TMS99XX

1.0

Generated by Doxygen 1.9.1

Chapter 1

LICENSE

Chapter 2

TMS99XX Video Display Processor Library

Software Library for TI TMS9918,28,29 video display processors.

author: Jay Convertino

data: 2022.01.31

license: MIT

2.1 Release Versions

2.1.1 Current

- pre-alpha

2.1.2 Past

- none

2.2 Requirements

- sdcc v4.0.0 or greater

2.3 Building

Must be built with the arch libraries for correct linkage.

2.4 Documentation

- See doxygen generated document

Chapter 3

Data Structure Documentation

3.1 u_tms99XX_colorTable::s_cDataNibbles Struct Reference

`#include <tms99XXdatatypes.h>`

Data Fields

- uint8_t color2:4
- uint8_t color1:4

3.1.1 Detailed Description

contains color fields

3.1.2 Field Documentation

3.1.2.1 color1

`u_tms99XX_colorTable::s_cDataNibbles::color1`

top nibble, color for 1

3.1.2.2 color2

`u_tms99XX_colorTable::s_cDataNibbles::color2`

bottom nibble, color for 0

The documentation for this struct was generated from the following file:

- tms99XXdatatypes.h

3.2 u_tms99XX_spriteAttributeTable::s_dataNibbles Struct Reference

`#include <tms99XXdatatypes.h>`

Data Fields

- uint8_t verticalPos
- uint8_t horizontalPos
- uint8_t name
- uint8_t colorCode:4
- uint8_t na:3
- uint8_t earlyClockBit:1

3.2.1 Detailed Description

contains sprite attribute fields

3.2.2 Field Documentation

3.2.2.1 colorCode

`u_tms99XX_spriteAttributeTable::s_dataNibbles::colorCode`

color to set sprite pixels. this is the lower 4 bits (3 to 0)

3.2.2.2 earlyClockBit

`u_tms99XX_spriteAttributeTable::s_dataNibbles::earlyClockBit`

feature for slow movement of sprites off screen. bit 7

3.2.2.3 horizontalPos

`u_tms99XX_spriteAttributeTable::s_dataNibbles::horizontalPos`

horizontal position of sprite on screen.

3.2.2.4 na

`u_tms99XX_spriteAttributeTable::s_dataNibbles::na`

nothing, always set to 0. bits 6 to 4.

3.2.2.5 name

`u_tms99XX_spriteAttributeTable::s_dataNibbles::name`

name of sprite on screen (pointer to pattern table object).

3.2.2.6 verticalPos

`u_tms99XX_spriteAttributeTable::s_dataNibbles::verticalPos`

vertical position of sprite on screen.

The documentation for this struct was generated from the following file:

- tms99XXdatatypes.h

3.3 u_tms99XX_BMPpixelBlock::s_pDataNibbles Struct Reference

`#include <tms99XXdatatypes.h>`

Data Fields

- uint8_t colorB:4
- uint8_t colorA:4
- uint8_t colorD:4
- uint8_t colorC:4

3.3.1 Detailed Description

contains color fields for bitmap mode. see page 8-13 of TI-VDP-Programmers_guide.pdf |-------—|-------—|

color A	color B
color C	color D
-------—	-------—

The above is a mapping of the pixel pattern on screen. The resolution is 64x48 pixels. Each row has rules for indexing from the name table into the pattern table. Rows 0,4,8,12,16,20 index into the first pixel block at offsets of 8 bytes. A name table value of 0, will be pattern pixel block 0. A name table value of 1 will actually map to byte offset of 8, the next chunk of pixels. Pixels are chunked in 8 byte sections. Since the rows name lookup will loop after 4 rows. Meaning Row 1,5,9,13,17,21 name table will resolve to the same value in the pattern table. Same for Row 0,4,8,12,16,20. To be clear row 0, name table value 0 gets pattern 0. Row 4, name table value 0 gets pattern 0 as well.

3.3.2 Field Documentation

3.3.2.1 colorA

`u_tms99XX_BMPpixelBlock::s_pDataNibbles::colorA`

color lower nibble

3.3.2.2 colorB

`u_tms99XX_BMPpixelBlock::s_pDataNibbles::colorB`

color upper nibble

3.3.2.3 colorC

`u_tms99XX_BMPpixelBlock::s_pDataNibbles::colorC`

color lower nibble

3.3.2.4 colorD

`u_tms99XX_BMPpixelBlock::s_pDataNibbles::colorD`

color upper nibble

The documentation for this struct was generated from the following file:

- tms99XXdatatypes.h

3.4 s_tms99XX Struct Reference

Struct for containing TMS99XX instances.

`#include <tms99XXdatatypes.h>`

Data Fields

- uint16_t spritePatternAddr
- uint16_t spriteAttributeAddr
- uint16_t patternTableAddr
- uint16_t nameTableAddr
- uint16_t colorTableAddr
- uint8_t vdpMode
- uint8_t register0
- uint8_t register1
- uint8_t colorReg

3.4.1 Detailed Description

Struct for containing TMS99XX instances.

DATA STRUCTURES

3.4.2 Field Documentation

3.4.2.1 colorReg

`s_tms99XX::colorReg`

color sent to register 7, background/text color.

3.4.2.2 colorTableAddr

`s_tms99XX::colorTableAddr`

color table address in VRAM

3.4.2.3 nameTableAddr

`s_tms99XX::nameTableAddr`

name table address in VRAM

3.4.2.4 patternTableAddr

`s_tms99XX::patternTableAddr`

pattern table address in VRAM

3.4.2.5 register0

`s_tms99XX::register0`

register 0 contents

3.4.2.6 register1

`s_tms99XX::register1`

register 1 contents

3.4.2.7 spriteAttributeAddr

`s_tms99XX::spriteAttributeAddr`

sprite attribute table address in VRAM

3.4.2.8 spritePatternAddr

`s_tms99XX::spritePatternAddr`

sprite pattern table address in VRAM

3.4.2.9 vdpMode

`s_tms99XX::vdpMode`

contains current mode of the VDP

The documentation for this struct was generated from the following file:

- tms99XXdatatypes.h

3.5 s_tms99XX_nameTable Struct Reference

Struct for containing TMS99XX instances.

`#include <tms99XXdatatypes.h>`

Data Fields

- uint8_t data

3.5.1 Detailed Description

Struct for containing TMS99XX instances.

3.5.2 Field Documentation

3.5.2.1 data

`s_tms99XX_nameTable::data`

byte for storing name table data.

The documentation for this struct was generated from the following file:

- tms99XXdatatypes.h

3.6 s_tms99XX_spritePatternTable16x16 Struct Reference

Struct for containing a 16x16 sprite pattern.

`#include <tms99XXdatatypes.h>`

Data Fields

- uint8_t data [32]

3.6.1 Detailed Description

Struct for containing a 16x16 sprite pattern.

3.6.2 Field Documentation

3.6.2.1 data

`s_tms99XX_spritePatternTable16x16::data`

contains 16x16 matrix of sprite pattern data. First 8, upper left. Second 8, lower left. Third 8, upper right. Last 8, lower right.

The documentation for this struct was generated from the following file:

- tms99XXdatatypes.h

3.7 s_tms99XX_spritePatternTable8x8 Struct Reference

Struct for containing a 8x8 sprite pattern.

`#include <tms99XXdatatypes.h>`

Data Fields

- uint8_t data [8]

3.7.1 Detailed Description

Struct for containing a 8x8 sprite pattern.

3.7.2 Field Documentation

3.7.2.1 data

`s_tms99XX_spritePatternTable8x8::data`

contains 8x8 matrix of sprite pattern data

The documentation for this struct was generated from the following file:

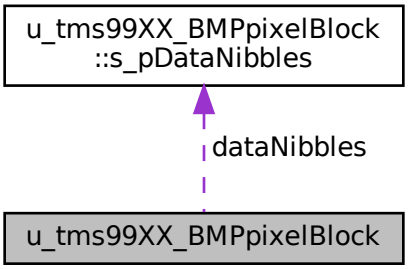
- tms99XXdatatypes.h

3.8 u_tms99XX_BMPpixelBlock Union Reference

Struct for containing a single multicolor mode pixel block.

`#include <tms99XXdatatypes.h>`

Collaboration diagram for u_tms99XX_BMPpixelBlock:



Data Structures

- struct s_pDataNibbles

Data Fields

- struct u_tms99XX_BMPpixelBlock::s_pDataNibbles dataNibbles
- uint8_t data [2]

3.8.1 Detailed Description

Struct for containing a single multicolor mode pixel block.

3.8.2 Field Documentation

3.8.2.1 data

`u_tms99XX_BMPpixelBlock::data`

array of 2 bytes for a 2x2 4 pixel map.

3.8.2.2 dataNibbles

`struct u_tms99XX_BMPpixelBlock::s_pDataNibbles u_tms99XX_BMPpixelBlock::dataNibbles`

The documentation for this union was generated from the following file:

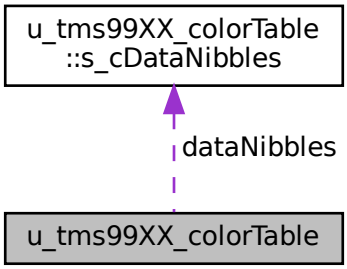
- tms99XXdatatypes.h

3.9 u_tms99XX_colorTable Union Reference

Union for containing TMS99XX color table data.

`#include <tms99XXdatatypes.h>`

Collaboration diagram for u_tms99XX_colorTable:



Data Structures

- struct s_cDataNibbles

Data Fields

- struct u_tms99XX_colorTable::s_cDataNibbles dataNibbles
- uint8_t data

3.9.1 Detailed Description

Union for containing TMS99XX color table data.

3.9.2 Field Documentation

3.9.2.1 data

`u_tms99XX_colorTable::data`

contains color data byte

3.9.2.2 dataNibbles

`struct u_tms99XX_colorTable::s_cDataNibbles u_tms99XX_colorTable::dataNibbles`

The documentation for this union was generated from the following file:

- tms99XXdatatypes.h

3.10 u_tms99XX_patternTable8x8 Union Reference

Struct for containing a 8x8 pattern table.

`#include <tms99XXdatatypes.h>`

Data Fields

- uint8_t data [8]

3.10.1 Detailed Description

Struct for containing a 8x8 pattern table.

3.10.2 Field Documentation

3.10.2.1 data

`u_tms99XX_patternTable8x8::data`

array of 8 bytes for a 8x8 matrix.

The documentation for this union was generated from the following file:

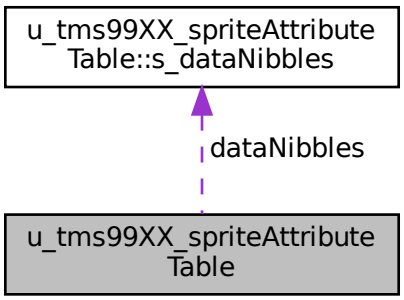
- tms99XXdatatypes.h

3.11 u_tms99XX_spriteAttributeTable Union Reference

union for containing sprite attribute data.

`#include <tms99XXdatatypes.h>`

Collaboration diagram for u_tms99XX_spriteAttributeTable:



Data Structures

- struct s_dataNibbles

Data Fields

- struct u_tms99XX_spriteAttributeTable::s_dataNibbles dataNibbles
- uint8_t data [4]

3.11.1 Detailed Description

union for containing sprite attribute data.

3.11.2 Field Documentation

3.11.2.1 data

`u_tms99XX_spriteAttributeTable::data`

array of sprite attribute data

3.11.2.2 dataNibbles

`struct u_tms99XX_spriteAttributeTable::s_dataNibbles u_tms99XX_spriteAttributeTable::dataNibbles`

The documentation for this union was generated from the following file:

- tms99XXdatatypes.h

Chapter 4

File Documentation

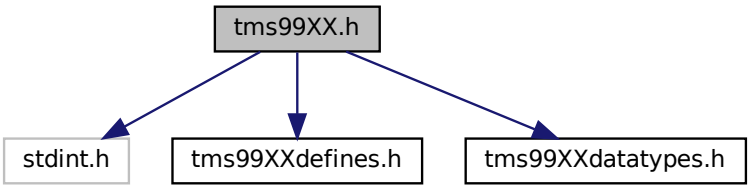
4.1 LICENSE.md File Reference

4.2 README.md File Reference

4.3 tms99XX.h File Reference

```
#include <stdint.h>
#include <tms99XXdefines.h>
#include <tms99XXdatatypes.h>
```

Include dependency graph for tms99XX.h:



Functions

- void initTMS99XX (struct s_tms99XX *const p_tms99XX, uint8_t vdpMode, uint8_t backColor)

  Initialize TMS99XX struct with ports to use for input output, must match direction registers above. Register 1 has VRAM set to 16k, VDP mode set, the blank enabled(0), and all other bits in there disabled state. Memory addresses will match defines. These could be changed latter and then the setTMS99XXmode run for custom settings on the fly. Better option is to change the defines in this headed to your use case.

- void setTMS99XXmode (struct s_tms99XX *const p_tms99XX, uint8_t vdpMode)

  Set the TMS99XX mode to one of 4. Text, Graphics I, Graphics II, and bitmap. This will also reset all addresses for the needed mode.

- void setTMS99XXblank (struct s_tms99XX ∗const p_tms99XX, uint8_t mode)

  Set the TMS99XX to blank the current sprite and pattern planes.
- void setTMS99XXirq (struct s_tms99XX ∗const p_tms99XX, uint8_t mode)

  Set the TMS99XX to irq to enabled or disabled.
- void setTMS99XXspriteSize (struct s_tms99XX ∗const p_tms99XX, uint8_t mode)

  Set the TMS99XX to sprite size to 8x8 or 16x16.
- void setTMS99XXspriteMagnify (struct s_tms99XX ∗const p_tms99XX, uint8_t mode)

  Set the TMS99XX to sprite magnify to on or off (double set size).
- void setTMS99XXtxtColor (struct s_tms99XX ∗const p_tms99XX, uint8_t color)

  Set the TMS99XX text color in text mode.
- void setTMS99XXbackgroundColor (struct s_tms99XX ∗const p_tms99XX, uint8_t color)

  Set the TMS99XX background color.
- void setTMS99XXreg (struct s_tms99XX ∗const p_tms99XX, uint8_t regNum, uint8_t regData)

  Set a register with a 8 bit value.
- int setTMS99XXvramTableData (struct s_tms99XX ∗const p_tms99XX, uint16_t tableAddr, void const ∗const p_data, int startNum, int number, int size)

  Write a pattern or patterns into vram pattern table. Alighned to pattern data size.
- void setTMS99XXvramWriteAddr (struct s_tms99XX ∗const p_tms99XX, uint16_t vramAddr)

  Set the start of the VRAM address to write to. After this is set writes will auto increment the address.
- void setTMS99XXvramReadAddr (struct s_tms99XX ∗const p_tms99XX, uint16_t vramAddr)

  Set the start of the VRAM address to read to. After this is set read will auto increment the address.
- int setTMS99XXvramData (struct s_tms99XX ∗const p_tms99XX, void const ∗const p_data, int size)

  Write array of byte data to VRAM.
- int setTMS99XXvramConstData (struct s_tms99XX ∗const p_tms99XX, uint8_t const data, int size)

  Set all data in VRAM to a constant value of some size.
- void setTMS99XXvramSpriteTerm (struct s_tms99XX ∗const p_tms99XX, uint8_t const num)

  Set all vertical field of selected sprite number to the 0xD0. The sprite terminator.
- int getTMS99XXvramData (struct s_tms99XX ∗const p_tms99XX, void ∗p_data, int size)

  Read array of byte data to VRAM.
- uint8_t getTMS99XXstatus (struct s_tms99XX ∗const p_tms99XX)

  Read status register of VDP.
- void clearTMS99XXvramData (struct s_tms99XX ∗const p_tms99XX)

  Clear all data from VRAM from 0x0000 to 0x3FFF. This will block till it has cleared all data.
- uint8_t checkTMS99XXvram (struct s_tms99XX ∗const p_tms99XX)

  Test all VRAM. This will block till all data written.

4.3.1 Function Documentation

4.3.1.1 checkTMS99XXvram()

```
uint8_t checkTMS99XXvram (
            struct s_tms99XX *const p_tms99XX )
```

Test all VRAM. This will block till all data written.

Parameters

p_tms99XX	pointer to struct to contain data.

Returns

0 for error, 1 for pass.

4.3.1.2 clearTMS99XXvramData()

```
void clearTMS99XXvramData (
            struct s_tms99XX *const p_tms99XX )
```

Clear all data from VRAM from 0x0000 to 0x3FFF. This will block till it has cleared all data.

Parameters

p_tms99XX	pointer to struct to contain data.

4.3.1.3 getTMS99XXstatus()

```
uint8_t getTMS99XXstatus (
            struct s_tms99XX *const p_tms99XX )
```

Read status register of VDP.

Parameters

p_tms99XX	pointer to struct to contain data.

Returns

Status register data byte.

4.3.1.4 getTMS99XXvramData()

```
int getTMS99XXvramData (
            struct s_tms99XX *const p_tms99XX,
            void * p_data,
            int size )
```

Read array of byte data to VRAM.

Parameters

p_tms99XX	pointer to struct to contain data.
p_data	pointer to data to store read data.
size	number of bytes to read from vram.

Returns

actual number of bytes read.

4.3.1.5 initTMS99XX()

```
void initTMS99XX (
            struct s_tms99XX *const p_tms99XX,
            uint8_t vdpMode,
            uint8_t backColor )
```

Initialize TMS99XX struct with ports to use for input output, must match direction registers above. Register 1 has VRAM set to 16k, VDP mode set, the blank enabled(0), and all other bits in there disabled state. Memory addresses will match defines. These could be changed latter and then the setTMS99XXmode run for custom settings on the fly. Better option is to change the defines in this headed to your use case.

the below includes define other tms stuffs see them for more info METHODS

Parameters

p_tms99XX	pointer to struct to contain data.
vdpMode	set or change the mode, 0 = Graphics I, 1 = Graphics II, 2 = bitmap, 4 = Text.
backColor	set background color to a 4 bit value.

4.3.1.6 setTMS99XXbackgroundColor()

```
void setTMS99XXbackgroundColor (
            struct s_tms99XX *const p_tms99XX,
            uint8_t color )
```

Set the TMS99XX background color.

Parameters

p_tms99XX	pointer to struct to contain data.
color	4 bit color value.

4.3.1.7 setTMS99XXblank()

```
void setTMS99XXblank (
            struct s_tms99XX *const p_tms99XX,
            uint8_t mode )
```

Set the TMS99XX to blank the current sprite and pattern planes.

Parameters

p_tms99XX	pointer to struct to contain data.
mode	1 is blank on, 0 is blank off

4.3.1.8 setTMS99XXirq()

```
void setTMS99XXirq (
            struct s_tms99XX *const p_tms99XX,
            uint8_t mode )
```

Set the TMS99XX to irq to enabled or disabled.

Parameters

p_tms99XX	pointer to struct to contain data.
mode	0 is off, anything else is on.

4.3.1.9 setTMS99XXmode()

```
void setTMS99XXmode (
            struct s_tms99XX *const p_tms99XX,
            uint8_t vdpMode )
```

Set the TMS99XX mode to one of 4. Text, Graphics I, Graphics II, and bitmap. This will also reset all addresses for the needed mode.

Parameters

p_tms99XX	pointer to struct to contain data.
vdpMode	set or change the mode, 0 = Graphics I, 1 = Graphics II, 2 = bitmap, 4 = Text.

4.3.1.10 setTMS99XXreg()

```
void setTMS99XXreg (
```

```
            struct s_tms99XX *const p_tms99XX,
            uint8_t regNum,
            uint8_t regData )
```

Set a register with a 8 bit value.

Parameters

p_tms99XX	pointer to struct to contain data.
regNum	which register to write to. 0 to 7.
regData	data to write to register.

4.3.1.11 setTMS99XXspriteMagnify()

```
void setTMS99XXspriteMagnify (
            struct s_tms99XX *const p_tms99XX,
            uint8_t mode )
```

Set the TMS99XX to sprite magnify to on or off (double set size).

Parameters

p_tms99XX	pointer to struct to contain data.
mode	0 is off, anything else is double sprite size.

4.3.1.12 setTMS99XXspriteSize()

```
void setTMS99XXspriteSize (
            struct s_tms99XX *const p_tms99XX,
            uint8_t mode )
```

Set the TMS99XX to sprite size to 8x8 or 16x16.

Parameters

p_tms99XX	pointer to struct to contain data.
mode	0 is 8x8, anything else is 16x16.

4.3.1.13 setTMS99XXtxtColor()

```
void setTMS99XXtxtColor (
            struct s_tms99XX *const p_tms99XX,
            uint8_t color )
```

Set the TMS99XX text color in text mode.

Parameters

p_tms99XX	pointer to struct to contain data.
color	4 bit color value.

4.3.1.14 setTMS99XXvramConstData()

```
int setTMS99XXvramConstData (
            struct s_tms99XX *const p_tms99XX,
            uint8_t const data,
            int size )
```

Set all data in VRAM to a constant value of some size.

Parameters

p_tms99XX	pointer to struct to contain data.
data	the constant to write.
size	number of bytes to set.

Returns

actual number of bytes wrote.

4.3.1.15 setTMS99XXvramData()

```
int setTMS99XXvramData (
            struct s_tms99XX *const p_tms99XX,
            void const *const p_data,
            int size )
```

Write array of byte data to VRAM.

Parameters

p_tms99XX	pointer to struct to contain data.
p_data	pointer to data to write to vdp.
size	number of bytes to write to VRAM.

Returns

actual number of bytes wrote.

4.3.1.16 setTMS99XXvramReadAddr()

```
void setTMS99XXvramReadAddr (
            struct s_tms99XX *const p_tms99XX,
            uint16_t vramAddr )
```

Set the start of the VRAM address to read to. After this is set read will auto increment the address.

Parameters

p_tms99XX	pointer to struct to contain data.
vramAddr	14 bit address into the vram.

4.3.1.17 setTMS99XXvramSpriteTerm()

```
void setTMS99XXvramSpriteTerm (
            struct s_tms99XX *const p_tms99XX,
            uint8_t const num )
```

Set all vertical field of selected sprite number to the 0xD0. The sprite terminator.

Parameters

p_tms99XX	pointer to struct to contain data.
num	the sprite number 0 to 31 to terminate.

4.3.1.18 setTMS99XXvramTableData()

```
int setTMS99XXvramTableData (
            struct s_tms99XX *const p_tms99XX,
            uint16_t tableAddr,
            void const *const p_data,
            int startNum,
            int number,
            int size )
```

Write a pattern or patterns into vram pattern table. Alighned to pattern data size.

Parameters

p_tms99XX	pointer to struct to contain data.
tableAddr	table start address, exe p_tms99XX->spriteAttributeAddr
p_data	void pointer data array that contains pattern data objects.
startNum	adds a offset to the base vram address.
number	quantity of patterns to write linearly.
size	of the data members in the table (all tables of member data, sizeof(data))

Returns

number of bytes actually wrote

4.3.1.19 setTMS99XXvramWriteAddr()

```
void setTMS99XXvramWriteAddr (
            struct s_tms99XX *const p_tms99XX,
            uint16_t vramAddr )
```

Set the start of the VRAM address to write to. After this is set writes will auto increment the address.

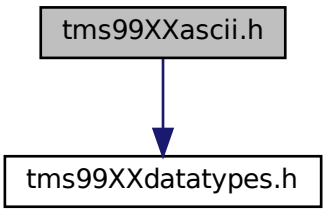
Parameters

p_tms99XX	pointer to struct to contain data.
vramAddr	14 bit address into the vram.

4.4 tms99XXascii.h File Reference

```
#include <tms99XXdatatypes.h>
```

Include dependency graph for tms99XXascii.h:



Variables

- const union u_tms99XX_patternTable8x8 c_tms99XX_ascii [ ]

4.4.1 Variable Documentation

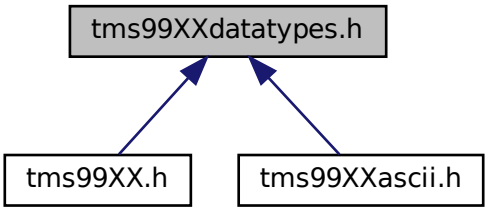
4.4.1.1 c_tms99XX_ascii

```
const union u_tms99XX_patternTable8x8 c_tms99XX_ascii[]
```

From TMS9918 datasheet Fixed a few bugs, duplicate > and bad lower case letters Added nulls for first 32 to pad out struct, easier to just create strings and go.

4.5 tms99XXdatatypes.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct s_tms99XX

  Struct for containing TMS99XX instances.
- union u_tms99XX_patternTable8x8

  Struct for containing a 8x8 pattern table.
- union u_tms99XX_BMPpixelBlock

  Struct for containing a single multicolor mode pixel block.
- struct u_tms99XX_BMPpixelBlock::s_pDataNibbles
- struct s_tms99XX_nameTable

  Struct for containing TMS99XX instances.
- union u_tms99XX_colorTable

  Union for containing TMS99XX color table data.
- struct u_tms99XX_colorTable::s_cDataNibbles
- struct s_tms99XX_spritePatternTable8x8

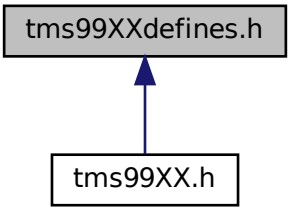
  Struct for containing a 8x8 sprite pattern.
- struct s_tms99XX_spritePatternTable16x16

  Struct for containing a 16x16 sprite pattern.
- union u_tms99XX_spriteAttributeTable

  union for containing sprite attribute data.
- struct u_tms99XX_spriteAttributeTable::s_dataNibbles

4.6 tms99XXdefines.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define GFXI_MODE 0
- #define GFXII_MODE 1
- #define BMP_MODE 2
- #define TXT_MODE 4
- #define VMEM_AMT_BIT 7
- #define BLK_SCRN_BIT 6
- #define IRQ_BIT 5
- #define SPRITE_SIZE_BIT 1
- #define SPRITE_MAG_BIT 0
- #define REGISTER_0 0
- #define REGISTER_1 1
- #define REGISTER_2 2
- #define REGISTER_3 3
- #define REGISTER_4 4
- #define REGISTER_5 5
- #define REGISTER_6 6
- #define REGISTER_7 7
- #define NAME_TABLE_ADDR 0x3800
- #define NAME_TABLE_ADDR_SCALE 10
- #define COLOR_TABLE_ADDR 0x2000
- #define COLOR_TABLE_ADDR_SCALE 6
- #define PATTERN_TABLE_ADDR 0x0000
- #define PATTERN_TABLE_ADDR_SCALE 11
- #define SPRITE_ATTRIBUTE_TABLE_ADDR 0x3B80
- #define SPRITE_ATTRIBUTE_TABLE_ADDR_SCALE 7
- #define SPRITE_PATTERN_TABLE_ADDR 0x1800
- #define SPRITE_PATTERN_TABLE_ADDR_SCALE 11
- #define TMS_TRANSPARENT 0x00
- #define TMS_BLACK 0x01
- #define TMS_MEDIUM_GREEN 0x02
- #define TMS_LIGHT_GREEN 0x03
- #define TMS_DARK_BLUE 0x04
- #define TMS_LIGHT_BLUE 0x05

- #define TMS_DARK_RED 0x06
- #define TMS_CYAN 7
- #define TMS_MEDIUM_RED 0x08
- #define TMS_LIGHT_RED 0x09
- #define TMS_DARK_YELLOW 0x0A
- #define TMS_LIGHT_YELLOW 0x0B
- #define TMS_DARK_GREEN 0x0C
- #define TMS_MAGENTA 0x0D
- #define TMS_GREY 0x0E
- #define TMS_WHITE 0x0F
- #define MEM_SIZE (1 << 14)
- #define SPRITE_TERM 0xD0

4.6.1 Macro Definition Documentation

4.6.1.1 BLK_SCRN_BIT

```
#define BLK_SCRN_BIT 6
```

blank screen bit

4.6.1.2 BMP_MODE

```
#define BMP_MODE 2
```

mode bit for bitmap(multicolor) mode

4.6.1.3 COLOR_TABLE_ADDR

```
#define COLOR_TABLE_ADDR 0x2000
```

default address for color table (R3 * 0x40[2^6]) 0x2000 address is set when R3 is 0x80.

GFXII can only be 0x0000(0x7F) or 0x2000(0xFF) Init takes care of this, but this can be easily broken if a bad address is used for the color table.

4.6.1.4 COLOR_TABLE_ADDR_SCALE

```
#define COLOR_TABLE_ADDR_SCALE 6
```

value to scale the color table address

4.6.1.5 GFXI_MODE

```
#define GFXI_MODE 0
```

DEFINES VDP MODE DEFINES

mode bit for graphics I mode

4.6.1.6 GFXII_MODE

```
#define GFXII_MODE 1
```

mode bit for graphics II mode

4.6.1.7 IRQ_BIT

```
#define IRQ_BIT 5
```

IRQ bit number

4.6.1.8 MEM_SIZE

```
#define MEM_SIZE (1 << 14)
```

MISC DEFINES

16K of memory: 0 to 0x3FFF (0x4000)

4.6.1.9 NAME_TABLE_ADDR

```
#define NAME_TABLE_ADDR 0x3800
```

VRAM ADDRESS DEFINES

default address for name table (R2 $*$ 0x400[2^10]) 0x3800 address is set when R2 is 0x0E

4.6.1.10 NAME_TABLE_ADDR_SCALE

```
#define NAME_TABLE_ADDR_SCALE 10
```

value to scale the name table address

4.6.1.11 PATTERN_TABLE_ADDR

```
#define PATTERN_TABLE_ADDR 0x0000
```

default address for pattern table (R4 $*$ 0x800[2^11]) 0x0000 address is set when R4 is 0x00

GFXII can only be 0x0000(0x03) or 0x2000(0x07) Init takes care of this, but this can be easily broken if a bad address is used for the pattern table.

4.6.1.12 PATTERN_TABLE_ADDR_SCALE

```
#define PATTERN_TABLE_ADDR_SCALE 11
```

value to scale the pattern table address

4.6.1.13 REGISTER_0

```
#define REGISTER_0 0
```

REGISTER DEFINES

mode and external video bits

4.6.1.14 REGISTER_1

```
#define REGISTER_1 1
```

mode, sprite, interrupt, blank, and memory bits.

4.6.1.15 REGISTER_2

```
#define REGISTER_2 2
```

name table address

4.6.1.16 REGISTER_3

```
#define REGISTER_3 3
```

color table address

4.6.1.17 REGISTER_4

```
#define REGISTER_4 4
```

pattern table address

4.6.1.18 REGISTER_5

```
#define REGISTER_5 5
```

sprite attribute address

4.6.1.19 REGISTER_6

```
#define REGISTER_6 6
```

sprite pattern address

4.6.1.20 REGISTER_7

```
#define REGISTER_7 7
```

background, text color

4.6.1.21 SPRITE_ATTRIBUTE_TABLE_ADDR

```
#define SPRITE_ATTRIBUTE_TABLE_ADDR 0x3B80
```

default address for sprite attribute table (R5 ∗ 0x80[2^7]) 0x3B80 address is set when R5 is 0x77

4.6.1.22 SPRITE_ATTRIBUTE_TABLE_ADDR_SCALE

```
#define SPRITE_ATTRIBUTE_TABLE_ADDR_SCALE 7
```

value to scale the sprite attribute address

4.6.1.23 SPRITE_MAG_BIT

```
#define SPRITE_MAG_BIT 0
```

sprite magnification bit

4.6.1.24 SPRITE_PATTERN_TABLE_ADDR

```
#define SPRITE_PATTERN_TABLE_ADDR 0x1800
```

default address for sprite pattern table (R6 ∗ 0x800[2^11]) 0x1800 address is set when R6 is 0x03

4.6.1.25 SPRITE_PATTERN_TABLE_ADDR_SCALE

```
#define SPRITE_PATTERN_TABLE_ADDR_SCALE 11
```

value to scale the sprite attribute address

4.6.1.26 SPRITE_SIZE_BIT

```
#define SPRITE_SIZE_BIT 1
```

sprite size bit

4.6.1.27 SPRITE_TERM

```
#define SPRITE_TERM 0xD0
```

Vertical field can contain a terminator value of 0xD0 to stop sprite processing.

4.6.1.28 TMS_BLACK

```
#define TMS_BLACK 0x01
```

color black

4.6.1.29 TMS_CYAN

```
#define TMS_CYAN 7
```

color cyan

4.6.1.30 TMS_DARK_BLUE

```
#define TMS_DARK_BLUE 0x04
```

color dark blue

4.6.1.31 TMS_DARK_GREEN

```
#define TMS_DARK_GREEN 0x0C
```

register

4.6.1.32 TMS_DARK_RED

```
#define TMS_DARK_RED 0x06
```

color dark red

4.6.1.33 TMS_DARK_YELLOW

```
#define TMS_DARK_YELLOW 0x0A
```

register

4.6.1.34 TMS_GREY

```
#define TMS_GREY 0x0E
```

register

4.6.1.35 TMS_LIGHT_BLUE

```
#define TMS_LIGHT_BLUE 0x05
```

color light blue

4.6.1.36 TMS_LIGHT_GREEN

```
#define TMS_LIGHT_GREEN 0x03
```

color light green

4.6.1.37 TMS_LIGHT_RED

```
#define TMS_LIGHT_RED 0x09
```

register

4.6.1.38 TMS_LIGHT_YELLOW

```
#define TMS_LIGHT_YELLOW 0x0B
```

register

4.6.1.39 TMS_MAGENTA

```
#define TMS_MAGENTA 0x0D
```

register

4.6.1.40 TMS_MEDIUM_GREEN

```
#define TMS_MEDIUM_GREEN 0x02
```

color medium green

4.6.1.41 TMS_MEDIUM_RED

```
#define TMS_MEDIUM_RED 0x08
```

color medium red

4.6.1.42 TMS_TRANSPARENT

```
#define TMS_TRANSPARENT 0x00
```

COLOR DEFINES

transparent for all plans/sprites

4.6.1.43 TMS_WHITE

```
#define TMS_WHITE 0x0F
```

register

4.6.1.44 TXT_MODE

```
#define TXT_MODE 4
```

mode bit for text mode

4.6.1.45 VMEM_AMT_BIT

```
#define VMEM_AMT_BIT 7
```

register 1 bit defines

amount of vram bit

Index