

[Description]

This API is used to display Car door image

[Parameter]

Member	Description
open	open door image index to display

[Return]

Member	Description
PP_VOID	

5.33 PPAPI_DISPLAY_VIEW_PGL_On

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_VIEW_PGL_On (PP_U32 IN sectionID, PP_BOOL isBw)
```

[Description]

This API is used to display Parking Guide Line.

[Parameter]

Member	Description
sectionID	section ID
isBW	PGL direction (backward / forward)

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Fail

5.34 PPAPI_DISPLAY_VIEW_PGL_SetAngle

[Syntax]

```
PP_VOID PPAPI_DISPLAY_VIEW_PGL_SetAngle (PP_S16 IN angle);
```

[Description]

This API is used to update angle of Parking Guide Line.

[Parameter]

Member	Description
angle	Parking Guide Line 의 angle (range: -35 ~ 35)

[Return]

Member	Description
PP_VOID	

5.35 PPAPI_DISPLAY_VIEW_PGL_Off

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_VIEW_PGL_Off (PP_VOID);
```

[Description]

This API is used to remove Parking Guide Line.

[Parameter]

Member	Description
PP_VOID	

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Fail

5.36 PPAPI_DISPLAY_LoadCacheAddr

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_LoadCacheAddr (PP_U32 IN carDeg, PP_U32* OUT
pu32FlashAddr, PP_U32* OUT pu32DramAddr, PP_U32* OUT u32Size);
```

[Description]

Get information to load car image for 360degree swing

This API can be used only once just after booting.

[Parameter]

Member	Description
carDeg	Angle of car image
pu32FlashAddr	Flash address of car image for target angle
pu32DramAddr	Target dram address
u32Size	image size

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

5.37 PPAPI_DISPLAY_LoadCacheAddr_Shadow

[Syntax]

```
PP_RESULT_E PPAPI_DISPLAY_LoadCacheAddr_Shadow (PP_U32 IN shadowDeg,
PP_U32* OUT pu32FlashAddr, PP_U32* OUT pu32DramAddr, PP_U32* OUT u32Size);
```

[Description]

This API is used to get information to load shadow image for 360 degree swing from flash memory. This API can be used only for 360 degree swing just after booting.

[Parameter]

Member	Description
shadowDeg	Angle of shadow image

pu32FlashAddr	Flash address of car image for input angle
pu32DramAddr	Dram address to be loaded
u32Size	image size

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Fail

5.38 PPAPI_DISPLAY_SetSwingCar

[Syntax]

```
PP_VOID PPAPI_DISPLAY_SetSwingCar (PP_U32 IN carDeg);
```

[Description]

Update car image for target angle

[Parameter]

Member	Description
carDeg	Car angle

[Return]

Member	Description
eSUCCESS	Success
eERROR_FAILURE	Failure

6 Revision History

Version	Date	Description
v0.1	2017.12.07	Draft
v0.2	2018.02.27	Update
v0.3	2018.04.04	Update
v0.31	2018.06.04	Renewal & Update
v 0.4	2018.07.31	Update
v 0.41	2018.08.17	Update
v 0.5	2018.11.16	Update