# HAL\_GPTimer\_Init

## Резюме

**void** **HAL\_GPTimer\_Init** (GPTimer\_X Timer, u64 Count, GPTimer\_State State) – функция инициализации таймера общего назначения GPTimer\_X Timer начальным значением u64 Count и состоянием GPTimer\_State State.

## Описание

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

**typedef** **enum** {

*GPTimer\_0* - таймер 0,

*GPTimer\_1* - таймер 1

} GPTimer\_X;

Период счета таймера задается входным аргументом u64 Count, диапазон его значений [0; 264 – 1].

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

**typedef** **enum** {

*GPTimer\_State\_Disable* - таймер выключен,

*GPTimer\_State\_Enable* - таймер включен

} GPTimer\_State;

## Пример

**#include** <hal\_1967VN034R1.h>

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

// Включение таймера 0 с периодом 1000:

HAL\_GPTimer\_Init (*GPTimer\_0*, 1000, *GPTimer\_State\_Enable*);

// Включение таймера 1 с периодом 1234:

HAL\_GPTimer\_Init (*GPTimer\_1*, 1234, *GPTimer\_State\_Enable*);

**while** (1) {

}

**return** 0;

}

# HAL\_GPTimer\_DeInit

## Резюме

**void** **HAL\_GPTimer\_DeInit** (GPTimer\_X Timer) – функция сброса настроек таймера GPTimer\_X Timer.

## Описание

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

## Пример

**#include** <hal\_1967VN034R1.h>

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

// Включение таймера 0 с периодом 1000:

HAL\_GPTimer\_Init (*GPTimer\_0*, 1000, *GPTimer\_State\_Enable*);

// Сброс таймера:

HAL\_GPTimer\_DeInit (*GPTimer\_0*);

// Период - 0, таймер выключен

**while** (1) {

}

**return** 0;

}

# HAL\_GPTimer\_GetCurrent

## Резюме

u64 **HAL\_GPTimer\_GetCurrent** (GPTimer\_X Timer) – функция возвращает текущее значение таймера GPTimer\_X Timer.

## Описание

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

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

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

## Пример

**#include** <hal\_1967VN034R1.h>

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

u64 T1 = -1;

// Включение таймера 1 с периодом T1:

HAL\_GPTimer\_Init (*GPTimer\_1*, T1, *GPTimer\_State\_Enable*);

**while** (1) {

T1 = HAL\_GPTimer\_GetCurrent (*GPTimer\_1*);

}

**return** 0;

}

# HAL\_GPTimer\_SetCount

## Резюме

**void** **HAL\_GPTimer\_SetCount** (GPTimer\_X Timer, u64 Count) – функция устанавливает период u64 Count таймера GPTimer\_X Timer.

## 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 GPTimer\_X Timer, допустимые значения которого описаны в функции **HAL\_GPTimer\_Init(…)**.

Период таймера задается входным аргументом u64 Count, допустимый диапазон значений которого описан в функции **HAL\_GPTimer\_Init(…)**.

## Пример

**#include** <hal\_1967VN034R1.h>

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

HAL\_GPTimer\_Init (*GPTimer\_0*, 1000, *GPTimer\_State\_Enable*);

// Счет таймера начался с 1000

HAL\_GPTimer\_SetCount (*GPTimer\_0*, 123);

// При достижении 0 значение таймера установится в 123

**while** (1) {

}

**return** 0;

}

# HAL\_GPTimer\_SetCountNow

## Резюме

**void** **HAL\_GPTimer\_SetCountNow** (GPTimer\_X Timer, u64 Count) – функция устанавливает период u64 Count таймера GPTimer\_X Timer не дожидаясь достижения 0.

## 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 GPTimer\_X Timer, допустимые значения которого описаны в функции **HAL\_GPTimer\_Init(…)**.

Период таймера задается входным аргументом u64 Count, допустимый диапазон значений которого описан в функции **HAL\_GPTimer\_Init(…)**.

## Пример

**#include** <hal\_1967VN034R1.h>

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

HAL\_GPTimer\_Init (*GPTimer\_0*, 1000, *GPTimer\_State\_Enable*);

// Счет таймера начался с 1000

HAL\_GPTimer\_SetCountNow (*GPTimer\_0*, 123);

// Счет таймер начал с 123

**while** (1) {

}

**return** 0;

}

# HAL\_GPTimer\_GetCount

## Резюме

u64 **HAL\_GPTimer\_GetCount** (GPTimer\_X Timer) – функция возвращает период таймера GPTimer\_X Timer.

## Описание

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

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

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

## Пример

**#include** <hal\_1967VN034R1.h>

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

u64 Count = 0;

HAL\_GPTimer\_Init (*GPTimer\_0*, -1, *GPTimer\_State\_Enable*);

Count = HAL\_GPTimer\_GetCount (*GPTimer\_0*);

// Count = 0xFFFFFFFFFFFFFFFF

**while** (1) {

}

**return** 0;

}

# HAL\_GPTimer\_SetState

## Резюме

**void** **HAL\_GPTimer\_SetState** (GPTimer\_X Timer, GPTimer\_State State) – функция изменяет состояние GPTimer\_State State (запущен/остановлен) таймера GPTimer\_X Timer.

## Описание

Функция позволяет запускать и останавливать таймер.

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

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

## Пример

**#include** <hal\_1967VN034R1.h>

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

HAL\_GPTimer\_Init (*GPTimer\_0*, 4321, *GPTimer\_State\_Disable*);

// таймер остановлен

HAL\_GPTimer\_SetState (*GPTimer\_0*, *GPTimer\_State\_Enable*);

// таймер запущен

**while** (1) {

}

**return** 0;

}

# HAL\_GPTimer\_GetState

## Резюме

GPTimer\_State **HAL\_GPTimer\_GetState** (GPTimer\_X Timer) – функция возвращает текущее состояние таймера GPTimer\_X Timer (запущен/остановлен).

## Описание

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

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

Диапазон допустимых возвращаемых значений аналогичен значениям аргумента GPTimer\_State State для функции **HAL\_GPTimer\_Init(…)**.

## Пример

**#include** <hal\_1967VN034R1.h>

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

GPTimer\_State S;

HAL\_GPTimer\_Init (*GPTimer\_0*, 4321, *GPTimer\_State\_Disable*);

S = HAL\_GPTimer\_GetState (*GPTimer\_0*); // S = GPTimer\_State\_Disable

HAL\_GPTimer\_SetState (*GPTimer\_0*, *GPTimer\_State\_Enable*);

S = HAL\_GPTimer\_GetState (*GPTimer\_0*); // S = GPTimer\_State\_Enable

**while** (1) {

}

**return** 0;

}

# HAL\_GPTimers\_SetState

## Резюме

**void** **HAL\_GPTimers\_SetState** (GPTimer\_State Timer0, GPTimer\_State Timer1) – функция позволяет одновременно устанавливать состояние GPTimer\_State Timer0 (запущен/остановлен) для таймера 0 и состояние GPTimer\_State Timer1 для 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 GPTimer\_State Timer0, допустимые значения которого аналогичны значениям аргумента GPTimer\_State State для функции **HAL\_GPTimer\_Init(…)**.

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

## Пример

**#include** <hal\_1967VN034R1.h>

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

HAL\_GPTimer\_SetCount (*GPTimer\_0*, 1234);

HAL\_GPTimer\_SetCount (*GPTimer\_0*, 5678);

HAL\_GPTimers\_SetState (*GPTimer\_State\_Disable*, *GPTimer\_State\_Enable*);

// таймер 0 - остановлен, таймер 1 - запущен

HAL\_GPTimers\_SetState (*GPTimer\_State\_Enable*, *GPTimer\_State\_Disable*);

// таймер 0 - запущен, таймер 1 - остановлен

HAL\_GPTimers\_SetState (*GPTimer\_State\_Enable*, *GPTimer\_State\_Enable*);

// таймер 0 - запущен, таймер 1 - запущен

HAL\_GPTimers\_SetState (*GPTimer\_State\_Disable*, *GPTimer\_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\_1967VN034R1.h>

uint A = 0;

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

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

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

// Включаем прерывания:

HAL\_Interrupt\_IMASKClear();

HAL\_Interrupt\_ILATClear();

HAL\_Interrupt\_GlobalEnable();

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

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

// Инициализируем таймеры:

HAL\_GPTimer\_Init (*GPTimer\_0*, 1000, *GPTimer\_State\_Enable*);

HAL\_GPTimer\_Init (*GPTimer\_1*, 2000, *GPTimer\_State\_Enable*);

**while** (1) {

}

**return** 0;

}

// Прерывания таймера 0:

**#pragma** interrupt

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

A++;

}

// Прерывание таймера 1:

**#pragma** interrupt

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

A--;

}