# Описание используемых типов

## **LCD\_ExGpio\_type**

|  |  |  |
| --- | --- | --- |
| Поле структуры | Допустимые значения | Описание |
| StartPosition | 1…1024 | Начальная позиция, в пиксельклоках |
| StopPosition | 1…1024 | Конечная позиция, в пиксельклоках |
| ActiveLevel | 0 или 1 | Активный уровень |

*Описание конфигурационной структуры дополнительных пинов*

## LCD\_Conf\_type

|  |  |
| --- | --- |
| Поле структуры | Описание |
| ulRgbMode | Формат цвета:  HAL\_LCD\_RGB888, HAL\_LCD\_RGB565, HAL\_LCD\_RGB444\_AR,  HAL\_LCD\_RGB444\_AL,  HAL\_LCD\_RGB233 |
| ulPixelClock | Частота синхроимпульса для LCD панели, кГц |
| ulPwmDiv | Делитель ШИМ (0 … 15) |
| bVSyncActiveLevel | Активный уровень VSYNC |
| bHSyncActiveLevel | Активный уровень HSYNC |
| bDReadyActiveLevel | Активный уровень линии готовности данных |
| bPClkInverse | Инверсия пиксель клока |
| bStopPclkWhenNoData | Останов пиксельклока, когда нет данных |
| bSwapRgbToBgr | 0 – RGB, 1 – BGR |
| usHSyncPos | Сдвиг HSYNC |
| usHSyncLen | Длительность HSYNC |
| usHSize | Видимые пиксели по горизонтали |
| usHFrontBlank | Количество пустых пикселей слева |
| usHBackBlank | Количество пустых пикселей справа |
| usVSyncPos | Сдвиг VSYNC |
| usVSyncLen | Длительность VSYNC |
| usVSize | Видимые линии по вертикали |
| usVFrontBlank | Количество пустых линий сверху |
| usVBackBlank | Количество пустых линий снизу |
| pxWindow | Указатель на структуру окна |

*Описание конфигурационной структуры LCD контроллера панели*

*Описание формата цвета*

|  |  |  |  |
| --- | --- | --- | --- |
|  | R | G | B |
| HAL\_LCD\_RGB888 | data[23:16] | data[15:8] | data[7:0] |
| HAL\_LCD\_RGB565 | data[15:11] | data[10:5] | data[4:0] |
| HAL\_LCD\_RGB444\_AR | data[13:10] | data[8:5] | data[3:0] |
| HAL\_LCD\_RGB444\_AL | data[14:11] | data[9:6] | data[4:1] |
| HAL\_LCD\_RGB233 | data[7:6] | data[5:3] | data[2:0] |

## LCD\_WindowConf\_type

|  |  |
| --- | --- |
| Поле структуры | Описание |
| usHSize | Размер окна по горизонтали |
| usVSize | Размер окна по вертикали |
| sHOffset | Смещение относительно usHSize |
| sVOffset | Смещение относительно usVSize |
| ulBackgndColor | Цвет фона неактивной области |

*Описание структуры окна*

# Описание используемых функций

## HAL\_LCD\_Enable

**Резюме**

**void** **HAL\_LCD\_Enable**( void )

**Описание**

Функция включения контроллера LCD.

## HAL\_LCD\_Disable

**Резюме**

**void** **HAL\_LCD\_Disable**( void )

**Описание**

Функция выключения контроллера LCD.

## HAL\_LCD\_GpioInit

**Резюме**

**void** **HAL\_LCD\_GpioInit**( void )

**Описание**

Функция конфигурации выводов используемых контроллером LCD панели, кроме дополнительных выводов GPIO\_0 .. GPIO\_3.

## HAL\_LCD\_GpioDeInit

**Резюме**

**void** **HAL\_LCD\_GpioDeInit**( void )

**Описание**

Функция сброса конфигурации выводов используемых контроллером LCD панели, кроме дополнительных выводов GPIO\_0 .. GPIO\_3.

## HAL\_LCD\_ExGpioInit

**Резюме**

**void** **HAL\_LCD\_ExGpioInit**( LCD\_ExGpioNum\_type eGpio, LCD\_ExGpio\_type \*pxExGpioCfg )

LCD\_ExGpioNum\_type eGpio –дополнительный вывод (*LCD\_Gpio\_0, LCD\_Gpio\_1, LCD\_Gpio\_2, LCD\_Gpio\_3*)

LCD\_ExGpio\_type \*pxExGpioCfg – указатель на конфигурацию дополнительно вывода

**Описание**

Функция конфигурирует один из дополнительных четырёх выводов контроллера LCD панели

## HAL\_LCD\_ExGpioDeInit

**Резюме**

**void** **HAL\_LCD\_ExGpioDeInit**( LCD\_ExGpioNum\_type eGpio )

LCD\_ExGpioNum\_type eGpio – дополнительный вывод (*LCD\_Gpio\_0, LCD\_Gpio\_1, LCD\_Gpio\_2, LCD\_Gpio\_3*);

**Описание**

Функция сбрасывает конфигурацию одного из дополнительных четырёх выводов контроллера LCD панели.

## HAL\_LCD\_Setup

**Резюме**

**void** **HAL\_LCD\_Setup**( LCD\_Conf\_type \*pxCfg )

LCD\_Conf\_type \*pxCfg – указатель на конфигурационную структуру

**Описание**

Функция конфигурирует контроллер LCD параметрами из структуры по указателю pxCfg***.***

Частота ШИМ подсветки LCD панели определяется следующей формулой:

## HAL\_LCD\_PwmSetDuty

**Резюме**

**void HAL\_LCD\_PwmSetDuty**( uint8\_t ucValue )

uint8\_t ucValue – значение коэффициента заполнения ШИМ подсветки

**Описание**

Функция задаёт заполнение ШИМ подсветки LCD панели. Минимальное значение 0, максимальное равно HAL\_LCD\_MAX\_PWM\_DUTY. В файле hal\_lcd.h определен макрос

**#define** HAL\_LCD\_MAX\_PWM\_DUTY 100

Данный макрос задает максимальное значение, при котором коэффициент заполнения ШИМ равен единице. Максимальное значение данного макроса 255.

## HAL\_LCD\_StartDma

**Резюме**

**void HAL\_LCD\_StartDma**( uint32\_t channel, **void** \*pvLcdBuf, uint32\_t ulXSize, uint32\_t ulYSize, uint32\_t ulRgbMode, **void** \*pvDmaIsrHandler )

uint32\_t channel – номер канала DMA, от 4-го до 7-го;

**void** \*pvLcdBuf – указатель на видеобуфер, который должен быть выровнен на границу квадрослова;

uint32\_t ulXSize – размер изображения по горизонтали;

uint32\_t ulYSize – размер изображения по вертикали;

uint32\_t ulRgbMode – формат цвета (HAL\_LCD\_RGB888 … HAL\_LCD\_RGB233);

**void** \*pvDmaIsrHandler – указатель на функцию обработчик прерывания, если прерывание не нужно, то передаем нулевой указатель.

**Описание**

Функция выводит изображение из заданного видеобуфера на LCD панель с использованием DMA

## HAL\_LCD\_StartDma2Q

**Резюме**

**void** **HAL\_LCD\_StartDma2Q** ( uint32\_t ulChannel, **void** \*pvLcdBuf, uint32\_t ulXSize, uint32\_t ulYSize, uint32\_t ulRgbMode, **void** \*pvDmaIsrHandler )

Аргументы аналогичны аргументам функции **HAL\_LCD\_StartDma.**

**Описание**

Функция выводит изображение из заданного видеобуфера на LCD панель с использованием режима DMA 2Q – загрузка сразу двух квадрослов.

# Пример работы

Для работы примера необходимо в ldf файле добавить после секции *M10Heap* секцию

sdr\_data

{

INPUT\_SECTIONS( $OBJECTS(.sdr\_data\*) )

} >MSSD0

Настройки приведены для LCD панели из отладочного комплекта.

**#include** <hal\_1967VN044.h>

**#include** <stdint.h>

**#define** N 40

**#define** Nlcd 65280

**#define** BUSCLK\_KHZ 80000

**#define** CORECLK\_KHZ 200000

**int** **main**()

{

uint32\_t src[N] = {1, 0,};

uint32\_t dst[N] = {0,};

uint32\_t **\_\_attribute\_\_**((aligned(4 \* (32/\_\_CHAR\_BIT\_\_)))) ulLcdBuf[Nlcd] = {1, 0};

LCD\_Conf\_type xLCD\_HX8257\_conf =

{

HAL\_LCD\_RGB565, // Режим цвета

8000, // Частота синхроимпульса для LCD панели, кГц

10, // Частота PWM

0, // Активный уровень VSYNC (0 - низкий, 1 - высокий)

0, // Активный уровень HSYNC (0 - низкий, 1 - высокий)

1, // Активный уровень линии готовности данных

0, // Инверсия пиксель клока

1, // Останов клока, если нет данных

0, // переключение из RGB в BGR

//>> Величины в периодах пиксель клока

1, // Сдвиг HSYNC

10, // Длительность HSYNC

480, // Видимые пиксели по горизонтали

41, // Количество пустых пикселей слева

7, // Количество пустых пикселей справа

//>> Величины в линиях

1, // Сдвиг VSYNC

10, // Длительность VSYNC

272, // Видимые линии по вертикали

11, // Количество пустых строк сверху

2, // Количество пустых строк снизу

0,

};

**int** i = 0;

uint32\_t color = 0xFC0;

**volatile** **int** flag = 0;

HAL\_PLL\_CoreSetup( CORECLK\_KHZ );

**for**( i = 0; i < Nlcd; i++ )

ulLcdBuf[ i ] = color;

LX\_CMU->CFG8.b.LCD\_DIS = 0; // Подаем тактирование на LCD

LX\_CMU->CFG8.b.VCAM\_DIS = 0;

HAL\_LCD\_Disable();

HAL\_LCD\_GpioInit();

HAL\_LCD\_StartDma( 4, ulLcdBuf, xLCD\_HX8257\_conf.usHSize, xLCD\_HX8257\_conf.usVSize, xLCD\_HX8257\_conf.ulRgbMode , 0 );

HAL\_LCD\_Setup( &xLCD\_HX8257\_conf );

HAL\_LCD\_PwmSetDuty( 50 );

HAL\_LCD\_Enable();

**return** 0;

}