



GPL951XX Picture Process Unit Driver User Manual

V0.1 – Feb. 31, 2017

Preliminary



Important Notice

GENERALPLUS TECHNOLOGY INC. reserves the right to change this documentation without prior notice. Information provided by GENERALPLUS TECHNOLOGY INC. is believed to be accurate and reliable. However, GENERALPLUS TECHNOLOGY INC. makes no warranty for any errors which may appear in this document. Contact GENERALPLUS TECHNOLOGY INC. to obtain the latest version of device specifications before placing your order. No responsibility is assumed by GENERALPLUS TECHNOLOGY INC. for any infringement of patent or other rights of third parties which may result from its use. In addition, GENERALPLUS products are not authorized for use as critical components in life support systems or aviation systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of Generalplus.

Table of Content

	PAGE
1 INTRODUCTION	8
1.1 GENERAL DESCRIPTION	8
1.2 FILE STRUCTURE	8
2 STRUCTURE, UNION AND CONSTANT DEFINE	9
2.1 STRUCTURE AND UNION DEFINE	9
2.1.1 GEN_REG_DEF.....	9
2.1.2 P_Tx_Attribute_DEF.....	10
2.1.3 P_Tx_Control_DEF.....	11
2.1.4 P_Sp_Control_DEF.....	13
2.1.5 P_Palette_Ctrl_DEF.....	14
2.1.6 P_IRQ_Control_DEF.....	15
2.1.7 P_IRQ_Status_DEF.....	16
2.1.8 P_Sp_Attribute0_DEF.....	17
2.1.9 P_Sp_Attribute1_DEF.....	19
2.1.10 P_PPU_Enable_DEF.....	19
2.1.11 SPRITE.....	21
2.2 CONSTANT DEFINE.....	22
2.2.1 In PPUSystem.h.....	22
2.2.2 In PPUDrv.h.....	24
3 VARIABLES.....	35
3.1 VARIABLES TABLE	35
4 API FUNCTION.....	39
4.1 MACRO DEFINE	39
4.1.1 Macro: SetCpuIrqON().....	39
4.1.2 Macro: SetCpuIrqOFF().....	39
4.2 FUNCTION DEFINE	39
4.2.1 Function: GetLongAddress().....	39
4.2.2 Function: MyMemSet().....	40
4.2.3 Function: SetDMACopy0().....	40
4.2.4 Function: SetColorPalette().....	40

4.2.5	Function: WriteColorPalette().....	41
4.2.6	Function: SetOneTransparentColor().....	41
4.2.7	Function: SetTransparentColor()	41
4.2.8	Function: SetAllTransparent().....	42
4.2.9	Function: ClearOneSprite().....	42
4.2.10	Function: ClearAllSprite().....	42
4.2.11	Function: ClearTextRegister().....	43
4.2.12	Function: ClearAllRam().....	43
4.2.13	Function: ClearSimple().....	43
4.2.14	Function: ClearAllSetting().....	43
4.2.15	Function: VblankIrqON()	44
4.2.16	Function: VblankIrqOFF().....	44
4.2.17	Function: SpDMAIrqON().....	44
4.2.18	Function: SpDMAIrqOFF().....	44
4.2.19	Function: PaletteErrIrqON().....	44
4.2.20	Function: PaletteErrIrqOFF().....	45
4.2.21	Function: TxUnderRunIrqON()	45
4.2.22	Function: TxUnderRunIrqOFF().....	45
4.2.23	Function: SpUnderRunIrqON();.....	45
4.2.24	Function: SpUnderRunIrqOFF().....	46
4.2.25	Function: ClearVblankIRQ().....	46
4.2.26	Function: ClearSpDMAIRQ()	46
4.2.27	Function: ClearPaletteErrIRQ()	46
4.2.28	Function: ClearTxUnderRunIRQ().....	47
4.2.29	Function: ClearSpUnderRunIRQ().....	47
4.2.30	Function: InitPPU()	47
4.2.31	Function: ppu_on ().....	48
4.2.32	Function: ppu_off ().....	48
4.2.33	Function: WaitBlanking()	48
4.2.34	Function: SetTextPosition()	48
4.2.35	Function: Screen_Fade_Out().....	49
4.2.36	Function: Screen_Fade_In().....	49
4.2.37	Function: Sprite_Hit_Chk().....	49
4.2.38	Function: HorizontalMovement().....	50
4.2.39	Function: PPU_Service()	50
4.2.40	Function: SetTextEffect()	50
4.2.41	Function: SetTxBlendMode().....	51

4.2.42	Function: SetBlendLevel4()	51
4.2.43	Function: SetTxBlendLevel64()	51
4.2.44	Function: HCompressEffect()	51
4.2.45	Function: VCompressEffect()	52
4.2.46	Function: HOffsetEffect()	52
4.2.47	Function: Sea_Hoffset()	52
4.2.48	Function: FlipEffect()	53
4.2.49	Function: InitCharacter()	53
4.2.50	Function: fnShow_Chr()	54
4.2.51	Function: InitBitmap()	54
4.2.52	Function: FnShow_Bmp()	55
4.2.53	Function: ScrollingEffect()	56
4.2.54	Function: SetSpriteCtrl()	56
4.2.55	Function: SetSpriteEffect()	57
4.2.56	Function: SetSpBlendMode()	57
4.2.57	Function: SetPaletteType()	57
4.2.58	Function: SetPaletteRamBank()	58
4.2.59	Function: SetTextMode()	58
4.2.60	Function: SetSpritePosition()	58
4.2.61	Function: SpCoordConvert()	59
4.2.62	Function: Init_Sprite()	59
4.2.63	Function: Paint_Sprite()	59
4.2.64	Function: SetSpRotate()	59
4.2.65	Function: SetSpZoom()	60
4.2.66	Function: SetSpDepth()	60
4.2.67	Function: SetSpBlend()	60
4.2.68	Function: SetSpMosaic()	61
4.2.69	Function: SetSpBlend64()	61
4.2.70	Function: SetSpriteFlip()	61
4.2.71	Function: SetSpPaletteBank()	61
4.2.72	Function: ClearICache()	62
4.2.73	Function: SetICache()	62
4.2.74	Function: VBLK_Service()	62
4.2.75	Function: InitBitmapHVGA()	62
4.2.76	Function: fnShow_Bmp_HVGA()	63
5	SYSTEM MEMORY REQUIREMENT	65



5.1	MEMORY REQUIREMENT	65
6	APPLICATION NOTE	66
6.1	V-BLANKING ISR	66

Revision History

Revision	Date	By	Remark
			1.
0.1	2/31/2017	Kingchao	First Edition.

1 Introduction

1.1 General Description

This guide describes how to use the PPU driver API for GPL951XX serials provided by Generalplus. With the aid of PPU driver API, users could realize quickly and easily various kinds of graphics operation for LCD game application.

1.2 File Structure

The GPL951XX PPU driver mainly consists of four files, PPUDrv.c, PPUDrv2.asm, PPUDrv.h, and PPUSystem.h. The PPUDrv.c and PPUDrv2.asm contain all source codes of PPU API functions, constant tables, and global variables used in this driver. The PPUSystem.h defines several unions and constant definition of PPU registers. The PPUSystem.h includes data type definitions, structure, constant definitions, and declarations of external global variables and API functions. Users could add these files provided by Generalplus to the program to control PPU easily.

2 Structure, Union and Constant Define

2.1 Structure and Union Define

2.1.1 GEN_REG_DEF

Declaration:

```
typedef union
{
    UInt16 W;
    struct
    {
        UInt16    bit0    : 1;
        UInt16    bit1    : 1;
        UInt16    bit2    : 1;
        UInt16    bit3    : 1;
        UInt16    bit4    : 1;
        UInt16    bit5    : 1;
        UInt16    bit6    : 1;
        UInt16    bit7    : 1;
        UInt16    bit8    : 1;
        UInt16    bit9    : 1;
        UInt16    bit10   : 1;
        UInt16    bit11   : 1;
        UInt16    bit12   : 1;
        UInt16    bit13   : 1;
        UInt16    bit14   : 1;
        UInt16    bit15   : 1;
    } B;
} GEN_REG_DEF;
```

Header File: PPUSystem.h

Remark: “W” means the 16bit word. “B” means each bit. You can use B.bitX to read/write the bit value or use W to read/write the 16 bit value.

2.1.2 P_Tx_Attribute_DEF

Declaration:

```
typedef union
{
    UInt16 W;
    struct
    {
        UInt16 Color    :2;
        UInt16 Flip     :2;
        UInt16 Hs       :2;
        UInt16 Vs       :2;
        UInt16 Palette  :4;
        UInt16 Depth    :2;
        UInt16 Size     :2;
    } B;
} P_Tx_Attribute_DEF;
```

Header File: PPUSystem.h

Remark: Text Attribute Setting (P_Tx_Attribute)

<i>bit 1 - 0</i>	<i>: Color</i>	<i>Color Mode</i>
	= 00	Color 4 Mode
	= 01	Color 16 Mode
	= 10	Color 64 Mode
	= 11	Color 256 Mode
<i>bit 3 - 2</i>	<i>: Flip</i>	<i>Character Flip</i>
	= 00	No Flip
	= 01	Horizontal Flip
	= 10	Vertical Flip
	= 11	Horizontal and Vertical Flip
<i>bit 5 - 4</i>	<i>: Hs</i>	<i>Horizontal size</i>
	= 00	8 pixels
	= 01	16 pixels
	= 10	32 pixels
	= 11	64 pixels
<i>bit 7 - 6</i>	<i>: Vs</i>	<i>Vertical size</i>
	= 00	8 pixels
	= 01	16 pixels

	= 10	32 pixels
	= 11	64 pixels
<i>bit 11 - 8</i>	<i>: Palette</i>	<i>Palette bank</i>
	= 0000	Palette bank 0
	= 0001	Palette bank 1
	= 0010	Palette bank 2
	= 0011	Palette bank 3
	= 0100	Palette bank 4
	= 0101	Palette bank 5
	= 0110	Palette bank 6
	= 0111	Palette bank 7
	= 1000	Palette bank 8
	= 1001	Palette bank 9
	= 1010	Palette bank 10
	= 1011	Palette bank 11
	= 1100	Palette bank 12
	= 1101	Palette bank 13
	= 1110	Palette bank 14
	= 1111	Palette bank 15
<i>bit 13 - 12</i>	<i>: Depth</i>	<i>Define Text depth layer</i>
	= 00	Depth 0
	= 01	Depth 1
	= 10	Depth 2
	= 11	Depth 3
<i>bit 15 - 14</i>	<i>: Size</i>	<i>Define Text size</i>
	= 00	512x256
	= 01	512x512
	= 10	1024x512
	= 11	1024x1024

2.1.3 P_Tx_Control_DEF

Declaration:

```
typedef union
{
    UInt16 W;
    struct
```

```

{
    UInt16 Linr      :1;
    UInt16 Rgm       :1;
    UInt16 Wap       :1;
    UInt16 Txe       :1;
    UInt16 Mve       :1;
    UInt16 Mode      :2;
    UInt16 RGB       :1;
    UInt16 Bld       :1;
    UInt16 Bldm      :1;
    UInt16 BldLev    :6;
} B;
} P_Tx_Control_DEF;

```

Header File: PPUSystem.h

Remark: Text Control (P_Tx_Control)

<i>bit 0</i>	<i>: BMPM</i>	<i>Bitmap Mode Enable</i>
	= 0	Character Mode
	= 1	Bitmap Mode
<i>bit 1</i>	<i>: REGM</i>	<i>Register Mode Enable</i>
	= 0	Attribute Array Effective
	= 1	Register Set Effective
<i>bit 2</i>	<i>: WALL</i>	<i>Wallpaper Effect Enable</i>
	= 0	Disable (normal)
	= 1	Enable (Only First Character/Line attribute is effective)
<i>bit 3</i>	<i>: TXEN</i>	<i>Text Enable</i>
	= 0	Disable (Non Visible)
	= 1	Enable (Visible)
<i>bit 4</i>	<i>: MVEN</i>	<i>Horizontal Movement Control Enable</i>
	= 0	Disable
	= 1	Enable
<i>bit 6 - 5</i>	<i>: MODE</i>	<i>Horizontal extension/compression enable(Just only Text1)</i>
	= 0	Disable
	= 1	Enable
<i>bit 7</i>	<i>: RGBM</i>	<i>32768 High Color Mode Enable</i>
	= 0	Normal Mode(4/16/64/256 colors mode)

	= 1	High Colors Mode (32768/65536)
<i>bit 8</i>	: <i>BLD</i>	<i>Blend Effect Control</i>
	= 0	Normal Mode
	= 1	Blend Mode
<i>bit 9</i>	: <i>BLDM</i>	<i>Blend Level Control</i>
	= 0	4-Level
	= 1	64-Level
<i>bit 15 - 10</i>	: <i>BLDLEV</i>	<i>Blend Level Value (64-Level)</i>

2.1.4 P_Sp_Control_DEF

Declaration:

```
typedef union
{
    UInt16 W;
    struct
    {
        UInt16 Spen      :1;
        UInt16 Coor      :1;
        UInt16 Bldm      :1;
        UInt16 Round     :1;
        UInt16 Dir        :1;
        UInt16 Mosen     :1;
        UInt16 Roten     :1;
        UInt16 Zoomen    :1;
        UInt16 Spnum     :8;
    } B;
} P_Sp_Control_DEF;
```

Header File: PPUSystem.h

Remark: Sprite Control Register (P_Sp_Control)

<i>bit 0</i>	: <i>Spen</i>	<i>Sprite Enable</i>
	= 0	Disable(Not Visible)
	= 1	Enable(Visible)
<i>bit 1</i>	: <i>Coor</i>	<i>Sprite Coordinates Mode</i>
	= 0	Origin coordinates(Center of the visible area)
	= 1	Left-upper corn of the visible area
<i>bit 2</i>	: <i>Bldm</i>	<i>Sprite Blending Mode</i>

	= 0	4-Level (Register)
	= 1	64-Level (Sprite RAM)
<i>bit 3</i>	<i>: Round</i>	<i>Round-robin Enable</i>
	= 0	Disable
	= 1	Enable
<i>bit 4</i>	<i>: Dir</i>	<i>Sprite Direct Address Mode</i>
	= 0	Relative Address Mode
	= 1	Direct Address Mode
<i>bit 5</i>	<i>: Mosen</i>	<i>Mosaic Mode Enable</i>
	= 0	Disable
	= 1	Enable
<i>bit 6</i>	<i>: Roten</i>	<i>Rotate Mode Enable</i>
	= 0	Disable
	= 1	Enable
<i>bit 7</i>	<i>: Zoomen</i>	<i>Zoom Mode Enable</i>
	= 0	Disable
	= 1	Enable
<i>bit 15 - 8</i>	<i>: Spnum</i>	<i>Sprite number</i>

2.1.5 P_Palette_Ctrl_DEF

Declaration:

```
typedef union
{
    UInt16 W;
    struct
    {
        UInt16 Type      :2;
        UInt16 Bank      :2;
        UInt16 Reserved  :12;
    } B;
} P_Palette_Ctrl_DEF;
```

Header File: PPUSystem.h

Remark: Palette Control Register (P_Palette_Ctrl)

<i>bit 1 - 0</i>	<i>: Type</i>	<i>Palette Type</i>
	= 0	16-bit Palette RAM 0 (Text/Sprite)
	= 1	16-bit Palette RAM 0 (Text), 16-bit Palette RAM 1

		(Sprite)
<i>bit 3 - 2</i>	: <i>Bank</i>	<i>Palette RAM Bank Select</i>
	= 0	Palette RAM Bank 0
	= 1	Palette RAM Bank 1
	= 2	Palette RAM Bank 2
	= 3	Palette RAM Bank 3
<i>bit 15 - 4</i>	: <i>Reserved</i>	

2.1.6 P_IRQ_Control_DEF

Declaration:

```
typedef union
{
    UInt16 W;
    struct
    {
        UInt16 BLK_IRQ_EN      :1;
        UInt16 DMA_IRQ_EN      :1;
        UInt16 PALERR_IRQ_EN   :1;
        UInt16 TXUR_IRQ_EN     :1;
        UInt16 SPUR_IRQ_EN     :1;
        UInt16 Reserved        :4;
    } B;
} P_IRQ_Control_DEF;
```

Header File: PPUSystem.h

Remark: PPU IRQ Control Resgister (P_PPU_IRQ_Ctrl)

<i>bit 0</i>	: <i>BLK_IRQ_EN</i>	<i>Vertical Blanking Period IRQ Enable Control</i>
	= 0	Disable
	= 1	Enable
<i>bit 1</i>	: <i>DMA_IRQ_EN</i>	<i>DMA IRQ Enable Control</i>
	= 0	Disable
	= 1	Enable
<i>bit 2</i>	: <i>PALERR_IRQ_EN</i>	<i>Palette Write Error IRQ Enable Control</i>
	= 0	Disable
	= 1	Enable
<i>bit 3</i>	: <i>TXUR_IRQ_EN</i>	<i>Text Engine Under-run IRQ Enable Control</i>
	= 0	Disable
	= 1	Enable

<i>bit 4</i>	<i>: SPUR_IRQ_EN</i>	<i>Sprite Engine Under-run IRQ Enable Control</i>
	<i>= 0</i>	<i>Disable</i>
	<i>= 1</i>	<i>Enable</i>
<i>bit 5 - 8</i>	<i>: Reserved</i>	

2.1.7 P_IRQ_Status_DEF

Declaration:

```
typedef union
{
    UInt16 W;
    struct
    {
        UInt16 BLK_IRQ      :1;
        UInt16 DMA_IRQ      :1;
        UInt16 PALERR_IRQ   :1;
        UInt16 TXUR_IRQ     :1;
        UInt16 SPUR_IRQ     :1;
        UInt16 Reserved     :4;
    } B;
} P_IRQ_Status_DEF;
```

Header File: PPUSystem.h

Remark: PPU IRQ Status Resgister (P_PPU_IRQ_Status)

<i>bit 0</i>	<i>: BLK_IRQ</i>	<i>Vertical Blanking IRQ Status</i>
	<i>: Write 0</i>	<i>No operation</i>
	<i>: Write 1</i>	<i>Vertical Blanking IRQ clear</i>
	<i>: Read 0</i>	<i>Vertical Blanking Period not occurred</i>
	<i>: Read 1</i>	<i>Vertical Blanking Period occurred</i>
<i>bit 1</i>	<i>: DMA_IRQ</i>	<i>DMA IRQ Status</i>
	<i>: Write 0</i>	<i>No operation</i>
	<i>: Write 1</i>	<i>DMA IRQ clear</i>
	<i>: Read 0</i>	<i>DMA IRQ not occurred</i>
	<i>: Read 1</i>	<i>DMA IRQ occurred</i>
<i>bit 2</i>	<i>: PALERR_IRQ</i>	<i>Palette Write Error IRQ Status</i>
	<i>: Write 0</i>	<i>No operation</i>
	<i>: Write 1</i>	<i>Palette Write Error IRQ clear</i>
	<i>: Read 0</i>	<i>Palette Write Error IRQ not occurred</i>
	<i>: Read 1</i>	<i>Palette Write Error IRQ occurred</i>

<i>bit 3</i>	<i>: TXUR_IRQ</i>	<i>Text Engine Under-run IRQ Status</i>
	<i>: Write 0</i>	No operation
	<i>: Write 1</i>	Text Engine Under-run IRQ clear
	<i>: Read 0</i>	Text Engine Under-run IRQ not occurred
	<i>: Read 1</i>	Text Engine Under-run IRQ occurred
<i>bit 4</i>	<i>: SPUR_IRQ</i>	<i>Sprite Engine Under-run IRQ Status</i>
	<i>: Write 0</i>	No operation
	<i>: Write 1</i>	Sprite Engine Under-run IRQ clear
	<i>: Read 0</i>	Sprite Engine Under-run IRQ not occurred
	<i>: Read 1</i>	Sprite Engine Under-run IRQ occurred
<i>bit 5 - 8</i>	<i>: Reserved</i>	

2.1.8 P_Sp_Attribute0_DEF

Declaration:

```
typedef union
{
    UInt16 W;
    struct
    {
        UInt16    Sp_Color    :2;
        UInt16    Sp_Flip     :2;
        UInt16    Sp_HSize    :2;
        UInt16    Sp_VSize    :2;
        UInt16    Sp_Palette  :4;
        UInt16    Sp_Depth    :2;
        UInt16    Sp_Blend    :1;
        UInt16    Sp_PB       :1;
    } B;
} P_Sp_Attribute0_DEF;
```

Header File: PPUSystem.h

Remark: Sprite Attributes register0 (P_SpriteN_Attribute0)

<i>bit 1 - 0</i>	<i>: Sp_Color</i>	<i>Sprite Color Mode</i>
	= 00	2 bits/ pixel 4 color mode
	= 01	4 bits/ pixel 16 color mode
	= 10	6 bits/ pixel 64 color mode
	= 11	8 bits/ pixel 256 color mode
<i>bit 3 - 2</i>	<i>: Sp_Flip</i>	<i>Sprite Character Flip</i>

	= 00	No Flip
	= 01	Horizontal Flip
	= 10	Vertical Flip
	= 11	Horizontal and Vertical Flip
<i>bit 5 - 4</i>	<i>: Sp_HSize</i>	<i>Sprite Character Horizontal Size</i>
	= 00	8 pixels
	= 01	16 pixels
	= 10	32 pixels
	= 11	64 pixels
<i>bit 7 - 6</i>	<i>: Sp_VSize</i>	<i>Sprite Character Vertical Size</i>
	= 00	8 pixels
	= 01	16 pixels
	= 10	32 pixels
	= 11	64 pixels
<i>bit 11 - 8</i>	<i>: Sp_Palette</i>	<i>Palette bank select</i>
	= 0000	Palette bank 0
	= 0001	Palette bank 1
	= 0010	Palette bank 2
	= 0011	Palette bank 3
	= 0100	Palette bank 4
	= 0101	Palette bank 5
	= 0110	Palette bank 6
	= 0111	Palette bank 7
	= 1000	Palette bank 8
	= 1001	Palette bank 9
	= 1010	Palette bank 10
	= 1011	Palette bank 11
	= 1100	Palette bank 12
	= 1101	Palette bank 13
	= 1110	Palette bank 14
	= 1111	Palette bank 15
<i>bit 13 - 12</i>	<i>: Sp_Depth</i>	<i>Define Sprite priority</i>
	= 00	Depth 0
	= 01	Depth 1
	= 10	Depth 2
	= 11	Depth 3
<i>bit 14</i>	<i>: Sp_Blend</i>	<i>Sprite Blending Effect control</i>

	= 0	Disable
	= 1	Enable
<i>bit 15</i>	<i>: Sprite palette select</i>	
	= 0	Palette 0
	= 1	Palette 1

2.1.9 P_Sp_Attribute1_DEF

Declaration:

```
typedef union
{
    UInt16 W;
    struct
    {
        UInt16    Sp_ChNum    :8;
        UInt16    Sp_BldLev   :6;
        UInt16    Sp_Mosaic   :2;
    } B;
} P_Sp_Attribute1_DEF;
```

Header File: PPUSystem.h

Remark: Sprite Attributes register1 (P_SpriteN_Attribute1)

bit 7 - 0	: Sp_ChNum	Sprite Character Number[23:16]
bit 13 - 8	: Sp_BldLev	Sprite Blend Level (64-Level)
bit 15 - 14	: Sp_Mosaic	Sprite Mosaic Mode
	= 00	No Effect
	= 01	2 pixels
	= 10	4 pixels
	= 11	8 pixels

2.1.10 P_PPU_Enable_DEF

Declaration:

```
typedef union
{
    UInt16 W;
    struct
    {
        UInt16    PPUen       :1;
        UInt16    Reserved1   :1;
    }
}
```

```

        UInt16    TxDirect      :1;
        UInt16    TxBotUp      :1;
        UInt16    VGAen        :1;
        UInt16    VGANonIntl   :1;
        UInt16    Free         :1;
        UInt16    FBen         :1;
        UInt16    FBformat     :1;
        UInt16    Sp25d        :1;
        UInt16    FBmono       :2;
        UInt16    Reserved2    :4;

```

```

    } B;
    } P_PPU_Enable_DEF;

```

Header File: PPUSystem.h

Remark: PPU control register (P_PPU_Enable)

<i>bit 0</i>	<i>: PPUen</i>	<i>PPU Enable</i>
	= 0	Disable
	= 1	Enable
<i>bit 2</i>	<i>: TxDirect</i>	<i>Text direct mode</i>
	= 0	Related address mode
	= 1	Direct address mode
<i>bit 3</i>	<i>: TxBotUp</i>	<i>Text calculation mode</i>
	= 0	Top to bottom layer
	= 1	Bottom to top layer
<i>bit 4</i>	<i>: VGAen</i>	<i>VGA mode</i>
	= 0	QVGA (320x240)
	= others	NC
<i>bit 5</i>	<i>: VGANonIntl</i>	<i>VGA non-interlace mode</i>
	= 0	VGA interlace
	= 1	VGA non-interlace
<i>bit 6</i>	<i>: Free</i>	<i>Free address mode</i>
	= 0	22-bit address mode
	= 1	27-bit address mode
<i>bit 7</i>	<i>: FBen</i>	<i>Frame buffer mode</i>
	= 0	line-base mode
	= others	NC
<i>bit 8</i>	<i>: FBformat</i>	<i>Frame buffer data format</i>
	= 0	RGB565

	= others	NC
<i>bit 9</i>	: <i>Sp25d</i>	<i>Sprite layer mode</i>
	= 0	2D mode
	= others	NC
<i>bit 11 - 10</i>	: <i>FB_Mono</i>	<i>Frame buffer pixel resolution</i>
	= 0	RGB565
	= others	NC
<i>bit 15 - 12</i>	: <i>Reserved.</i>	

2.1.11 SPRITE

Declaration:

```
typedef struct
{
    int      ldx;
    int      x;
    int      y;
    int      xSize;
    int      ySize;
    const int *sprcell;
    int      att0;
    int      att1;
    int      pos1;
    int      pos2;
    int      pos3;
} SPRITE;
```

Header File: PPUDrv.h

Remark:

ldx;	ID number
x;	X Position
y;	Y Position
xSize;	Sprite x size
ySize;	Sprite y size
*sprcell;	Sprite cell data set
att0;	Sprite attribute 0
att1;	Sprite attribute 1
pos1;	Sprite position 1
pos2;	Sprite position 2

pos3; Sprite position 3

2.2 Constant Define

2.2.1 In PPUSystem.h

Name	Value/Declaration	Remark
P_Tx1_X_Position_ADDR	0x7010	Text 1 x position register address
P_Tx1_Y_Position_ADDR	0x7011	Text 1 y position register address
P_Tx2_X_Position_ADDR	0x7016	Text 2 x position register address
P_Tx2_Y_Position_ADDR	0x7017	Text 2 y position register address
P_Tx3_X_Position_ADDR	0x7000	Text 3 x position register address
P_Tx3_Y_Position_ADDR	0x7001	Text 3 y position register address
P_Tx_HvOffset_ADDR	0x7100	Text horizontal movement register start address
P_HComp_Value_ADDR	0x7200	Text horizontal compression register start address
P_Palette_ADDR	0x7300	Palette RAM start address
P_Sp_Num_ADDR	0x7400	Sprite Ram start address
P_Tx1_X_Position_B	((volatile GEN_REG_DEF *) (P_Tx1_X_Position_ADDR))	Text1 horizontal position
P_Tx1_Y_Position_B	((volatile GEN_REG_DEF *) (P_Tx1_Y_Position_ADDR))	Text1 vertical position
P_Tx1_Attribute_B	((volatile P_Tx_Attribute_DEF *) (0x7012))	Text1 attribute register
P_Tx1_Control_B	((volatile P_Tx_Control_DEF *) (0x7013))	Text1 control register
P_Tx1_N_Ptr_B	((volatile GEN_REG_DEF *) (0x7014))	Text1 Character Number array pointer
P_Tx1_A_Ptr_B	((volatile GEN_REG_DEF *) (0x7015))	Text1 Attribute array pointer
P_Tx2_X_Position_B	((volatile GEN_REG_DEF *) (P_Tx2_X_Position_ADDR))	Text2 horizontal position
P_Tx2_Y_Position_B	((volatile GEN_REG_DEF *) (P_Tx2_Y_Position_ADDR))	Text2 vertical position
P_Tx2_Attribute_B	((volatile P_Tx_Attribute_DEF *) (0x7018))	Text2 attribute register
P_Tx2_Control_B	((volatile P_Tx_Control_DEF *) (0x7019))	Text2 control register
P_Tx2_N_Ptr_B	((volatile GEN_REG_DEF *) (0x701a))	Text2 Character Number array pointer

Name	Value/Declaration	Remark
P_Tx2_A_Ptr_B	((volatile GEN_REG_DEF *) (0x701b))	Text2 Attribute array pointer
P_Tx3_X_Position_B	((volatile GEN_REG_DEF *) (P_Tx3_X_Position_ADDR))	Text3 horizontal position
P_Tx3_Y_Position_B	((volatile GEN_REG_DEF *) (P_Tx3_Y_Position_ADDR))	Text3 vertical position
P_Tx3_X_Offset_B	((volatile GEN_REG_DEF *) (0x7002))	Text3 horizontal offset
P_Tx3_Y_Offset_B	((volatile GEN_REG_DEF *) (0x7003))	Text3 vertical offset
P_Tx3_Attribute_B	((volatile P_Tx_Attribute_DEF *) (0x7004))	Text3 attribute register
P_Tx3_Control_B	((volatile P_Tx_Control_DEF *) (0x7005))	Text3 control register
P_Tx3_N_Ptr_B	((volatile GEN_REG_DEF *) (0x7006))	Text3 Character Number array pointer
P_Tx3_A_Ptr_B	((volatile GEN_REG_DEF *) (0x7007))	Text3 Attribute array pointer
P_VComp_Value_B	((volatile GEN_REG_DEF *) (0x701c))	Vertical compression value control register
P_VComp_Offset_B	((volatile GEN_REG_DEF *) (0x701d))	Vertical compression offset control register
P_VComp_Step_B	((volatile GEN_REG_DEF *) (0x701e))	Vertical compression step value control register
P_Segment_Tx1_B	((volatile GEN_REG_DEF *) (0x7020))	Text1 Segment register
P_Segment_Tx2_B	((volatile GEN_REG_DEF *) (0x7021))	Text2 Segment register
P_Segment_Sp_B	((volatile GEN_REG_DEF *) (0x7022))	Sprite Segment register
P_Segment_Tx3_B	((volatile GEN_REG_DEF *) (0x7023))	Text3 Segment register
P_Blending_B	((volatile GEN_REG_DEF *) (0x702a))	Text blending effect control register
P_Segment_Tx1H_B	((volatile GEN_REG_DEF *) (0x702b))	TEXT1 segment address high word register
P_Segment_Tx2H_B	((volatile GEN_REG_DEF *) (0x702c))	TEXT2 segment address high word register
P_Segment_SpH_B	((volatile GEN_REG_DEF *) (0x702d))	Sprite segment address high word register
P_Segment_Tx3H_B	((volatile GEN_REG_DEF *) (0x702e))	TEXT3 segment address high word register
P_Fade_Control_B	((volatile GEN_REG_DEF *) (0x7030))	Fade effect control register
P_Palette_Control_B	((volatile P_Palette_Ctrl_DEF *) (0x703a))	Palette control register
P_Sp_Control_B	((volatile P_Sp_Control_DEF *) (0x7042))	Sprite Control Register
P_PPU_IRQ_Control_B	((volatile P_IRQ_Control_DEF *) (0x7062))	IRQ Control Register
P_PPU_IRQ_Status_B	((volatile P_IRQ_Status_DEF *) (0x7063))	IRQ Status Register
P_SPDMA_Source_B	((volatile GEN_REG_DEF *) (0x7070))	Sprite DMA source start address
P_SPDMA_Target_B	((volatile GEN_REG_DEF *) (0x7071))	Sprite DMA target start

Name	Value/Declaration	Remark
		address
P_SPDMA_Number_B	((volatile GEN_REG_DEF *) (0x7072))	Sprite DMA transfer number
P_PPU_Enable_B	((volatile P_PPU_Enable_DEF *) (0x707f))	PPU control register
P_Tx_HvOffsetN_B	((volatile GEN_REG_DEF *) (P_Tx_HvOffset_ADDR))	Text Horizontal movement control registers
P_HComp_ValueN_B	((volatile GEN_REG_DEF *) (P_HComp_Value_ADDR))	Text Horizontal extension/compression scale register
P_PaletteN_B	((volatile GEN_REG_DEF *) (P_Palette_ADDR))	Color Palette
P_SpriteN_CharNum_B	((volatile GEN_REG_DEF *) (P_Sp_Num_ADDR))	Sprite Character Number
P_SpriteN_X_B	((volatile GEN_REG_DEF *) (0x7401))	Sprite Horizontal position
P_SpriteN_Y_B	((volatile GEN_REG_DEF *) (0x7402))	Sprite Vertical position
P_SpriteN_Attribute0_B	((volatile P_Sp_Attribute_DEF *) (0x7403))	Sprite Attributes register 0
P_SpriteN_Attribute1_B	((volatile P_Sp_Attribute_DEF *) (0x7400))	Sprite Attributes register 1

2.2.2 In PPUDrv.h

Name	Value	Remark
SP_MAX	256	Max Sprite Number
SP_MAX_BUF	SP_MAX*4	Max Sprite Buffer Size
SP_MAX_CHAR	64	Maximum Combination Character To Consist Of One 2D Sprite
MAX_INDEX_LENGTH	1024	Maximum Index Table Length
MAX_ATT_LENGTH	512	Maximum Attribute Table Length
TX1_INDEX_LENGTH	1024	Text1 Index Table Length, users can modify the length by application.
TX1_ATT_LENGTH	512	Text1 Attribute Table Length, users can modify the length by application.
TX2_INDEX_LENGTH	1024	Text2 Index Table Length, users can modify the length by application.
TX2_ATT_LENGTH	512	Text2 Attribute Table Length, users can modify the length by application.
TX3_INDEX_LENGTH	1024	Text3 Index Table Length, users can modify the length by application.
TX3_ATT_LENGTH	512	Text3 Attribute Table Length, users can modify the length by application.
CELL_LENGTH	17	One Cell data length for Sprite
ATT_LENGTH	9	Attribute0 data length for Sprite
ATT_LENGTH2	3	Attribute1 data length for Sprite

Name	Value	Remark
QVGA_SCREENWIDTH	320	Screen Width of QVGA
QVGA_SCREENHEIGHT	240	Screen Height of QVGA
SP_ROTATE0	(0 << 10)	Sprite Rotate Level 0
SP_ROTATE1	(1 << 10)	Sprite Rotate Level 1
SP_ROTATE2	(2 << 10)	Sprite Rotate Level 2
SP_ROTATE3	(3 << 10)	Sprite Rotate Level 3
SP_ROTATE4	(4 << 10)	Sprite Rotate Level 4
SP_ROTATE5	(5 << 10)	Sprite Rotate Level 5
SP_ROTATE6	(6 << 10)	Sprite Rotate Level 6
SP_ROTATE7	(7 << 10)	Sprite Rotate Level 7
SP_ROTATE8	(8 << 10)	Sprite Rotate Level 8
SP_ROTATE9	(9 << 10)	Sprite Rotate Level 9
SP_ROTATE10	(10 << 10)	Sprite Rotate Level 10
SP_ROTATE11	(11 << 10)	Sprite Rotate Level 11
SP_ROTATE12	(12 << 10)	Sprite Rotate Level 12
SP_ROTATE13	(13 << 10)	Sprite Rotate Level 13
SP_ROTATE14	(14 << 10)	Sprite Rotate Level 14
SP_ROTATE15	(15 << 10)	Sprite Rotate Level 15
SP_ROTATE16	(16 << 10)	Sprite Rotate Level 16
SP_ROTATE17	(17 << 10)	Sprite Rotate Level 17
SP_ROTATE18	(18 << 10)	Sprite Rotate Level 18
SP_ROTATE19	(19 << 10)	Sprite Rotate Level 19
SP_ROTATE20	(20 << 10)	Sprite Rotate Level 20
SP_ROTATE21	(21 << 10)	Sprite Rotate Level 21
SP_ROTATE22	(22 << 10)	Sprite Rotate Level 22
SP_ROTATE23	(23 << 10)	Sprite Rotate Level 23
SP_ROTATE24	(24 << 10)	Sprite Rotate Level 24
SP_ROTATE25	(25 << 10)	Sprite Rotate Level 25
SP_ROTATE26	(26 << 10)	Sprite Rotate Level 26
SP_ROTATE27	(27 << 10)	Sprite Rotate Level 27
SP_ROTATE28	(28 << 10)	Sprite Rotate Level 28
SP_ROTATE29	(29 << 10)	Sprite Rotate Level 29
SP_ROTATE30	(30 << 10)	Sprite Rotate Level 30
SP_ROTATE31	(31 << 10)	Sprite Rotate Level 31
SP_ROTATE32	(32 << 10)	Sprite Rotate Level 32
SP_ROTATE33	(33 << 10)	Sprite Rotate Level 33
SP_ROTATE34	(34 << 10)	Sprite Rotate Level 34
SP_ROTATE35	(35 << 10)	Sprite Rotate Level 35
SP_ROTATE36	(36 << 10)	Sprite Rotate Level 36
SP_ROTATE37	(37 << 10)	Sprite Rotate Level 37
SP_ROTATE38	(38 << 10)	Sprite Rotate Level 38

Name	Value	Remark
SP_ROTATE39	(39 << 10)	Sprite Rotate Level 39
SP_ROTATE40	(40 << 10)	Sprite Rotate Level 40
SP_ROTATE41	(41 << 10)	Sprite Rotate Level 41
SP_ROTATE42	(42 << 10)	Sprite Rotate Level 42
SP_ROTATE43	(43 << 10)	Sprite Rotate Level 43
SP_ROTATE44	(44 << 10)	Sprite Rotate Level 44
SP_ROTATE45	(45 << 10)	Sprite Rotate Level 45
SP_ROTATE46	(46 << 10)	Sprite Rotate Level 46
SP_ROTATE47	(47 << 10)	Sprite Rotate Level 47
SP_ROTATE48	(48 << 10)	Sprite Rotate Level 48
SP_ROTATE49	(49 << 10)	Sprite Rotate Level 49
SP_ROTATE50	(50 << 10)	Sprite Rotate Level 50
SP_ROTATE51	(51 << 10)	Sprite Rotate Level 51
SP_ROTATE52	(52 << 10)	Sprite Rotate Level 52
SP_ROTATE53	(53 << 10)	Sprite Rotate Level 53
SP_ROTATE54	(54 << 10)	Sprite Rotate Level 54
SP_ROTATE55	(55 << 10)	Sprite Rotate Level 55
SP_ROTATE56	(56 << 10)	Sprite Rotate Level 56
SP_ROTATE57	(57 << 10)	Sprite Rotate Level 57
SP_ROTATE58	(58 << 10)	Sprite Rotate Level 58
SP_ROTATE59	(59 << 10)	Sprite Rotate Level 59
SP_ROTATE60	(60 << 10)	Sprite Rotate Level 60
SP_ROTATE61	(61 << 10)	Sprite Rotate Level 61
SP_ROTATE62	(62 << 10)	Sprite Rotate Level 62
SP_ROTATE63	(63 << 10)	Sprite Rotate Level 63
SP_ZOOM0	(0 << 10)	Sprite Zoom Level 0
SP_ZOOM1	(1 << 10)	Sprite Zoom Level 1
SP_ZOOM2	(2 << 10)	Sprite Zoom Level 2
SP_ZOOM3	(3 << 10)	Sprite Zoom Level 3
SP_ZOOM4	(4 << 10)	Sprite Zoom Level 4
SP_ZOOM5	(5 << 10)	Sprite Zoom Level 5
SP_ZOOM6	(6 << 10)	Sprite Zoom Level 6
SP_ZOOM7	(7 << 10)	Sprite Zoom Level 7
SP_ZOOM8	(8 << 10)	Sprite Zoom Level 8
SP_ZOOM9	(9 << 10)	Sprite Zoom Level 9
SP_ZOOM10	(10 << 10)	Sprite Zoom Level 10
SP_ZOOM11	(11 << 10)	Sprite Zoom Level 11
SP_ZOOM12	(12 << 10)	Sprite Zoom Level 12
SP_ZOOM13	(13 << 10)	Sprite Zoom Level 13
SP_ZOOM14	(14 << 10)	Sprite Zoom Level 14
SP_ZOOM15	(15 << 10)	Sprite Zoom Level 15

Name	Value	Remark
SP_ZOOM16	(16 << 10)	Sprite Zoom Level 16
SP_ZOOM17	(17 << 10)	Sprite Zoom Level 17
SP_ZOOM18	(18 << 10)	Sprite Zoom Level 18
SP_ZOOM19	(19 << 10)	Sprite Zoom Level 19
SP_ZOOM20	(20 << 10)	Sprite Zoom Level 20
SP_ZOOM21	(21 << 10)	Sprite Zoom Level 21
SP_ZOOM22	(22 << 10)	Sprite Zoom Level 22
SP_ZOOM23	(23 << 10)	Sprite Zoom Level 23
SP_ZOOM24	(24 << 10)	Sprite Zoom Level 24
SP_ZOOM25	(25 << 10)	Sprite Zoom Level 25
SP_ZOOM26	(26 << 10)	Sprite Zoom Level 26
SP_ZOOM27	(27 << 10)	Sprite Zoom Level 27
SP_ZOOM28	(28 << 10)	Sprite Zoom Level 28
SP_ZOOM29	(29 << 10)	Sprite Zoom Level 29
SP_ZOOM30	(30 << 10)	Sprite Zoom Level 30
SP_ZOOM31	(31 << 10)	Sprite Zoom Level 31
SP_ZOOM32	(32 << 10)	Sprite Zoom Level 32
SP_ZOOM33	(33 << 10)	Sprite Zoom Level 33
SP_ZOOM34	(34 << 10)	Sprite Zoom Level 34
SP_ZOOM35	(35 << 10)	Sprite Zoom Level 35
SP_ZOOM36	(36 << 10)	Sprite Zoom Level 36
SP_ZOOM37	(37 << 10)	Sprite Zoom Level 37
SP_ZOOM38	(38 << 10)	Sprite Zoom Level 38
SP_ZOOM39	(39 << 10)	Sprite Zoom Level 39
SP_ZOOM40	(40 << 10)	Sprite Zoom Level 40
SP_ZOOM41	(41 << 10)	Sprite Zoom Level 41
SP_ZOOM42	(42 << 10)	Sprite Zoom Level 42
SP_ZOOM43	(43 << 10)	Sprite Zoom Level 43
SP_ZOOM44	(44 << 10)	Sprite Zoom Level 44
SP_ZOOM45	(45 << 10)	Sprite Zoom Level 45
SP_ZOOM46	(46 << 10)	Sprite Zoom Level 46
SP_ZOOM47	(47 << 10)	Sprite Zoom Level 47
SP_ZOOM48	(48 << 10)	Sprite Zoom Level 48
SP_ZOOM49	(49 << 10)	Sprite Zoom Level 49
SP_ZOOM50	(50 << 10)	Sprite Zoom Level 50
SP_ZOOM51	(51 << 10)	Sprite Zoom Level 51
SP_ZOOM52	(52 << 10)	Sprite Zoom Level 52
SP_ZOOM53	(53 << 10)	Sprite Zoom Level 53
SP_ZOOM54	(54 << 10)	Sprite Zoom Level 54
SP_ZOOM55	(55 << 10)	Sprite Zoom Level 55
SP_ZOOM56	(56 << 10)	Sprite Zoom Level 56

Name	Value	Remark
SP_ZOOM57	(57 << 10)	Sprite Zoom Level 57
SP_ZOOM58	(58 << 10)	Sprite Zoom Level 58
SP_ZOOM59	(59 << 10)	Sprite Zoom Level 59
SP_ZOOM60	(60 << 10)	Sprite Zoom Level 60
SP_ZOOM61	(61 << 10)	Sprite Zoom Level 61
SP_ZOOM62	(62 << 10)	Sprite Zoom Level 62
SP_ZOOM63	(63 << 10)	Sprite Zoom Level 63
SPBLEND_LEVEL0	(0 << 8)	Sprite Blend Level 0
SPBLEND_LEVEL1	(1 << 8)	Sprite Blend Level 1
SPBLEND_LEVEL2	(2 << 8)	Sprite Blend Level 2
SPBLEND_LEVEL3	(3 << 8)	Sprite Blend Level 3
SPBLEND_LEVEL4	(4 << 8)	Sprite Blend Level 4
SPBLEND_LEVEL5	(5 << 8)	Sprite Blend Level 5
SPBLEND_LEVEL6	(6 << 8)	Sprite Blend Level 6
SPBLEND_LEVEL7	(7 << 8)	Sprite Blend Level 7
SPBLEND_LEVEL8	(8 << 8)	Sprite Blend Level 8
SPBLEND_LEVEL9	(9 << 8)	Sprite Blend Level 9
SPBLEND_LEVEL10	(10 << 8)	Sprite Blend Level 10
SPBLEND_LEVEL11	(11 << 8)	Sprite Blend Level 11
SPBLEND_LEVEL12	(12 << 8)	Sprite Blend Level 12
SPBLEND_LEVEL13	(13 << 8)	Sprite Blend Level 13
SPBLEND_LEVEL14	(14 << 8)	Sprite Blend Level 14
SPBLEND_LEVEL15	(15 << 8)	Sprite Blend Level 15
SPBLEND_LEVEL16	(16 << 8)	Sprite Blend Level 16
SPBLEND_LEVEL17	(17 << 8)	Sprite Blend Level 17
SPBLEND_LEVEL18	(18 << 8)	Sprite Blend Level 18
SPBLEND_LEVEL19	(19 << 8)	Sprite Blend Level 19
SPBLEND_LEVEL20	(20 << 8)	Sprite Blend Level 20
SPBLEND_LEVEL21	(21 << 8)	Sprite Blend Level 21
SPBLEND_LEVEL22	(22 << 8)	Sprite Blend Level 22
SPBLEND_LEVEL23	(23 << 8)	Sprite Blend Level 23
SPBLEND_LEVEL24	(24 << 8)	Sprite Blend Level 24
SPBLEND_LEVEL25	(25 << 8)	Sprite Blend Level 25
SPBLEND_LEVEL26	(26 << 8)	Sprite Blend Level 26
SPBLEND_LEVEL27	(27 << 8)	Sprite Blend Level 27
SPBLEND_LEVEL28	(28 << 8)	Sprite Blend Level 28
SPBLEND_LEVEL29	(29 << 8)	Sprite Blend Level 29
SPBLEND_LEVEL30	(30 << 8)	Sprite Blend Level 30
SPBLEND_LEVEL31	(31 << 8)	Sprite Blend Level 31
SPBLEND_LEVEL32	(32 << 8)	Sprite Blend Level 32
SPBLEND_LEVEL33	(33 << 8)	Sprite Blend Level 33

Name	Value	Remark
SPBLEND_LEVEL34	(34 << 8)	Sprite Blend Level 34
SPBLEND_LEVEL35	(35 << 8)	Sprite Blend Level 35
SPBLEND_LEVEL36	(36 << 8)	Sprite Blend Level 36
SPBLEND_LEVEL37	(37 << 8)	Sprite Blend Level 37
SPBLEND_LEVEL38	(38 << 8)	Sprite Blend Level 38
SPBLEND_LEVEL39	(39 << 8)	Sprite Blend Level 39
SPBLEND_LEVEL40	(40 << 8)	Sprite Blend Level 40
SPBLEND_LEVEL41	(41 << 8)	Sprite Blend Level 41
SPBLEND_LEVEL42	(42 << 8)	Sprite Blend Level 42
SPBLEND_LEVEL43	(43 << 8)	Sprite Blend Level 43
SPBLEND_LEVEL44	(44 << 8)	Sprite Blend Level 44
SPBLEND_LEVEL45	(45 << 8)	Sprite Blend Level 45
SPBLEND_LEVEL46	(46 << 8)	Sprite Blend Level 46
SPBLEND_LEVEL47	(47 << 8)	Sprite Blend Level 47
SPBLEND_LEVEL48	(48 << 8)	Sprite Blend Level 48
SPBLEND_LEVEL49	(49 << 8)	Sprite Blend Level 49
SPBLEND_LEVEL50	(50 << 8)	Sprite Blend Level 50
SPBLEND_LEVEL51	(51 << 8)	Sprite Blend Level 51
SPBLEND_LEVEL52	(52 << 8)	Sprite Blend Level 52
SPBLEND_LEVEL53	(53 << 8)	Sprite Blend Level 53
SPBLEND_LEVEL54	(54 << 8)	Sprite Blend Level 54
SPBLEND_LEVEL55	(55 << 8)	Sprite Blend Level 55
SPBLEND_LEVEL56	(56 << 8)	Sprite Blend Level 56
SPBLEND_LEVEL57	(57 << 8)	Sprite Blend Level 57
SPBLEND_LEVEL58	(58 << 8)	Sprite Blend Level 58
SPBLEND_LEVEL59	(59 << 8)	Sprite Blend Level 59
SPBLEND_LEVEL60	(60 << 8)	Sprite Blend Level 60
SPBLEND_LEVEL61	(61 << 8)	Sprite Blend Level 61
SPBLEND_LEVEL62	(62 << 8)	Sprite Blend Level 62
SPBLEND_LEVEL63	(63 << 8)	Sprite Blend Level 63
SP_MOSAIC0	(0 << 14)	Sprite Mosaic Level 0
SP_MOSAIC1	(1 << 14)	Sprite Mosaic Level 1
SP_MOSAIC2	(2 << 14)	Sprite Mosaic Level 2
SP_MOSAIC3	(3 << 14)	Sprite Mosaic Level 3
SP_PBANK0	(0 << 15)	Sprite 16-bit Palette Bank0
SP_PBANK1	(1 << 15)	Sprite 16-bit Palette Bank1
SPBLEND_DISABLE	(0 << 14)	Sprite Blending Effect Disable
SPBLEND_ENABLE	(1 << 14)	Sprite Blending Effect Enable
SP_DEPTH1	(0 << 12)	Sprite Depth 0
SP_DEPTH3	(1 << 12)	Sprite Depth 1
SP_DEPTH5	(2 << 12)	Sprite Depth 2

Name	Value	Remark
SP_DEPTH7	(3 << 12)	Sprite Depth 3
SP_PALETTE0	(0 << 8)	Sprite Palette Bank0
SP_PALETTE1	(1 << 8)	Sprite Palette Bank1
SP_PALETTE2	(2 << 8)	Sprite Palette Bank2
SP_PALETTE3	(3 << 8)	Sprite Palette Bank3
SP_PALETTE4	(4 << 8)	Sprite Palette Bank4
SP_PALETTE5	(5 << 8)	Sprite Palette Bank5
SP_PALETTE6	(6 << 8)	Sprite Palette Bank6
SP_PALETTE7	(7 << 8)	Sprite Palette Bank7
SP_PALETTE8	(8 << 8)	Sprite Palette Bank8
SP_PALETTE9	(9 << 8)	Sprite Palette Bank9
SP_PALETTE10	(10 << 8)	Sprite Palette Bank10
SP_PALETTE11	(11 << 8)	Sprite Palette Bank11
SP_PALETTE12	(12 << 8)	Sprite Palette Bank12
SP_PALETTE13	(13 << 8)	Sprite Palette Bank13
SP_PALETTE14	(14 << 8)	Sprite Palette Bank14
SP_PALETTE15	(15 << 8)	Sprite Palette Bank15
SP_VSIZE8	(0 << 6)	Sprite Character size 8 for Y
SP_VSIZE16	(1 << 6)	Sprite Character size 16 for Y
SP_VSIZE32	(2 << 6)	Sprite Character size 32 for Y
SP_VSIZE64	(3 << 6)	Sprite Character size 64 for Y
SP_HSIZE8	(0 << 4)	Sprite Character size 8 for X
SP_HSIZE16	(1 << 4)	Sprite Character size 16 for X
SP_HSIZE32	(2 << 4)	Sprite Character size 32 for X
SP_HSIZE64	(3 << 4)	Sprite Character size 64 for X
SPVFLIP_DISABLE	0x0000	V-Flip Disable
SPVFLIP_ENABLE	0x0008	V-Flip Enable
SPHFLIP_DISABLE	0x0000	H-Flip Disable
SPHFLIP_ENABLE	0x0004	H-Flip Enable
SP_COLOR4	0x0000	Sprite 4 Color Mode
SP_COLOR16	0x0001	Sprite 16 Color Mode
SP_COLOR64	0x0002	Sprite 64 Color Mode
SP_COLOR256	0x0003	Sprite 256 Color Mode
SpPalettePage0	0x0	Sprite Palette Page0
SpPalettePage1	0x1	Sprite Palette Page1
SpPalettePage2	0x2	Sprite Palette Page2
SpPalettePage3	0x3	Sprite Palette Page3
SpPalettePage4	0x4	Sprite Palette Page4
SpPalettePage5	0x5	Sprite Palette Page5
SpPalettePage6	0x6	Sprite Palette Page6
SpPalettePage7	0x7	Sprite Palette Page7

Name	Value	Remark
SpPalettePage8	0x8	Sprite Palette Page8
SpPalettePage9	0x9	Sprite Palette Page9
SpPalettePage10	0xa	Sprite Palette Page10
SpPalettePage11	0xb	Sprite Palette Page11
SpPalettePage12	0xc	Sprite Palette Page12
SpPalettePage13	0xd	Sprite Palette Page13
SpPalettePage14	0xe	Sprite Palette Page14
SpPalettePage15	0xf	Sprite Palette Page15
SpriteDepth1	0x0	Sprite Depth 1
SpriteDepth3	0x1	Sprite Depth 3
SpriteDepth5	0x2	Sprite Depth 5
SpriteDepth7	0x3	Sprite Depth 7
COLOR4	0x0	Text 4 Color Mode
COLOR16	0x1	Text 16 Color Mode
COLOR64	0x2	Text 64 Color Mode
COLOR256	0x3	Text 256 Color Mode
COLOR32768	0x4	Text 32768 Color Mode
COLOR65536	0x5	Text 65536 Color Mode
H_SIZE8	8	Character horizontal size 8
H_SIZE16	16	Character horizontal size 16
H_SIZE32	32	Character horizontal size 32
H_SIZE64	64	Character horizontal size 64
V_SIZE8	8	Character vertical size 8
V_SIZE16	16	Character vertical size 16
V_SIZE32	32	Character vertical size 32
V_SIZE64	64	Character vertical size 64
TextPalettePage0	0x0	Text Palette Page0
TextPalettePage1	0x1	Text Palette Page1
TextPalettePage2	0x2	Text Palette Page2
TextPalettePage3	0x3	Text Palette Page3
TextPalettePage4	0x4	Text Palette Page4
TextPalettePage5	0x5	Text Palette Page5
TextPalettePage6	0x6	Text Palette Page6
TextPalettePage7	0x7	Text Palette Page7
TextPalettePage8	0x8	Text Palette Page8
TextPalettePage9	0x9	Text Palette Page9
TextPalettePage10	0xa	Text Palette Page10
TextPalettePage11	0xb	Text Palette Page11
TextPalettePage12	0xc	Text Palette Page12
TextPalettePage13	0xd	Text Palette Page13
TextPalettePage14	0xe	Text Palette Page14

Name	Value	Remark
TextPalettePage15	0xf	Text Palette Page15
TextDepth0	0x0	Text Depth 0
TextDepth2	0x1	Text Depth 2
TextDepth4	0x2	Text Depth 4
TextDepth6	0x3	Text Depth 6
Text_CharMode	0x0	Text Character mode
Text_LinerMode	0x1	Text Linear mode
Text_AttnArray	0x0	Text Attribute in array
Text_AttnRegister	0x1	Text Attribute in register
BlendLevel0	0x0	Blending Level 0
BlendLevel1	0x1	Blending Level 1
BlendLevel2	0x2	Blending Level 2
BlendLevel3	0x3	Blending Level 3
TEXT1	0x0	Select TEXT 1
TEXT2	0x1	Select TEXT 2
TEXT3	0x2	Select TEXT 3
Text_RGBDDisable	0x0	Text High Color(32768/65536) disable
Text_RGBEnable	0x1	Text High Color(32768/65536) enable
Text_BlendDisable	0x0	Text Blending disable
Text_BlendEnable	0x1	Text Blending enable
Sp_BlendDisable	0x0	Sprite Blending disable
Sp_BlendEnable	0x1	Sprite Blending enable
Hv_Disable	0x0	Text Horizontal Movement disable
Hv_Enable	0x1	Text Horizontal Movement enable
WALLPAPER	0	Text Effect Mode - Wallpaper
HOFFSET	1	Text Effect Mode - Hoffset
BLEND	2	Text Effect Mode-Blend
FLIP	3	Text Effect Mode-Flip
LINES_SAME	0	H-Compression Data for Same mode
LINES_DIFFERENT	1	H-Compression Data for Different mode
DISABLE	0	Disable define
ENABLE	1	Enable define
SCROLLING_LEFT	0	Scrolling direction left
SCROLLING_RIGHT	1	Scrolling direction right
SCROLLING_UP	2	Scrolling direction up
SCROLLING_DOWN	3	Scrolling direction down
PALETTE_RAM	0	Palette Ram mode define
PALETTE_BUFF	1	Palette Buff mode define
PALETTE_RAMBANK0	0	Palette RAM Bank0
PALETTE_RAMBANK1	1	Palette RAM Bank1
PALETTE_RAMBANK2	2	Palette RAM Bank2

Name	Value	Remark
PALETTE_RAMBANK3	3	Palette RAM Bank3
TEXT_PALETTE0	0	Text select palette 0
SPRITE_PALETTE0	1	Sprite select palette 0
TEXT_PALETTE1	2	Text select palette 1
SPRITE_PALETTE1	3	Sprite select palette 1
PPU_DISABLE	0	PPU Control Flag
PPU_ENABLE	1	PPU Control Flag
TEXT_RELATED	0	PPU Control Flag
TEXT_DIRECT	1	PPU Control Flag
TEXT_TOP2BOT	0	PPU Control Flag
TEXT_BOT2TOP	1	PPU Control Flag
VGA_DISABLE	0	PPU Control Flag
FREE_DISABLE	0	PPU Control Flag
FREE_ENABLE	1	PPU Control Flag
NO_FLIP	0	Text/Sprite Flip Flag (No Flip)
H_FLIP	1	Text/Sprite Flip Flag (H Flip)
V_FLIP	2	Text/Sprite Flip Flag (V Flip)
HV_FLIP	3	Text/Sprite Flip Flag (HV Flip)
SP_DISABLE	0	Sprite Control Flag (Disable)
SP_ENABLE	1	Sprite Control Flag (Enable)
SP_COORD1	0	Sprite Control Flag
SP_COORD2	1	Sprite Control Flag
SP_NOROUND	0	Sprite Control Flag
SP_ROUND	1	Sprite Control Flag
SP_RELATED	0	Sprite Control Flag
SP_DIRECT	1	Sprite Control Flag
SP_MOSAIC	0	Sprite Effect Flag (Mosaic)
SP_ROTATE	1	Sprite Effect Flag (Rotate)
SP_ZOOM	2	Sprite Effect Flag (Zoom)
PALETTE16_SHARE	0	Palette Type
PALETTE16_SEPARATE	1	Palette Type
MODE_2D	0	Text Working Mode
MODE_HCMP	1	Text Working Mode
MODE_ROTATE	1	Text Working Mode
MODE_VCMP	2	Text Working Mode
MODE_HVCMP	3	Text Working Mode
BLEND4	0	4-Level Blend Mode
BLEND64	1	64-Level Blend Mode
TEXT_512x256	0	The Size of Text Layer
TEXT_512x512	1	The Size of Text Layer
TEXT_1024x512	2	The Size of Text Layer

Name	Value	Remark
TEXT_1024x1024	3	The Size of Text Layer
LINE_BASE	0	Line Base Mode
RGB565	0	RGB 565 mode
BPP_16	0	16bits/pixel mode
BPP_1	1	1bit/pixel mode
BPP_2	2	2bits/pixel mode
BPP_4	3	4bits/pixel mode
PaletteBank0	0	Palette RAM Bank0
PaletteBank1	1	Palette RAM Bank1
SP2D_MODE	0	Sprite Working Mode – 2D

3 Variables

3.1 Variables Table

All the global variables of PPU driver are defined in "PPUDrv.c". For the details, please refer to the Table3.1 and for typedef, and Table3.2 for variables.

Name	Type Define
U16	unsigned int
S16	int
U32	unsigned long
S32	long
SPRITE	Struct ^{*1}

Table3.1

Note:

1.

```

typedef struct                                //SPRITE Structure
{
    int      Idx;                             // ID number (0 ~ 32767)
    int      x;                               // X Position
    int      y;                               // Y Position
    int      xSize;                           // Sprite x size
    int      ySize;                           // Sprite y size
    const int *sprcell;                       // Pointer to sprite cell data set
    int      att0;                             // Sprite attribute 0
    int      att1;                             // Sprite attribute 1
    int      x1;                               // X1 position
    int      x2;                               // X2 position
    int      x3;                               // X3 position
    int      y1;                               // Y1 position
    int      y2;                               // Y2 position
    int      y3;                               // Y3 position
} SPRITE;

```

Name	Type	Remark
Text1IndexTable[TX1_INDEX_LENGTH] ^{*1}	int	Text1 index table (number array)
Text2IndexTable[TX2_INDEX_LENGTH] ^{*2}	int	Text2 index table (number array)
Text3IndexTable[TX3_INDEX_LENGTH] ^{*3}	int	Text3 index table (number array)
Text1AttrTable[TX1_ATT_LENGTH] ^{*5}	int	Text1 attribute table (attribute array)
Text2AttrTable[TX2_ATT_LENGTH] ^{*6}	int	Text2 attribute table (attribute array)
Text3AttrTable[TX3_ATT_LENGTH] ^{*7}	int	Text3 attribute table (attribute array)
Sprite_Buff[SP_MAX] ^{*9}	SPRITE	Sprite structure
SpriteBuf0[SP_MAX_BUF] ^{*10}	int	Sprite buffer 0
SpriteBuf1[SP_MAX_BUF] ^{*10}	int	Sprite buffer 1
spriteFlag	int	Sprite data update flag
textScroll	int	Text scrolling flag
text1_x	int	Text1 scrolling position X
text1_y	int	Text1 scrolling position Y
text2_x	int	Text2 scrolling position X
text2_y	int	Text2 scrolling position Y
text3_x	int	Text3 scrolling position X
text3_y	int	Text3 scrolling position Y
TextModeFlag	int	Text mode update flag
Text1Mode	int	Text1 mode value
Text2Mode	int	Text2 mode value
Text3Mode	int	Text3 mode value
ptrTxPalette0	U32	Text Palette0 pointer
ptrTxPalette1	U32	Text Palette1 pointer
ptrSpPalette0	U32	Sprite Palette0 pointer
ptrSpPalette1	U32	Sprite Palette1 pointer
HoffsetBuff[240] ^{*11}	int	Horizontal offset buffer
TxPaletteFlag0	int	Text palette 0 update flag
SpPaletteFlag0	int	Sprite palette 0 update flag
TxPaletteFlag1	int	Text palette 1 update flag
SpPaletteFlag1	int	Sprite palette 1 update flag
HoffsetFlag	int	Horizontal offset update flag
HCmpFlag	int	Horizontal compression update flag
LineMode	int	Line mode value
HCmpValue	int	Horizontal compression value
ptrHCmpTable	U32	Horizontal compression data pointer
VCmpFlag	int	Vertical compression update flag

Name	Type	Remark
VCmpValue	int	Vertical compression value
VCmpOffset	int	Vertical compression offset value
VCmpStep	int	Vertical compression step value
Text3RotateFlag	int	Text3 rotation update flag
Text3CenterX	int	Text3 X rotation center
Text3CenterY	int	Text3 Y rotation center
Text3OffsetX	int	Text3 X offset value
Text3OffsetY	int	Text3 Y offset value
old_game_time	U16	old game time
game_time	U16	game time
Scrolling_H1	int	Text1 horizontal start character
Scrolling_V1	int	Text1 vertical start character
Scrolling_H2	int	Text2 horizontal start character
Scrolling_V2	int	Text2 vertical start character
Scrolling_H3	int	Text3 horizontal start character
Scrolling_V3	int	Text3 vertical start character
iHoffsetCount	int	counter for Sea offset
ppu_flag	int	PPU active flag
ppu_count	U16	PPU frame counter
FadeInFlag	int	Fade in flag
FadeOutFlag	int	Fade out flag
FadePeriod	U16	frame interval for fade effect
cpu_done	int	CPU done flag for frame base mode
FadeCount	U16	Frame counter of fade

Table3.2
Note:

1. The default value of "TX1_INDEX_LENGTH" is set to 1024 in PPUDrv.h, but users could modify it according to the real application. If users don't need any index table (number array), please set it to 1.
2. The default value of "TX2_INDEX_LENGTH" is set to 1024 in PPUDrv.h, but users could modify it according to the real application. If users don't need any index table (number array), please set it to 1.
3. The default value of "TX3_INDEX_LENGTH" is set to 1024 in PPUDrv.h, but users could modify it according to the real application. If users don't need any index table (number array), please set it to 1.
4. The default value of "TX1_ATT_LENGTH" is set to 512 in PPUDrv.h, but users could modify it according to the real application. If users don't need any index table (number array), please set it to 1.
5. The default value of "TX2_ATT_LENGTH" is set to 512 in PPUDrv.h, but users could modify it according to the real application. If users don't need any index table (number array), please set it to 1.
6. The default value of "TX3_ATT_LENGTH" is set to 512 in PPUDrv.h, but users could modify it

according to the real application. If users don't need any index table (number array), please set it to 1.

7. The default value of "SP_MAX" is set to 256 in PPUDrv.h, but users could modify it according to the real application. If users don't use any sprite, please set it to 1.
8. The default value of "SP_MAX_BUF" is set to 1024 in PPUDrv.h, but users could modify it according to the real application. If users don't use any sprite, please set it to 1.
9. If the horizontal movement function is not used completely, please set the array length of "HoffsetBuff" to 1.

4 API Function

4.1 Macro Define

4.1.1 Macro: SetCpulrqON()

Syntax:

ASM: SetCpulrqON(void)

Parameters: None

Return Value: None

Library: < PPUDrv.h>

Remark: Set IRQ on

4.1.2 Macro: SetCpulrqOFF()

Syntax:

ASM: SetCpulrqOFF(void)

Parameters: None

Return Value: None

Library: < PPUDrv.h>

Remark: Set IRQ off

4.2 Function Define

4.2.1 Function: GetLongAddress()

Syntax:

C: GetLongAddress(int index)

Parameters: int index: The index of PPU resource address table

S16 Value: Start address of label Data

U32 Num: Data number to write to memory (words)

Return Value: 32-bit long address

Library: < PPUDrv2.asm>

Remark: Get 32-bit long address from PPU resource address table.

4.2.2 Function: MyMemSet()

Syntax:

C: MyMemSet(U32 DestMem, S16 value, U32 num)

Parameters: U32 DestMem : Start address of target memory
S16 Value: Start address of label Data
U32 Num: Data number to write to memory (words)

Return Value: None

Library: < PPUDrv.c>

Remark: Set the value to memory.

4.2.3 Function: SetDMACopy0()

Syntax:

C: SetDMACopy0 (U32 SrcMem, U32 DestMem, U32 Num)

Parameters: U32 SrcMem: Start address of source memory
U32 DestMem: Start address of target memory
U32 Num: DMA Transfer data number (words)

Return Value: None

Library: < PPUDrv.c>

Remark: 1. After transferring one data, both source and destination address will increase one.
2. Use DMA channel 0 to transfer data.

4.2.4 Function: SetColorPalette()

Syntax:

C: SetColorPalette(int col_st, int col_end, int value, int pal_sel)

Parameters: int col_st: Palette start index address
int col_end: Palette end index address
int value: Palette data value
int pal_sel: Palette selection
0: Text Palette 0
1: Sprite Palette 0
2: Text Palette 1
3: Sprite Palette 1

Return Value: None

Library: < PPUDrv.c>

Remark: Set color palette by manual data (all same data)

4.2.5 Function: WriteColorPalette()

Syntax:

C: WriteColorPalette(void)

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Write color palette data to palette RAM.

4.2.6 Function: SetOneTransparentColor()

Syntax:

C: SetOneTransparentColor(int pal_index, int value, int pal_sel)

Parameters:

int pal_index:	palette index address
int value:	0: clear transparent / 1: set transparent
int pal_sel:	Palette selection (0~3)
	0: Text Palette 0
	1: Sprite Palette 0
	2: Text Palette 1
	3: Sprite Palette 1

Return Value: None

Library: < PPUDrv.c>

Remark: Set one palette color to be transparent.

4.2.7 Function: SetTransparentColor()

Syntax:

C: SetTransparentColor(int col_st, int col_end, int value, int pal_sel)

Parameters:

int col_st:	palette start index address
int col_end:	palette end index address
int value:	0: clear transparent/1: set transparent
int pal_sel:	Palette selection (0~3)
	0: Text Palette 0
	1: Sprite Palette 0
	2: Text Palette 1
	3: Sprite Palette 1

Return Value: None

Library: < PPUDrv.c>
Remark: Set one block of palette colors to be transparent.

4.2.8 Function: SetAllTransparent()

Syntax:

C: SetAllTransparent(int value, int pal_sel)

Parameters: int value: 0: clear transparent/1: set transparent
int pal_sel: Palette selection (0~3)
0: Text Palette 0
1: Sprite Palette 0
2: Text Palette 1
3: Sprite Palette 1

Return Value: None

Library: < PPUDrv.c>

Remark: Set all palette color to be transparent.

4.2.9 Function: ClearOneSprite()

Syntax:

C: ClearOneSprite(int Spno)

Parameters: int Spno: Sprite number

Return Value: None

Library: < PPUDrv.c>

Remark: Clear one sprite data.

4.2.10 Function: ClearAllSprite()

Syntax:

C: ClearAllSprite(void)

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Clear all sprite data.

4.2.11 Function: ClearTextRegister()

Syntax:

C: ClearTextRegister(int txn)

Parameters: int txn: Text selection (0~2)
0: Text1
1: Text2
2: Text3

Return Value: None

Library: < PPUDrv.c>

Remark: Clear text register values.

4.2.12 Function: ClearAllRam()

Syntax:

C: ClearAllRam (void)

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Clear Internal SRAM (10KW, 0 ~ 0x27FF).

4.2.13 Function: ClearSimple()

Syntax:

C: ClearSimple (void)

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Clear PPU's buffer and initialize variables.

4.2.14 Function: ClearAllSetting()

Syntax:

C: ClearAllSetting (void)

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Clear all PPU registers/RAM and API's variable values.

4.2.15 Function: VblankIrqON()

Syntax:

C: VblankIrqON (void)

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Enable V-Blanking interrupt.

4.2.16 Function: VblankIrqOFF()

Syntax:

C: VblankIrqOFF (void)

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Disable V-Blanking interrupt.

4.2.17 Function: SpDMAIrqON()

Syntax:

C: SpDMAIrqON (void)

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Enable Sprite DMA transfer end interrupt.

4.2.18 Function: SpDMAIrqOFF()

Syntax:

C: SpDMAIrqOFF (void)

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Disable Sprite DMA transfer end interrupt.

4.2.19 Function: PaletteErrIrqON()

Syntax:

C: PaletteErrIrqON (void)
Parameters:
Return Value: None
Library: < PPUDrv.c>
Remark: Enable palette write error interrupt.

4.2.20 Function: PaletteErrIrqOFF()

Syntax:
C: PaletteErrIrqOFF (void)
Parameters:
Return Value: None
Library: < PPUDrv.c>
Remark: Disable palette write error interrupt.

4.2.21 Function: TxUnderRunIrqON()

Syntax:
C: TxUnderRunIrqON (void)
Parameters:
Return Value: None
Library: < PPUDrv.c>
Remark: Enable text under run interrupt.

4.2.22 Function: TxUnderRunIrqOFF()

Syntax:
C: TxUnderRunIrqOFF (void)
Parameters:
Return Value: None
Library: < PPUDrv.c>
Remark: Disable text under run interrupt.

4.2.23 Function: SpUnderRunIrqON();

Syntax:
C: SpUnderRunIrqON (void)

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Enable sprite under run interrupt.

4.2.24 Function: SpUnderRunIrqOFF()

Syntax:

C: SpUnderRunIrqOFF (void)

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Disable sprite under run interrupt.

4.2.25 Function: ClearVblankIRQ()

Syntax:

C: ClearVblankIRQ (void)

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Clear V-Blanking interrupt status.

4.2.26 Function: ClearSpDMAIRQ()

Syntax:

C: ClearSpDMAIRQ (void)

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Clear sprite DMA transfer end interrupt status.

4.2.27 Function: ClearPaletteErrIRQ()

Syntax:

C: ClearPaletteErrIRQ (void)

Parameters:

Return Value: None
Library: < PPUDrv.c>
Remark: Clear palette write error interrupt status.

4.2.28 Function: ClearTxUnderRunIRQ()

Syntax:

C: ClearTxUnderRunIRQ (void)

Parameters:

Return Value: None
Library: < PPUDrv.c>
Remark: Clear palette write error interrupt status.

4.2.29 Function: ClearSpUnderRunIRQ()

Syntax:

C: ClearSpUnderRunIRQ (void)

Parameters:

Return Value: None
Library: < PPUDrv.c>
Remark: Clear sprite under run interrupt status.

4.2.30 Function: InitPPU()

Syntax:

C: InitPPU(int ppu_en, int tx_direct, int tx_botup, int vga_en, int vga_nonintl, int free, int fb_en, int fb_format, int sp_25d, int fb_mono)

Parameters:

int ppu_en:	0:PPU_DISABLE/1:PPU_ENABLE
int tx_direct:	0:TEXT_RELATED/1:TEXT_DIRECT
int tx_botup:	0:TEXT_TOP2BOT/1:TEXT_BOT2TOP
int vga_en:	0:VGA_DISABLE/ others:NC
int vga_nonintl:	0:VGA_INTERLACE/1:VGA_NON_INTERLACE
int free:	0:FREE_DISABLE (Max. 22-bit address mode) 1:FREE_ENABLE (Max. 27-bit address mode)
int fb_en:	0:LINE_BASE / others:NC
int fb_format:	0:RGB565 / others:NC
int sp_25d:	0:SP2D_MODE / others:NC
int fb_mono:	0:16bpp/1:1bpp/2:2bpp/3:4bpp

Return Value: None
Library: < PPUDrv.c>
Remark: Initial PPU system control.

4.2.31 Function: ppu_on ()

Syntax:

C: ppu_on (void)

Parameters:

Return Value: None
Library: < PPUDrv.c>
Remark: Turn on PPU.

4.2.32 Function: ppu_off ()

Syntax:

C: ppu_off (void)

Parameters:

Return Value: None
Library: < PPUDrv.c>
Remark: Turn off PPU.

4.2.33 Function: WaitBlanking()

Syntax:

C: WaitBlanking (void)

Parameters:

Return Value: None
Library: < PPUDrv.c>
Remark: Wait Vertical blanking IRQ.

4.2.34 Function: SetTextPosition()

Syntax:

C: SetTextPosition(int text, int x, int y)
Parameters: int text: TEXT selection (0~2)
0:Text1
1:Text2

2:Text3

int x: x position

int y: y position

Return Value: None

Library: < PPUDrv.c>

Remark: 1. Set Text Position.

2. New Text position will be updated to register at v-blanking IRQ.

4.2.35 Function: Screen_Fade_Out()

Syntax:

C: int Screen_Fade_Out(int n)

Parameters: int n: Frame interval (n > 0)

Return Value: 0: fade out process isn't completed/1:fade out process is completed

Library: < PPUDrv.c>

Remark: 1. Screen fade out from original palette to black.

2. Execute fade out once every n frame till the screen is become black.

3. Return "1" when the screen is faded out to black completely.

4.2.36 Function: Screen_Fade_In()

Syntax:

C: int Screen_Fade_In(int n)

Parameters: int n: Frame interval (n > 0)

Return Value: 0: fade in process isn't completed/1:fade in process is completed

Library: < PPUDrv.c>

Remark: 1. Screen fade in from black to original palette.

2. Execute fade in once every n frame till the screen is recovered to original palette.

3. Return "1" when the screen is faded in to original palette completely.

4.2.37 Function: Sprite_Hit_Chk()

Syntax:

C: int Sprite_Hit_Chk(int Main_Index, int Enemy_Index);

Parameters: int Main_Index: SpriteBuf Index 1 (0~255)

int Enemy_Index: SpriteBuf Index 2 (0~255)

Return Value: 0: no hit /1: hit

Library: < PPUDrv.c>

Remark: Check Two Sprite x,y Position Hit

4.2.38 Function: HorizontalMovement()

Syntax:

C: HorizontalMovement(int StartScanline,int EndScanline,U32 IpScanline)

Parameters: int StartScanline: Start Scanline (0~239)
int EndScanline: End Scanline (0~239)
U32 IpScanline: Scanline Movement data table

Return Value: None

Library: < PPUDrv.c>

Remark: 1. Text layer every scanline Horizontal Movement.
2. Horizontal movement value will be updated from buffer to register at v-blanking IRQ.

4.2.39 Function: PPU_Service()

Syntax:

C: PPU_Service(void)

Parameters: None

Return Value: None

Library: < PPUDrv.c>

Remark: PPU service routine to update PPU registers and SRAM.

4.2.40 Function: SetTextEffect()

Syntax:

C: SetTextEffect(int nText,int nMode,int nEnable)

Parameters: int nText: Text selection (0~2) 0:TEXT1/1:TEXT2/2:TEXT3
int nMode: Text effect selection (0~3) 0:WALLPAPER/1:HOFFSET/2:BLEND/
3:FLIP
nEnable: Effect enable (0~3) 0:DISABLE/1:ENABLE
(Flip Mode): 0:NO_FLIP/1:H_FLIP/2:V_FLIP/3:HV_FLIP

Return Value: None

Library: < PPUDrv.c>

Remark: Set text effect.

4.2.41 Function: SetTxBlendMode()

Syntax:

C: SetTxBlendMode(int nText, int mode)

Parameters: int nText: Text selection (0~2) 0:TEXT1/1:TEXT2/2:TEXT3
int mode: Mode selection (0~1)
0:4-level blending mode
1:64-level blending mode

Return Value: None

Library: < PPUDrv.c>

Remark: Set text blend mode.

4.2.42 Function: SetBlendLevel4()

Syntax:

C: SetBlendLevel4(int nBlendLevel)

Parameters: int nBlendLevel: Blending level (0~3)

Return Value: None

Library: < PPUDrv.c>

Remark: Set 4-level blending level.

4.2.43 Function: SetTxBlendLevel64()

Syntax:

C: SetTxBlendLevel64(int nText, int nBlendLevel)

Parameters: int nText: Text selection (0~2) 0:TEXT1/1:TEXT2/2:TEXT3
int nBlendLevel: Blending level (0~63)

Return Value: None

Library: < PPUDrv.c>

Remark: Set text 64-level blending level.

4.2.44 Function: HCompressEffect()

Syntax:

C: HCompressEffect(int nMode, int nCompValue, U32 ptrCmpTable)

Parameters: int nMode: Mode selection (0~1) 0:LINES_SAME/1:LINES_DIFFERENT
int nCompValue: Horizontal compression value (1~128)
U32 ptrCmpTable: Horizontal compression value table address

Return Value: None

Library: < PPUDrv.c>

Remark:

1. Set Horizontal compression value.
2. If compress every line with the same value, just write value to nCompValue.
3. If compress every line with the different value, you need to set up one value table.
4. Horizontal compression effect is available only for Text1.

4.2.45 Function: VCompressEffect()

Syntax:

C: VCompressEffect(int nCompValue,int nCompStep,int nCompOffset)

Parameters:

int nCompValue:	vertical compression value
int nCompStep:	vertical compression step value
int nCompOffset:	vertical compression offset value

Return Value: None

Library: < PPUDrv.c>

Remark: Set vertical compression value.

4.2.46 Function: HOffsetEffect()

Syntax:

C: HOffsetEffect(int nMode, int nMoveValue, U32 ptrMoveTable)

Parameters:

int nMode:	Mode selection (0~1) 0:LINES_SAME/1:LINES_DIFFERENT
int nMoveValue:	Horizontal movement value
U32 ptrMoveTable:	Horizontal movement value table

Return Value: None

Library: < PPUDrv.c>

Remark:

1. Set horizontal movement value.
2. If move every line with the same value, just write value to nMoveValue
3. If move every line with the different value, you need to set up a value table

4.2.47 Function: Sea_Hoffset()

Syntax:

C: Sea_Hoffset(void)

Parameters:

Return Value: None

Library: < PPUDrv.c>
Remark: Horizontal movement as sea.

4.2.48 Function: FlipEffect()

Syntax:

C: FlipEffect(int nText,int nFlipMode,int nSizeX,int nSizeY)

Parameters: int nText: Text selection (0~2) 0:TEXT1/1:TEXT2/2:TEXT3
int nFlipMode: Flip Mode(0~3): 0:NO_FLIP/1:H_FLIP/2:V_FLIP/3:HV_FLIP
int nSizeX: Real text size x
int nSizeY: Real text size y

Return Value: None

Library: < PPUDrv.c>

Remark: 1. Text Flip Effect.
2. Bitmap-mode text can't use this function.

4.2.49 Function: InitCharacter()

Syntax:

C: InitCharacter(int nText, U32 IdxTab, U32 AttrTab, int nCellX, int nCellY, int nSizeX, int nSizeY, int nTextSize, int nColorMode, int nDepth, int nPalette)

Parameters: int nText: Text selection (0~2) 0:TEXT1/1:TEXT2/2:TEXT3
U32 IdxTab: Character index data address
U32 AttrTab: Attribute data address
int nCellX: Horizontal character size (8/16/32/64)
int nCellY: Vertical character size (8/16/32/64)
int nSizeX: Real text size x
int nSizeY: Real text size y
int nTextSize: The size of text layer (0~3)
0:512x256 / 1:512x512 / 2:1024x512 /3:1024x1024
int nColorMode: Color mode (0~5) 0:
4 color/1:16 color/2:64 color/3:256 color4:32768 color/5:65536 color
int nDepth: Depth layer (0~2)
0:Text Depth 0/1:Text Depth 2/2:Text Depth 4
int nPalette: 4-bit palette selection (0~15)

Return Value: None

Library: < PPUDrv.c>

Remark: Initial character-mode text

4.2.50 Function: fnShow_Chr()

Syntax:

C: fnShow_Chr(int nText,U32 IdxTab,U32 AttrTab,U32 CharData,int nCellX,int nCellY,int nSizeX,int nSizeY,int nTextSize,int nColor,int nDepth,int nPalette,int nPalBank)

Parameters:

int nText:	Text selection (0~2) 0:TEXT1/1:TEXT2/2:TEXT3
U32 IdxTab:	Character index data address
U32 AttrTab:	Attribute data address
U32 CharData:	Character data address
int nCellX:	Horizontal character size (8/16/32/64)
int nCellY:	Vertical character size (8/16/32/64)
int nSizeX:	Real text size x
int nSizeY:	Real text size y
int nTextSize:	The size of text layer (0~3) 0:512x256 / 1:512x512 / 2:1024x512 /3:1024x1024
int nColorMode:	Color mode (0~5) 0: 4 color/1:16 color/2:64 color/3:256 color4:32768 color/5:65536 color
int nDepth:	Depth layer (0~3) 0:Text Depth 0/1:Text Depth 2/2:Text Depth 4
int nPalette:	4-bit palette selection (0~15)
int nPalBank:	Text palette RAM bank (0~1)

Return Value: None

Library: < PPUDrv.c>

Remark: Show character-mode text.

4.2.51 Function: InitBitmap()

Syntax:

C: InitBitmap(int nText,U32 bmpdata,int nSizeX,int nSizeY,int nTextSize,int nColorMode,int nDepth,int nPalette)

Parameters:

int nText:	Text selection (0~3) 0:TEXT1/1:TEXT2/2:TEXT3
U32 bmpdata:	bitmap data address
int nSizeX:	Real text size x
int nSizeY:	Real text size y

int nTextSize: The size of text layer (0~3)
 0:512x256 / 1:512x512 / 2:1024x512 /3:1024x1024

int nColorMode: Color mode (0~5) 0:
 4 color/1:16 color/2:64 color/3:256 color4:32768 color/5:65536 color

int nDepth: Depth layer (0~2)
 0:Text Depth 0/1:Text Depth 2/2:Text Depth 4

int nPalette: 4-bit palette selection (0~15)

Return Value: None

Library: < PPUDrv.c>

Remark:

1. Initial bitmap-mode text.
2. The bitmap special mode is applied only when horizontal size is 320/640/512/1024.
3. The bitmap special mode means that PPU hardware computes start address of each line automatically.

4.2.52 Function: FnShow_Bmp()

Syntax:

C: fnShow_Bmp(int nText, U32 bmpdata, int nSizeX, int nSizeY, int nTextSize, int nColorMode, int nDepth, int nPalette, int nAuto, int nPalBank)

Parameters:

int nText: Text selection (0~2) 0:TEXT1/1:TEXT2/2:TEXT3

U32 bmpdata: bitmap data address

int nSizeX: Real text size x

int nSizeY: Real text size y

int nTextSize: The size of text layer (0~3)
 0:512x256 / 1:512x512 / 2:1024x512 /3:1024x1024

int nColorMode: Color mode (0~5) 0:
 4 color/1:16 color/2:64 color/3:256 color4:32768 color/5:65536 color

int nDepth: Depth layer (0~2)
 0:Text Depth 0/1:Text Depth 2/2:Text Depth 4

int nPalette: 4-bit palette selection (0~15)

int nPalBank: Text palette RAM bank (0~1)

Return Value: None

Library: < PPUDrv.c>

Remark:

1. Show bitmap-mode text.
2. The bitmap special mode is applied automatically only when horizontal size is 320/640/512/1024.

3. The bitmap special mode means that PPU hardware computes start address of each line automatically.

4.2.53 Function: ScrollingEffect()

Syntax:

C: ScrollingEffect(int nText,U32 IdxTab,int nHValue,int nVValue,int nSizeX,int nSizeY)

Parameters:

int nText:	Text selection (0~2) 0:TEXT1/1:TEXT2/2:TEXT3
U32 IdxTab:	Character index table address
int nHValue:	Horizontal movement value nHValue > 0 : scrolling left nHValue < 0 : scrolling right
int nVValue:	Vertical movement value nHValue > 0 : scrolling up nHValue < 0 : scrolling down
int nSizeX:	Real text size x
int nSizeY:	Real text size y

Return Value: None

Library: < PPUDrv.c>

Remark:

1. Scrolling text (character mode).
2. Bitmap-mode text can't use this function.

4.2.54 Function: SetSpriteCtrl()

Syntax:

C: SetSpriteCtrl(int spen, int coord, int round, int direct, int spnum)

Parameters:

int spen:	Sprite layer enable (0~1) 0:Disable/1:Enable
int coord:	Sprite coordinate system (0~1) 0:Center Coordinate mode/1:Left-top Coordinate mode
int round:	Sprite round-robin mode enable (0~1) 0:Disable/1:Enable
int direct:	Sprite addressing mode (0~1) 0:Related addressing mode 1:Direct addressing mode
int spnum:	Maximum available sprite number (1~256)

Return Value: None

Library: < PPUDrv.c>

Remark: Set sprite control mode.

4.2.55 Function: SetSpriteEffect()

Syntax:

C: SetSpriteEffect(int nMode, int nEnable)

Parameters: int nMode: Sprite special effect selection (0~2)
0: Mosaic effect
1: Rotate effect
2: Zoom effect
nEnable: Effect enable (0~1) 0:Disable/1:Enable

Return Value: None

Library: < PPUDrv.c>

Remark: Set sprite effect.

4.2.56 Function: SetSpBlendMode()

Syntax:

C: SetSpBlendMode(int mode)

Parameters: int mode: Blending mode (0~1) 0:4-level blending mode/1:64-level blending mode

Return Value: None

Library: < PPUDrv.c>

Remark: Set sprite blend mode.

4.2.57 Function: SetPaletteType()

Syntax:

C: SetPaletteType(int type)

Parameters: int type: Palette type (0~1)
0:PALETTE16_SHARE (Text and Sprite share the same 16-bit palette)
1:PALETTE16_SEPARATE (Text and Sprite have the different 16-bit palette)

Return Value: None

Library: < PPUDrv.c>

Remark: Set palette type.

4.2.58 Function: SetPaletteRamBank()

Syntax:

C: SetPaletteRamBank(int bank)
Parameters: int bank: Bank value (0~3)
Return Value: None
Library: < PPUDrv.c>
Remark: Set palette ram bank.

4.2.59 Function: SetTextMode()

Syntax:

C: SetTextMode(int nText, int mode)
Parameters: int nText: Text selection (0~2) 0:TEXT1/1:TEXT2/2:TEXT3
 int mode: Working mode (0~3)
 0: MODE_2D (TEXT1/2/3)
 1: MODE_HCMP/MODE_ROTATE (TEXT1)/(TEXT3)
 2: MODE_VCMP (TEXT1/TEXT2)
 3: MODE_HVCMP (TEXT1)
Return Value: None
Library: < PPUDrv.c>
Remark: Set text layer working mode.

4.2.60 Function: SetSpritePosition()

Syntax:

C: SetSpritePosition(int nSpldx, int x, int y, int scroll)
Parameters: int nSpldx: Sprite index (0~255)
 int x: X position increasement
 int y: Y position increasement
 int scroll: Sprite scroll enable (0~1)
 0: Disable
 1: Enable (when the sprite exceeds screen boundary, it will appear
 in opposite direction.)
Return Value: None
Library: < PPUDrv.c>
Remark: 1. Set Sprite Position.
 2. The original is on top-left conner of text screen.

4.2.61 Function: SpCoordConvert()

Syntax:

C: SpCoordConvert(int nSpldx, int *x, int *y)

Parameters: nSpldx: Sprite index (0~255)
*x: X position pointer
*y: Y position pointer

Return Value: None

Library: < PPUDrv.c>

Remark: 1. Sprite coordinate system conversion.
2. Convert left-top coordinate to center coordinate.

4.2.62 Function: Init_Sprite()

Syntax:

C: Init_Sprite(U32 SpCharData)

Parameters: U32 SpCharData: Sprite character data address

Return Value: None

Library: < PPUDrv.c>

Remark: 1. Initial sprite buffer and set sprite segment.
2. After calling this function, please run "Paint_Sprite()" to show sprites.

4.2.63 Function: Paint_Sprite()

Syntax:

C: Paint_Sprite()

Parameters: None

Return Value: None

Library: < PPUDrv.c>

Remark: 1. Show sprites on screen.
2. Before calling this function, please run "Init_Sprite()" once.

4.2.64 Function: SetSpRotate()

Syntax:

C: SetSpRotate (int nSpldx, int nRotate)

Parameters: int nSpldx: Sprite index (0~255)

int nRotate: Rotate level (0~63)

Return Value: None

Library: < PPUDrv.c>

Remark: Set Sprite rotate level.

4.2.65 Function: SetSpZoom()

Syntax:

C: SetSpZoom(int nSpldx, int nZoom)

Parameters: int nSpldx: Sprite index (0~255)
int nZoom: Zoom level (0~63)

Return Value: None

Library: < PPUDrv.c>

Remark: Set Sprite zoom level.

4.2.66 Function: SetSpDepth()

Syntax:

C: SetSpDepth(int nSpldx, int nDepth)

Parameters: int nSpldx: Sprite index (0~255)
int nDepth: Depth (0~3)
0: Sprite depth 1
1: Sprite depth 3
2: Sprite depth 5
3: Sprite depth 7

Return Value: None

Library: < PPUDrv.c>

Remark: Set Sprite Depth.

4.2.67 Function: SetSpBlend()

Syntax:

C: SetSpBlend(int nSpldx, int nEnable)

Parameters: int nSpldx: Sprite index (0~255)
int nEnable: Blend enable (0~1) 0:Disable/1:Enable

Return Value: None

Library: < PPUDrv.c>

Remark: Set Sprite Blending effect.

4.2.68 Function: SetSpMosaic()

Syntax:

C: SetSpMosaic(int nSpldx, int nMosaic)

Parameters: int nSpldx: Sprite index (0~255)
int nMosaic: Mosaic level (0~3)
0: No mosaic/1: 2x2 mosaic
2: 4x4 mosaic/3: 8x8 mosaic

Return Value: None

Library: < PPUDrv.c>

Remark: Set Sprite Mosaic Level.

4.2.69 Function: SetSpBlend64()

Syntax:

C: SetSpBlend64(int nSpldx, int nBlend)

Parameters: int nSpldx: Sprite index (0~255)
int nBlend: Blend level (0~63)

Return Value: None

Library: < PPUDrv.c>

Remark: Set Sprite 64-level Blending Level.

4.2.70 Function: SetSpriteFlip()

Syntax:

C: SetSpriteFlip(int nSpldx, int nFlip)

Parameters: int nSpldx: Sprite index (0~255)
int nFlip: Flip mode (0~3)
0: NO_FLIP/1: H_FLIP
2: V_FLIP/3: HV_FLIP

Return Value: None

Library: < PPUDrv.c>

Remark: Set Sprite Flip effect.

4.2.71 Function: SetSpPaletteBank()

Syntax:

C: SetSpPaletteBank(int nSpldx, int sel)

Parameters: int nSpldx: Sprite index (0~255)

	int sel:	Palette selection (0~15)
Return Value:	None	
Library:	< PPUDrv.c>	
Remark:	Set Sprite 4-bit Palette selection.	

4.2.72 Function: ClearlCache()

Syntax:

C: ClearlCache ()

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark: Clear CPU I-Cache.

4.2.73 Function: SetlCache()

Syntax:

C: SetlCache(int size)

Parameters: int size: Cache size (0~3) 0:8 words/1:1k bytes /2:2k bytes/3:4k bytes

Return Value: None

Library: < PPUDrv.c>

Remark: Set CPU Instruction Cache.

4.2.74 Function: VBLK_Service()

Syntax:

C: VBLK_Service ()

Parameters:

Return Value: None

Library: < PPUDrv.c>

Remark:

1. Call this function in IRQ5
2. PPU v-blanking IRQ service

4.2.75 Function: InitBitmapHVGA()

Syntax:

C: InitBitmapHVGA(int nText,U32 bmpdata,int nSizeX,int nSizeY,int nTextSize,int nColorMode,int nDepth,int nPalette)

Parameters:	int nText:	Text selection (0~2) 0:TEXT1/1:TEXT2/2:TEXT3
	U32 bmpdata:	bitmap data address

int nSizeX:	Real text size x
int nSizeY:	Real text size y
int nTextSize:	The size of text layer (0~3)
	0:512x256
	1:512x512
	2:1024x512
	3:1024x1024
int nColorMode:	Color mode (0~5)
	0:4 color/1:16 color/2:64 color /3:256 color/4:32768 color/5:65536 color
int nDepth:	Depth layer (0~2)
	0:Text Depth 0
	1:Text Depth 2
	2:Text Depth 4
int nPalette:	4-bit palette selection (0~15)

Return Value: None

Library: < PPUDrv.c>

Remark: Initial HVGA bitmap-mode text

4.2.76 Function: fnShow_Bmp_HVGA()

Syntax:

C: fnShow_Bmp_HVGA(int nText, U32 bmpdata, int nSizeX, int nSizeY, int nTextSize, int nColorMode, int nDepth, int nPalette, int nAuto, int nPalBank)

Parameters:

int nText:	Text selection (0~2) 0:TEXT1/1:TEXT2/2:TEXT3
U32 bmpdata:	bitmap data address
int nSizeX:	Real text size x
int nSizeY:	Real text size y
int nTextSize:	The size of text layer (0~3)
	0:512x256
	1:512x512
	2:1024x512
	3:1024x1024
int nColorMode:	Color mode (0~5)
	0:4 color/1:16 color/2:64 color /3:256 color/4:32768 color/5:65536 color
int nDepth:	Depth layer (0~2)
	0:Text Depth 0
	1:Text Depth 2

2:Text Depth 4

int nPalette: 4-bit palette selection (0~15)

int nPalBank: Text palette RAM bank (0~1)

Return Value: None

Library: < PPUDrv.c>

Remark:

1. Show HVGA bitmap-mode text.
2. The bitmap special mode is applied automatically only when horizontal size is 320/640/512/1024.
3. The bitmap special mode means that PPU hardware computes start address of each line automatically.

5 System Memory Requirement

5.1 Memory Requirement

The memory requirement of GPL951XX PPU driver API divided into two main parts: system working RAM and program code. For default setting in driver, the required working RAM size is about 2498 words(1 text 512x256 and 256 sprite) and the code size is 16346 words. The system working RAM size can be adjusted by users in different applications. For example, if Text size is reduce, and then the working RAM size will be decreased. If users don't need all 256 sprites, modify the value of "SP_MAX" in PPUDrv.h to reduce the working RAM size occupied by PPU driver.

6 Application Note

6.1 V-Blanking ISR

There is a V-Blanking ISR in PPU Driver, called VBLK_Service(). If the V-Blanking ISR is not satisfied your application, please modify it. Generalplus suggests that DO NOT modify the one in the PPUDrv.c. You can add one more function, says User_VBLK_Service, in your own project. So it will not be confused if Generalplus updates PPU Driver.

For example:

Before modify

_IRQ5:

```
push r1, r5 to [sp]
call _VBLK_Service //in PPUDrv.c
pop r1, r5 from [sp]
reti
```

After modify

_IRQ5:

```
push r1, r5 to [sp]
call User_VBLK_Service //in your own PPUDrv_User.c
pop r1, r5 from [sp]
reti
```