# Описание используемых функций

## HAL\_Timer\_Init

**Резюме**

**void** **HAL\_Timer\_Init**( Timer\_X\_type Timer, uint64\_t Count, Timer\_State\_type State )

Timer\_X\_type Timer ­– номер таймера

uint64\_t Count – начальное значение

Timer\_State\_type State ­– состояние

**Описание**

Функция позволяет настроить период и включить один из таймеров общего назначения. Выбор таймера задается входным аргументом Timer\_X\_type Timer, который может принимать следующие значения:

**typedef** **enum** {

*Timer\_0* - таймер 0,

*Timer\_1* - таймер 1

} Timer\_X\_type;

Период счета таймера задается входным аргументом uint64\_t Count, диапазон его значений [0; 264 – 1].

Включение или выключение таймера задается входным аргументом Timer\_State\_type State, который может принимать следующие значения:

**typedef** **enum** {

*Timer\_State\_Disable* - таймер выключен,

*Timer\_State\_Enable* - таймер включен

} Timer\_State\_type;

**Пример**

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

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

// Включение таймера 0 с периодом 1000:

HAL\_Timer\_Init (*Timer\_0*, 1000, *Timer\_State\_Enable*);

// Включение таймера 1 с периодом 1234:

HAL\_Timer\_Init (*Timer\_1*, 1234, *Timer\_State\_Enable*);

**while** (1) {

}

**return** 0;

}

## HAL\_Timer\_DeInit

**Резюме**

**void** **HAL\_Timer\_DeInit**( Timer\_X\_type Timer )

Timer\_X\_type Timer ­– номер таймера

**Описание**

Функция сбрасывает период счета таймера и выключает его. Выбор таймера задается входным аргументом Timer\_X\_type Timer, допустимые значения которого описаны в функции **HAL\_Timer\_Init(…)**.

**Пример**

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

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

// Включение таймера 0 с периодом 1000:

HAL\_Timer\_Init (*Timer\_0*, 1000, *Timer\_State\_Enable*);

// Сброс таймера:

HAL\_Timer\_DeInit (*Timer\_0*);

// Период - 0, таймер выключен

**while** (1) {

}

**return** 0;

}

## HAL\_Timer\_GetCurrent

**Резюме**

uint64\_t **HAL\_Timer\_GetCurrent**( Timer\_X\_type Timer )

Timer\_X\_type Timer ­– номер таймера

**Описание**

Функция возвращает текущее значение таймера из регистров TIMER*x*L и TIMER*x*H, где *x* = 0; 1.

Выбор таймера задается входным аргументом Timer\_X\_type Timer, допустимые значения которого описаны в функции **HAL\_Timer\_Init(…)**.

Допустимый диапазон возвращаемого значения: [0; 264 – 1].

**Пример**

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

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

int T1 = -1;

// Включение таймера 1 с периодом T1:

HAL\_Timer\_Init (*Timer\_1*, T1, *Timer\_State\_Enable*);

**while** (1) {

T1 = HAL\_Timer\_GetCurrent (*Timer\_1*);

}

**return** 0;

}

## HAL\_Timer\_SetCount

**Резюме**

**void** **HAL\_Timer\_SetCount**( Timer\_X\_type Timer, uint64\_t Count )

Timer\_X\_type Timer ­– номер таймера

uint64\_t Count – значение

**Описание**

Период записывается в регистры TMRIN*x*L и TMRIN*x*H, где *x* = 0; 1 и применяет при достижении таймером 0.

Выбор таймера задается входным аргументом Timer\_X\_type Timer, допустимые значения которого описаны в функции **HAL\_Timer\_Init(…)**.

Период таймера задается входным аргументом uint64\_t Count, допустимый диапазон значений которого описан в функции **HAL\_Timer\_Init(…)**.

**Пример**

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

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

HAL\_Timer\_Init (*Timer\_0*, 1000, *Timer\_State\_Enable*);

// Счет таймера начался с 1000

HAL\_Timer\_SetCount (*Timer\_0*, 123);

// При достижении 0 значение таймера установится в 123

**while** (1) {

}

**return** 0;

}

## HAL\_Timer\_SetCountNow

**Резюме**

**void** **HAL\_Timer\_SetCountNow**( Timer\_X\_type Timer, uint64\_t Count )

Timer\_X\_type Timer ­– номер таймера

uint64\_t Count – значение

**Описание**

Функция устанавливает период таймера в регистры TMRIN*x*L и TMRIN*x*H, где *x* = 0; 1 и сбрасывает его текущее значение на установленный период.

Выбор таймера задается входным аргументом Timer\_X\_type Timer, допустимые значения которого описаны в функции **HAL\_Timer\_Init(…)**.

Период таймера задается входным аргументом uint64\_t Count, допустимый диапазон значений которого описан в функции **HAL\_Timer\_Init(…)**.

**Пример**

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

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

HAL\_Timer\_Init (*Timer\_0*, 1000, *Timer\_State\_Enable*);

// Счет таймера начался с 1000

HAL\_Timer\_SetCountNow (*Timer\_0*, 123);

// Счет таймер начал с 123

**while** (1) {

}

**return** 0;

}

## HAL\_Timer\_GetCount

**Резюме**

uint64\_t **HAL\_Timer\_GetCount**( Timer\_X\_type Timer )

Timer\_X\_type Timer ­– номер таймера

**Описание**

Функция возвращает значение периода таймера из регистров TMRIN*x*L и TMRIN*x*H, где *x* = 0; 1.

Выбор таймера задается входным аргументом Timer\_X\_type Timer, допустимые значения которого описаны в функции **HAL\_Timer\_Init(…)**.

Допустимый диапазон возвращаемого значения: [0; 264 – 1].

**Пример**

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

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

int Count = 0;

HAL\_Timer\_Init (*Timer\_0*, -1, *Timer\_State\_Enable*);

Count = HAL\_Timer\_GetCount (*Timer\_0*);

// Count = 0xFFFFFFFFFFFFFFFF

**while** (1) {

}

**return** 0;

}

## HAL\_Timer\_SetState

**Резюме**

**void** **HAL\_Timer\_SetState**( Timer\_X\_type Timer, Timer\_State\_type   
State )

Timer\_X\_type Timer ­– номер таймера

Timer\_State\_type State – состояние

**Описание**

Функция позволяет запускать и останавливать таймер.

Выбор таймера задается входным аргументом Timer\_X\_type Timer, допустимые значения которого описаны в функции **HAL\_Timer\_Init(…)**.

Запуск или остановка таймера определяется входным аргументом Timer\_State\_type State, допустимые значения которого описаны в функции **HAL\_Timer\_Init(…)**.

**Пример**

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

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

HAL\_Timer\_Init (*Timer\_0*, 4321, *Timer\_State\_Disable*);

// таймер остановлен

HAL\_Timer\_SetState (*Timer\_0*, *Timer\_State\_Enable*);

// таймер запущен

**while** (1) {

}

**return** 0;

}

## HAL\_Timer\_GetState

**Резюме**

Timer\_State\_type **HAL\_Timer\_GetState** (Timer\_X\_type Timer)

Timer\_X\_type Timer ­– номер таймера

**Описание**

Функция возвращает текущее значение бита TMR*x*RN регистра INTCTL, где *x* = 0; 1.

Выбор таймера задается входным аргументом Timer\_X\_type Timer, допустимые значения которого описаны в функции **HAL\_Timer\_Init(…)**.

Диапазон допустимых возвращаемых значений аналогичен значениям аргумента Timer\_State\_type State для функции **HAL\_Timer\_Init(…)**.

**Пример**

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

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

Timer\_State\_type S;

HAL\_Timer\_Init (*Timer\_0*, 4321, *Timer\_State\_Disable*);

S = HAL\_Timer\_GetState (*Timer\_0*); // S = Timer\_State\_Disable

HAL\_Timer\_SetState (*Timer\_0*, *Timer\_State\_Enable*);

S = HAL\_Timer\_GetState (*Timer\_0*); // S = Timer\_State\_Enable

**while** (1) {

}

**return** 0;

}

## HAL\_Timers\_SetState

**Резюме**

**void** **HAL\_Timers\_SetState** (Timer\_State\_type Timer0, Timer\_State\_type Timer1)

Timer\_State\_type Timer0 – состояние таймера 0

Timer\_State\_type Timer1 – состояние таймера 1

**Описание**

Функция одновременно записывает значения состояний для обоих таймеров в биты TMR0RN и TMR1RN регистра INTCTL.

Состояние таймера 0 задается с помощью входного аргумента Timer\_State\_type Timer0, допустимые значения которого аналогичны значениям аргумента Timer\_State\_type State для функции **HAL\_Timer\_Init(…)**.

Состояние таймера 1 задается с помощью входного аргумента Timer\_State\_type Timer1, допустимые значения которого аналогичны значениям аргумента Timer\_State\_type State для функции **HAL\_Timer\_Init(…)**.

**Пример**

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

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

HAL\_Timer\_SetCount (*Timer\_0*, 1234);

HAL\_Timer\_SetCount (*Timer\_0*, 5678);

HAL\_Timers\_SetState (*Timer\_State\_Disable*, *Timer\_State\_Enable*);

// таймер 0 - остановлен, таймер 1 - запущен

HAL\_Timers\_SetState (*Timer\_State\_Enable*, *Timer\_State\_Disable*);

// таймер 0 - запущен, таймер 1 - остановлен

HAL\_Timers\_SetState (*Timer\_State\_Enable*, *Timer\_State\_Enable*);

// таймер 0 - запущен, таймер 1 - запущен

HAL\_Timers\_SetState (*Timer\_State\_Disable*, *Timer\_State\_Disable*);

// таймер 0 - остановлен, таймер 1 - остановлен

**while** (1) {

}

**return** 0;

}

# Прерывания

Для работы с прерываниями от таймеров общего назначения никаких действий с самими таймерами не требуется. Для включения или отключения таймеров достаточно использовать только **HAL Interrupt**.

## Пример

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

int A = 0;

**\_\_attribute**((interrupt))

**void** **T0\_Handler**();

**\_\_attribute**((interrupt))

**void** **T1\_Handler**();

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

// Включаем прерывания:

HAL\_Interrupt\_IMASKClear();

HAL\_Interrupt\_ILATClear();

HAL\_Interrupt\_GlobalEnable();

HAL\_Interrupt\_Enable (*intTMR0HP*, &T0\_Handler);

HAL\_Interrupt\_Enable (*intTMR1HP*, &T1\_Handler);

// Инициализируем таймеры:

HAL\_Timer\_Init (*Timer\_0*, 1000, *Timer\_State\_Enable*);

HAL\_Timer\_Init (*Timer\_1*, 2000, *Timer\_State\_Enable*);

**while** (1) {

}

**return** 0;

}

// Прерывание от таймера 0:

**void** **T0\_Handler**() {

A++;

}

// Прерывание от таймера 1:

**void** **T1\_Handler**() {

A--;

}