# Описание используемых типов

## GTIMER\_CntInit\_type

|  |  |
| --- | --- |
| **Имя поля** | **Описание** |
| IniCounter | Начальное значение счетчика |
| Prescaler | Делитель тактовой частоты |
| Period | Основание счета (период) |
| CounterMode | Режим счета:  *GTIMER\_CntMode\_ClkFixedDir* – счет без изменения направления от внутреннего тактового сигнала  *GTIMER\_CntMode\_ClkChangeDir* – счет с изменением направления от внутреннего тактового сигнала  *GTIMER\_CntMode\_EvtFixedDir* – счет без изменения направления от событий  *GTIMER\_CntMode\_EvtChangeDir* – счет с изменением направления от событий |
| CounterDirection | Направление счета:  *GTIMER\_CntDir\_Up* – счет вверх  *GTIMER\_CntDir\_Down –* счет вниз |
| EventSource | Источник событий:  *GTIMER\_EvSrc\_None* ­– нет событий  *GTIMER\_EvSrc\_TM1* – CNT == ARR в таймере 1  *GTIMER\_EvSrc\_TM2* – CNT == ARR в таймере 2  *GTIMER\_EvSrc\_CH1* – событие на первом канале  *GTIMER\_EvSrc\_CH2* – событие на втором канале  *GTIMER\_EvSrc\_CH3* – событие на третьем канале  *GTIMER\_EvSrc\_CH4* – событие на четвертом канале  *GTIMER\_EvSrc\_ETR\_RE* – событие переднего фронта ETR  *GTIMER\_EvSrc\_ETR\_FE* – событие заднего фронта ETR |
| FilterSampling | Частота семплирования данных:  *GTIMER\_FdtsDiv\_1* – частота семплирования каждый TIM\_CLK  *GTIMER\_FdtsDiv\_2* – частота семплирования каждый второй TIM\_CLK  *GTIMER\_FdtsDiv\_3* – частота семплирования каждый третий TIM\_CLK  *GTIMER\_FdtsDiv\_4* – частота семплирования каждый четвертый TIM\_CLK |
| ARR\_UpdateMode | Режим обновления регистра ARR:  *GTIMER\_Update\_Immediately* – обновление возможно в любой момент времени  *GTIMER\_Update\_On\_CNT\_Overflow* – обновление будет осуществлено только при CNT = 0 |
| ETR\_FilterConf | Настройка фильтра на входе ETR:  *GTIMER\_Filter\_1FF\_at\_GTIMER\_CLK* – сигнал зафиксирован в 1 триггере на частоте TIM\_CLK  *GTIMER\_Filter\_2FF\_at\_GTIMER\_CLK* – сигнал зафиксирован в 2 триггерах на частоте TIM\_CLK  *GTIMER\_Filter\_4FF\_at\_GTIMER\_CLK* – сигнал зафиксирован в 4 триггерах на частоте TIM\_CLK  *GTIMER\_Filter\_8FF\_at\_GTIMER\_CLK* – сигнал зафиксирован в 8 триггерах на частоте TIM\_CLK  *GTIMER\_Filter\_6FF\_at\_FDTS\_div\_2* – сигнал зафиксирован в 6 триггерах на частоте FDTS/2  *GTIMER\_Filter\_8FF\_at\_FDTS\_div\_2* – сигнал зафиксирован в 8 триггерах на частоте FDTS/2  *GTIMER\_Filter\_6FF\_at\_FDTS\_div\_4* – сигнал зафиксирован в 6 триггерах на частоте FDTS/4  *GTIMER\_Filter\_8FF\_at\_FDTS\_div\_4* – сигнал зафиксирован в 8 триггерах на частоте FDTS/4  *GTIMER\_Filter\_6FF\_at\_FDTS\_div\_8* – сигнал зафиксирован в 6 триггерах на частоте FDTS/8  *GTIMER\_Filter\_8FF\_at\_FDTS\_div\_8* –сигнал зафиксирован в 8 триггерах на частоте FDTS/8  *GTIMER\_Filter\_5FF\_at\_FDTS\_div\_16* – сигнал зафиксирован в 5 триггерах на частоте FDTS/16  *GTIMER\_Filter\_6FF\_at\_FDTS\_div\_16* – сигнал зафиксирован в 6 триггерах на частоте FDTS/16  *GTIMER\_Filter\_8FF\_at\_FDTS\_div\_16* – сигнал зафиксирован в 8 триггерах на частоте FDTS/16  *GTIMER\_Filter\_5FF\_at\_FDTS\_div\_32* – сигнал зафиксирован в 5 триггерах на частоте FDTS/32  *GTIMER\_Filter\_6FF\_at\_FDTS\_div\_32* – сигнал зафиксирован в 6 триггерах на частоте FDTS/32  *GTIMER\_Filter\_8FF\_at\_FDTS\_div\_32* – сигнал зафиксирован в 8 триггерах на частоте FDTS/32 |
| ETR\_Prescaler | Асинхронный пред делитель внешней частоты:  *GTIMER\_Prescaler\_None* – нет деления  *GTIMER\_Prescaler\_2* – /2  *GTIMER\_Prescaler\_4* – /4  *GTIMER\_Prescaler\_8* – /8 |
| ETR\_Polarity | Инверсия ETR:  *GTIMER\_Polarity\_NonInverted* – нет инверсии входа  *GTIMER\_Polarity\_Inverted* – инверсия входа |
| BRK\_Polarity | Инверсия BRK:  *GTIMER\_Polarity\_NonInverted* – нет инверсии входа  *GTIMER\_Polarity\_Inverted* – инверсия входа |

## GTIMER\_ChnInit\_type

|  |  |
| --- | --- |
| **Имя поля** | **Описание** |
| ChNumber | Номер канала таймера:  *GTIMER\_CH1*  *GTIMER\_CH2*  *GTIMER\_CH3*  *GTIMER\_CH4* |
| Mode | Режим работы канала:  *GTIMER\_CH\_MODE\_PWM* – канал работает в режиме ШИМ  *GTIMER\_CH\_MODE\_CAPTURE* – канал работает в режиме Захват |
| ETR\_Ena | Разрешение работы ETR:  *DISABLE*  *ENABLE* |
| ETR\_Reset | Разрешение сброса по выводу ETR:  *GTIMER\_CH\_RESET\_DISABLE* – сброс по внешнему выводу запрещен  *GTIMER\_CH\_RESET\_ENABLE* – сброс по внешнему выводу разрешен |
| BRK\_Reset | Разрешение сброса по выводу BRK:  *GTIMER\_CH\_RESET\_DISABLE* – сброс по внешнему выводу запрещен  *GTIMER\_CH\_RESET\_ENABLE* – сброс по внешнему выводу разрешен |
| REF\_Format | Формат выработки сигнала REF в режиме ШИМ:  *GTIMER\_CH\_REF\_Format0* – REF = 0  *GTIMER\_CH\_REF\_Format1* – REF = 1 если: (CNT == CCR) или ((CNT == CCR1) и CCR1\_EN = 1), иначе REF = 0  *GTIMER\_CH\_REF\_Format2* – REF = 0 если: (CNT == CCR) или ((CNT == CCR1) и CCR1\_EN = 1), иначе REF = 1  *GTIMER\_CH\_REF\_Format3* – Переключение REF если: (CNT == CCR) или ((CNT == CCR1) и CCR1\_EN = 1)  *GTIMER\_CH\_REF\_Format4* – REF = 0  *GTIMER\_CH\_REF\_Format5* – REF = 1  *GTIMER\_CH\_REF\_Format6* – REF = ~DIR если: (CCR1\_EN = 0 и (CNT < CCR)) или (CCR1\_EN = 1 и (CCR1 < CNT < CCR)), иначе REF = DIR  *GTIMER\_CH\_REF\_Format7* – REF = DIR если: (CCR1\_EN = 0 и (CNT < CCR)) или (CCR1\_EN = 1 и (CCR1 < CNT < CCR)), иначе REF = ~DIR |
| Prescaler | Предварительный делитель входного канала:  *GTIMER\_Prescaler\_None* – нет деления  *GTIMER\_Prescaler\_2* – /2  *GTIMER\_Prescaler\_4* – /4  *GTIMER\_Prescaler\_8* – /8 |
| EventSource | Выбор события по входному каналу:  *GTIMER\_EventIn\_PE* – Событие входного канала положительный фронт  *GTIMER\_EventIn\_NE* – Событие входного канала отрицательный фронт  *GTIMER\_EventIn\_EX1* – Событие входного канала положительный фронт от других каналов ( 1<-2, 2<-3, 3<-4, 4<-1 )  *GTIMER\_EventIn\_EX2* – Событие входного канала положительный фронт от других каналов ( 1<-3, 2<-4, 3<-1, 4<-2 ) |
| FilterConf | Фильтрация канала:  *GTIMER\_Filter\_1FF\_at\_GTIMER\_CLK* – сигнал зафиксирован в 1 триггере на частоте TIM\_CLK  *GTIMER\_Filter\_2FF\_at\_GTIMER\_CLK* – сигнал зафиксирован в 2 триггерах на частоте TIM\_CLK  *GTIMER\_Filter\_4FF\_at\_GTIMER\_CLK* – сигнал зафиксирован в 4 триггерах на частоте TIM\_CLK  *GTIMER\_Filter\_8FF\_at\_GTIMER\_CLK* – сигнал зафиксирован в 8 триггерах на частоте TIM\_CLK  *GTIMER\_Filter\_6FF\_at\_FDTS\_div\_2* – сигнал зафиксирован в 6 триггерах на частоте FDTS/2  *GTIMER\_Filter\_8FF\_at\_FDTS\_div\_2* – сигнал зафиксирован в 8 триггерах на частоте FDTS/2  *GTIMER\_Filter\_6FF\_at\_FDTS\_div\_4* – сигнал зафиксирован в 6 триггерах на частоте FDTS/4  *GTIMER\_Filter\_8FF\_at\_FDTS\_div\_4* – сигнал зафиксирован в 8 триггерах на частоте FDTS/4  *GTIMER\_Filter\_6FF\_at\_FDTS\_div\_8* – сигнал зафиксирован в 6 триггерах на частоте FDTS/8  *GTIMER\_Filter\_8FF\_at\_FDTS\_div\_8* –сигнал зафиксирован в 8 триггерах на частоте FDTS/8  *GTIMER\_Filter\_5FF\_at\_FDTS\_div\_16* – сигнал зафиксирован в 5 триггерах на частоте FDTS/16  *GTIMER\_Filter\_6FF\_at\_FDTS\_div\_16* – сигнал зафиксирован в 6 триггерах на частоте FDTS/16  *GTIMER\_Filter\_8FF\_at\_FDTS\_div\_16* – сигнал зафиксирован в 8 триггерах на частоте FDTS/16  *GTIMER\_Filter\_5FF\_at\_FDTS\_div\_32* – сигнал зафиксирован в 5 триггерах на частоте FDTS/32  *GTIMER\_Filter\_6FF\_at\_FDTS\_div\_32* – сигнал зафиксирован в 6 триггерах на частоте FDTS/32  *GTIMER\_Filter\_8FF\_at\_FDTS\_div\_32* – сигнал зафиксирован в 8 триггерах на частоте FDTS/32 |
| CCR\_UpdateMode | Настройка обновления регистров CCR и CCR1:  *GTIMER\_Update\_Immediately* – обновление возможно в любой момент времени  *GTIMER\_Update\_On\_CNT\_Overflow* – обновление будет осуществлено только при CNT = 0 |
| CCR1\_Ena | Разрешение работы регистра CCR1:  *DISABLE*  *ENABLE* |
| CCR1\_EventSource | Выбор события по входному каналу для CAP1:  *GTIMER\_EventIn\_PE* – событие входного канала положительный фронт  *GTIMER\_EventIn\_NE* – событие входного канала отрицательный фронт  *GTIMER\_EventIn\_EX1* – событие входного канала положительный фронт от других каналов ( 1<-2, 2<-3, 3<-4, 4<-1 )  *GTIMER\_EventIn\_EX2* – событие входного канала положительный фронт от других каналов ( 1<-3, 2<-4, 3<-1, 4<-2 ) |

## GTIMER\_ChnOutInit\_type

|  |  |
| --- | --- |
| **Имя поля** | **Описание** |
| ChNumber | Номер канала таймера:  *GTIMER\_CH1*  *GTIMER\_CH2*  *GTIMER\_CH3*  *GTIMER\_CH4* |
| DirOut\_Polarity | Инверсия прямого канала:  *GTIMER\_CHOPolarity\_NonInverted* – выход не инвертируется  *GTIMER\_CHOPolarity\_Inverted* – выход не инвертируется |
| DirOut\_Source | Источник сигнала прямого канала:  *GTIMER\_CH\_OutSrc\_Only\_0* – На выход подается всегда 0  *GTIMER\_CH\_OutSrc\_Only\_1* – На выход подается всегда 1  *GTIMER\_CH\_OutSrc\_REF* – На выход выдается сигнал REF  *GTIMER\_CH\_OutSrc\_DTG* – На выход выдается сигнал с DTG |
| DirOut\_Mode | Режим работы прямого канала:  *GTIMER\_CH\_OutMode\_Input* – Канал работает на вход  *GTIMER\_CH\_OutMode\_Output* – Канал всегда работает на выход  *GTIMER\_CH\_OutMode\_REF\_as\_OE* – Режим канала зависит от сигнала REF  *GTIMER\_CH\_OutMode\_DTG\_as\_OE* – Режим канала зависит от сигнала DTG |
| NegOut\_Polarity | Инверсия инверсного канала:  *GTIMER\_CHOPolarity\_NonInverted* – выход не инвертируется  *GTIMER\_CHOPolarity\_Inverted* – выход не инвертируется |
| NegOut\_Source | Источник сигнала инверсного канала:  *GTIMER\_CH\_OutSrc\_Only\_0* – На выход подается всегда 0  *GTIMER\_CH\_OutSrc\_Only\_1* – На выход подается всегда 1  *GTIMER\_CH\_OutSrc\_REF* – На выход выдается сигнал REF  *GTIMER\_CH\_OutSrc\_DTG* – На выход выдается сигнал с DTG |
| NegOut\_Mode | Режим работы инверсного канала:  *GTIMER\_CH\_OutMode\_Input* – Канал работает на вход  *GTIMER\_CH\_OutMode\_Output* – Канал всегда работает на выход  *GTIMER\_CH\_OutMode\_REF\_as\_OE* – Режим канала зависит от сигнала REF  *GTIMER\_CH\_OutMode\_DTG\_as\_OE* – Режим канала зависит от сигнала DTG |
| DTG\_MainPrescaler | Основной делитель частоты генератора мертвого времени |
| DTG\_AuxPrescaler | Предварительный делитель частоты генератора мертвого времени |
| DTG\_ClockSource | Источник тактирования генератора мертвой времени:  *GTIMER\_CH\_DTG\_ClkSrc\_GTIMER\_CLK* – Частота TIM\_CLK  *GTIMER\_CH\_DTG\_ClkSrc\_FDTS* – Частота FDTS |

# Описание используемых функций

## HAL\_GTIMER\_DeInit

**Резюме**

**void HAL\_GTIMER\_DeInit**( GTIMER\_type\* GTIMERx )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

**Описание**

Функция сбрасывает периферию выбранного таймера – записывает нули в регистры.

## HAL\_GTIMER\_CntInit

**Резюме**

**void** **HAL\_GTIMER\_CntInit**( GTIMER\_type\* GTIMERx, **const** GTIMER\_CntInit\_type\* GTIMER\_CntInitStruct )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

**const** GTIMER\_CntInit\_type\* GTIMER\_CntInitStruct – указатель на структуру с параметрами инициализации

**Описание**

Функция инициализирует регистры отвечающие за режим счета таймера параметрами из инициализационной структуры.

## HAL\_GTIMER\_CntStructInit

**Резюме**

**void** **HAL\_GTIMER\_CntStructInit**( GTIMER\_CntInit\_type\*   
GTIMER\_CntInitStruct )

GTIMER\_CntInit\_type\* GTIMER\_CntInitStruct – указатель на структуру с параметрами инициализации

**Описание**

Функция задает параметры инициализационной структуры значениями по умолчанию:

GTIMER\_CntInitStruct->IniCounter = 0;

GTIMER\_CntInitStruct->Prescaler = 0;

GTIMER\_CntInitStruct->Period = 0;

GTIMER\_CntInitStruct->CounterMode = *GTIMER\_CntMode\_ClkFixedDir*;

GTIMER\_CntInitStruct->CounterDirection = *GTIMER\_CntDir\_Up*;

GTIMER\_CntInitStruct->EventSource = *GTIMER\_EvSrc\_None*;

GTIMER\_CntInitStruct->FilterSampling = *GTIMER\_FdtsDiv\_1*;

GTIMER\_CntInitStruct->ARR\_UpdateMode = *GTIMER\_Update\_Immediately*;

GTIMER\_CntInitStruct->ETR\_FilterConf = *GTIMER\_Filter\_1FF\_at\_GTIMER\_CLK*;

GTIMER\_CntInitStruct->ETR\_Prescaler = *GTIMER\_Prescaler\_None*;

GTIMER\_CntInitStruct->ETR\_Polarity = *GTIMER\_Polarity\_NonInverted*;

GTIMER\_CntInitStruct->BRK\_Polarity = *GTIMER\_Polarity\_NonInverted*;

## HAL\_GTIMER\_Cmd

**Резюме**

**void** **HAL\_GTIMER\_Cmd**( GTIMER\_type\* GTIMERx, FunctionalState\_type NewState )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

FunctionalState\_type NewState – состояние (*ENABLE, DISABLE*)

**Описание**

Функция включат или выключает выбранный таймер.

**HAL\_GTIMER\_SetCounter**

**Резюме**

**void** **HAL\_GTIMER\_SetCounter**( GTIMER\_type\* GTIMERx, uint32\_t Counter )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

uint32\_t Counter – значение счетчика

**Описание**

Функция задает значение счетчика (регистр CNT) выбранного таймера.

## HAL\_GTIMER\_SetCntPrescaler

**Резюме**

**void HAL\_GTIMER\_SetCntPrescaler**( GTIMER\_type\* GTIMERx, uint16\_t Prescaler )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

uint16\_t Prescaler – значение делителя частоты

**Описание**

Функция задает значение делителя частоты тактирования (регистр PSG) выбранного таймера.

## HAL\_GTIMER\_CntAutoreloadConfig

**Резюме**

**void** **HAL\_GTIMER\_CntAutoreloadConfig**( GTIMER\_type\* GTIMERx, uint32\_t Autoreload, GTIMER\_Update\_type UpdateMode )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

uint32\_t Autoreload – значение при котором происходит обнуление счетчика

GTIMER\_Update\_type UpdateMode – когда загружать новое значение (сразу – *GTIMER\_Update\_Immediately*, при CNT = 0 – *GTIMER\_Update\_On\_CNT\_Overflow*)

**Описание**

Функция задает значение регистра основания счета (регистр ARR) выбранного таймера и момент его обновления.

## HAL\_GTIMER\_GetCounter

**Резюме**

uint32\_t **HAL\_GTIMER\_GetCounter**( GTIMER\_type\* GTIMERx )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

**Описание**

Функция возвращает текущее значение счетного регистра CNT выбранного таймера.

## HAL\_GTIMER\_GetCounterDirection

**Резюме**

GTIMER\_CNTRL\_DIR\_type **HAL\_GTIMER\_GetCounterDirection**( GTIMER\_type\* GTIMERx )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

**Описание**

Функция возвращает текущее значение направления счета выбранного таймера:

*GTIMER\_CntDir\_Up* – счет вверх

*GTIMER\_CntDir\_Down* – счет вниз

## HAL\_GTIMER\_GetCntWriteComplete

**Резюме**

State\_type **HAL\_GTIMER\_GetCntWriteComplete**( GTIMER\_type\* GTIMERx )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

**Описание**

Функция возвращает текущее состояние записи регистров CNT, PSG и ARR выбранного таймера:

*RESET* = 0

*SET* = 1

## HAL\_GTIMER\_ETRInputConfig

**Резюме**

**void** **HAL\_GTIMER\_ETRInputConfig**( GTIMER\_type\* GTIMERx, GTIMER\_Prescaler\_type Prescaler, GTIMER\_Polarity\_type Polarity, GTIMER\_Filter\_type Filter )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_Prescaler\_type Prescaler – предделитель

GTIMER\_Polarity\_type Polarity – настройка инверсии

GTIMER\_Filter\_type Filter – цифровой фильтр

**Описание**

Функция настраивает параметры входа внешнего тактирования ETR выбранного таймера. Значения аргументов описаны в структуре GTIMER\_ChnInit\_type и соответствуют полям ETR\_Prescaler, ETR\_Polarity, ETR\_FilterConf.

## HAL\_GTIMER\_BRKPolarityConfig

**Резюме**

**void** **HAL\_GTIMER\_BRKPolarityConfig**( GTIMER\_type\* GTIMERx, GTIMER\_Polarity\_type Polarity )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_Polarity\_type Polarity – настройка инверсии

**Описание**

Функция настраивает параметры входа внешнего сигнала сброса BRK выбранного таймера. Значение аргумента описаны в структуре GTIMER\_ChnInit\_type и соответствуют полю BRK\_Polarity.

## HAL\_GTIMER\_ChnInit

**Резюме**

**void** **HAL\_GTIMER\_ChnInit**( GTIMER\_type\* GTIMERx, **const** GTIMER\_ChnInit\_type\* GTIMER\_ChnInitStruct )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

**const** GTIMER\_ChnInit\_type\* GTIMER\_ChnInitStruct – указатель на структуру с параметрами инициализации канала таймера

**Описание**

Функция инициализирует регистры отвечающие за режим работы канала таймера параметрами из инициализационной структуры.

## HAL\_GTIMER\_ChnStructInit

**Резюме**

**void** **HAL\_GTIMER\_ChnStructInit**( GTIMER\_ChnInit\_type\*   
GTIMER\_ChnInitStruct )

GTIMER\_ChnInit\_type\* GTIMER\_ChnInitStruct – указатель на структуру с параметрами инициализации канала таймера

**Описание**

Функция задает параметры инициализационной структуры канала таймера значениями по умолчанию:

GTIMER\_ChnInitStruct->ChNumber = *GTIMER\_CH1*;

GTIMER\_ChnInitStruct->Mode = *GTIMER\_CH\_MODE\_PWM*;

GTIMER\_ChnInitStruct->ETR\_Ena = *DISABLE*;

GTIMER\_ChnInitStruct->ETR\_Reset = *GTIMER\_CH\_RESET\_DISABLE*;

GTIMER\_ChnInitStruct->BRK\_Reset = *GTIMER\_CH\_RESET\_DISABLE*;

GTIMER\_ChnInitStruct->REF\_Format = *GTIMER\_CH\_REF\_Format0*;

GTIMER\_ChnInitStruct->Prescaler = *GTIMER\_Prescaler\_None*;

GTIMER\_ChnInitStruct->EventSource = *GTIMER\_EventIn\_PE*;

GTIMER\_ChnInitStruct->FilterConf = *GTIMER\_Filter\_1FF\_at\_GTIMER\_CLK*;

GTIMER\_ChnInitStruct->CCR\_UpdateMode = *GTIMER\_Update\_Immediately*;

GTIMER\_ChnInitStruct->CCR1\_Ena = *DISABLE*;

GTIMER\_ChnInitStruct->CCR1\_EventSource = *GTIMER\_EventIn\_PE*;

## HAL\_GTIMER\_ChnCompareSet

**Резюме**

**void** **HAL\_GTIMER\_ChnCompareSet**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, uint32\_t Compare )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

uint32\_t Compare – значение регистра сравнения

**Описание**

Функция задает значение регистра сравнения/захвата канала у выбранного таймера.

## HAL\_GTIMER\_ChnCompareConfig

**Резюме**

**void** **HAL\_GTIMER\_ChnCompareConfig**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, uint32\_t Compare, GTIMER\_Update\_type UpdateMode )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

uint32\_t Compare – значение регистра сравнения

GTIMER\_Update\_type UpdateMode – когда загружать новое значение (сразу – *GTIMER\_Update\_Immediately*, при CNT = 0 – *GTIMER\_Update\_On\_CNT\_Overflow*)

**Описание**

Функция задает значение регистра сравнения/захвата канала у выбранного таймера, с заданием момента его обновления.

## HAL\_GTIMER\_ChnCompare1Set

**Резюме**

**void HAL\_GTIMER\_ChnCompare1Set**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, uint32\_t Compare )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

uint32\_t Compare – значение регистра сравнения

**Описание**

Функция задает значение регистра сравнения/захвата один канала у выбранного таймера.

## HAL\_GTIMER\_ChnCompare1Config

**Резюме**

**void** **HAL\_GTIMER\_ChnCompare1Config**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, uint32\_t Compare, GTIMER\_Update\_type UpdateMode )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

uint32\_t Compare – значение регистра сравнения

GTIMER\_Update\_type UpdateMode – когда загружать новое значение (сразу – *GTIMER\_Update\_Immediately*, при CNT = 0 – *GTIMER\_Update\_On\_CNT\_Overflow*)

**Описание**

Функция задает значение регистра сравнения/захвата один канала у выбранного таймера, с заданием момента его обновления.

## HAL\_GTIMER\_GetChnCapture

**Резюме**

uint32\_t **HAL\_GTIMER\_GetChnCapture**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

**Описание**

Функция возвращает текущее значение регистра сравнения/захвата канала у выбранного таймера.

## HAL\_GTIMER\_GetChnCapture1

**Резюме**

uint32\_t **HAL\_GTIMER\_GetChnCapture1**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

**Описание**

Функция возвращает текущее значение регистра сравнения/захвата один канала у выбранного таймера.

## HAL\_GTIMER\_ChnETR\_Cmd

**Резюме**

**void** **HAL\_GTIMER\_ChnETR\_Cmd**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, FunctionalState\_type NewState )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

FunctionalState\_type NewState – состояние (*ENABLE, DISABLE*)

**Описание**

Функция управления выводом входа внешней тактовой частоты ETR.

## HAL\_GTIMER\_ChnETRResetConfig

**Резюме**

**void** **HAL\_GTIMER\_ChnETRResetConfig**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, FunctionalState\_type NewState )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

FunctionalState\_type NewState – состояние (*ENABLE, DISABLE*)

**Описание**

Функция управления сбросом от вывода входа внешней тактовой частоты ETR.

## HAL\_GTIMER\_ChnBRKResetConfig

**Резюме**

**void** **HAL\_GTIMER\_ChnBRKResetConfig**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, FunctionalState\_type NewState )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

FunctionalState\_type NewState – состояние (*ENABLE, DISABLE*)

**Описание**

Функция управления сбросом от вывода входа внешнего сигнала сброса BRK.

## HAL\_GTIMER\_ChnREFFormatConfig

**Резюме**

**void** **HAL\_GTIMER\_ChnREFFormatConfig**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, GTIMER\_PWM\_REF\_type Format )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

GTIMER\_PWM\_REF\_type Format – формат

**Описание**

Функция настройки формата REF сигнала канала у выбранного таймера. Формат Format описан в структуре GTIMER\_ChnInit\_type в поле REF\_Format*.*

## HAL\_GTIMER\_ChnCapturePrescalerConfig

**Резюме**

**void** **HAL\_GTIMER\_ChnCapturePrescalerConfig**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, GTIMER\_Prescaler\_type Prescaler )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

GTIMER\_Prescaler\_type Prescaler – значение пред делителя частоты

**Описание**

Функция настройки пред делителя канала у выбранного таймера. Формат Prescaler описан в структуре GTIMER\_ChnInit\_type в поле Prescaler*.*

## HAL\_GTIMER\_ChnEventSourceConfig

**Резюме**

**void** **HAL\_GTIMER\_ChnEventSourceConfig**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, GTIMER\_EventIn\_type EventSource )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

GTIMER\_EventIn\_type EventSource – вид события

**Описание**

Функция настройки входного события канала у выбранного таймера. Формат EventSource описан в структуре GTIMER\_ChnInit\_type в поле EventSource*.*

## HAL\_GTIMER\_ChnFilterConfig

**Резюме**

**void** **HAL\_GTIMER\_ChnFilterConfig**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, GTIMER\_Filter\_type Filter )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

GTIMER\_Filter\_type Filter – параметры фильтра

**Описание**

Функция настройки входного фильтра канала у выбранного таймера. Формат Filter описан в структуре GTIMER\_ChnInit\_type в поле FilterConf*.*

## HAL\_GTIMER\_GetChnWriteComplete

**Резюме**

State\_type **HAL\_GTIMER\_GetChnWriteComplete**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

**Описание**

Функция возвращает текущее состояние записи регистра CCR канала выбранного таймера:

*RESET* = 0

*SET* = 1

## HAL\_GTIMER\_ChnCCR1\_EventSourceConfig

**Резюме**

**void** **HAL\_GTIMER\_ChnCCR1\_EventSourceConfig**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, GTIMER\_EventIn\_type EventSource )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

GTIMER\_EventIn\_type EventSource – вид события

**Описание**

Функция настройки входного события CAP1 канала у выбранного таймера. Формат EventSource описан в структуре GTIMER\_ChnInit\_type в поле EventSource*.*

## HAL\_GTIMER\_ChnCCR1\_Cmd

**Резюме**

**void** **HAL\_GTIMER\_ChnCCR1\_Cmd**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, FunctionalState\_type NewState )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

FunctionalState\_type NewState – состояние (*ENABLE, DISABLE*)

**Описание**

Функция управления работой регистра CCR1 канала у выбранного таймера.

## HAL\_GTIMER\_ChnOutInit

**Резюме**

**void** **HAL\_GTIMER\_ChnOutInit**( GTIMER\_type\* GTIMERx, **const** GTIMER\_ChnOutInit\_type\* GTIMER\_ChnOutInitStruct )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

**const** GTIMER\_ChnOutInit\_type\* GTIMER\_ChnOutInitStruct – указатель на структуру с параметрами выхода канала таймера

**Описание**

Функция настройки прямого и инверсного выхода канала выбранного таймера.

## HAL\_GTIMER\_ChnOutStructInit

**Резюме**

**void** **HAL\_GTIMER\_ChnOutStructInit**( GTIMER\_ChnOutInit\_type\* GTIMER\_ChnOutInitStruct )

GTIMER\_ChnOutInit\_type\* GTIMER\_ChnOutInitStruct – указатель на структуру с параметрами выхода канала таймера

**Описание**

Функция задает параметры инициализационной структуры выхода канала таймера значениями по умолчанию:

GTIMER\_ChnOutInitStruct->DirOut\_Polarity = *GTIMER\_CHOPolarity\_NonInverted*;

GTIMER\_ChnOutInitStruct->DirOut\_Source = *GTIMER\_CH\_OutSrc\_Only\_0*;

GTIMER\_ChnOutInitStruct->DirOut\_Mode = *GTIMER\_CH\_OutMode\_Input*;

GTIMER\_ChnOutInitStruct->NegOut\_Polarity = *GTIMER\_CHOPolarity\_NonInverted*;

GTIMER\_ChnOutInitStruct->NegOut\_Source = *GTIMER\_CH\_OutSrc\_Only\_0*;

GTIMER\_ChnOutInitStruct->NegOut\_Mode = *GTIMER\_CH\_OutMode\_Input*;

GTIMER\_ChnOutInitStruct->DTG\_MainPrescaler = 0;

GTIMER\_ChnOutInitStruct->DTG\_AuxPrescaler = 0;

GTIMER\_ChnOutInitStruct->DTG\_ClockSource = *GTIMER\_CH\_DTG\_ClkSrc\_GTIMER\_CLK*;

## HAL\_GTIMER\_ChnOutConfig

**Резюме**

**void** **HAL\_GTIMER\_ChnOutConfig**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, GTIMER\_CHO\_SOURCE\_type OutSource, GTIMER\_CHO\_MODE\_type Mode, GTIMER\_CHO\_POLARITY\_type Polarity )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

GTIMER\_CHO\_SOURCE\_type OutSource – источник сигнала

GTIMER\_CHO\_MODE\_type Mode – режим работы (вход/выход)

GTIMER\_CHO\_POLARITY\_type Polarity – инверсия выхода

**Описание**

Функция настройки прямого выхода канала выбранного таймера. Значения аргументов описаны в структуре GTIMER\_ChnOutInit\_type и соответствуют полям DirOut\_Source, DirOut\_Mode, DirOut\_Polarity.

## HAL\_GTIMER\_ChnNOutConfig

**Резюме**

**void HAL\_GTIMER\_ChnNOutConfig**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, GTIMER\_CHO\_SOURCE\_type OutSource, GTIMER\_CHO\_MODE\_type Mode, GTIMER\_CHO\_POLARITY\_type Polarity )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

GTIMER\_CHO\_SOURCE\_type OutSource – источник сигнала

GTIMER\_CHO\_MODE\_type Mode – режим работы (вход/выход)

GTIMER\_CHO\_POLARITY\_type Polarity – инверсия выхода

**Описание**

Функция настройки инвесного выхода канала выбранного таймера. Значения аргументов описаны в структуре GTIMER\_ChnOutInit\_type и соответствуют полям NegOut\_Source, NegOut\_Mode, NegOut\_Polarity.

## HAL\_GTIMER\_ChnOutDTGConfig

**Резюме**

**void HAL\_GTIMER\_ChnOutDTGConfig**( GTIMER\_type\* GTIMERx, GTIMER\_CH\_type Channel, uint16\_t MainPrescaler, uint16\_t AuxPrescaler, GTIMER\_CH\_DTG\_CLKSRC\_type ClockSource )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

GTIMER\_CH\_type Channel – канала таймера (*GTIMER\_CH1 – GTIMER\_CH4*)

uint16\_t MainPrescaler – основной делитель частоты

uint16\_t AuxPrescaler – предварительный делитель частоты

GTIMER\_CH\_DTG\_CLKSRC\_type ClockSource – источник тактирования (*GTIMER\_CH\_DTG\_ClkSrc\_GTIMER\_CLK* или *GTIMER\_CH\_DTG\_ClkSrc\_FDTS*)

**Описание**

Функция настройки генератора мертвого времени. Тактироваться генератор может как тактовой частотой таймера, так и частотой семплирования таймера.

DTG\_delay = MainPrescaler \* ( AuxPrescaler + 1 )

## HAL\_GTIMER\_GetStatus

**Резюме**

uint32\_t **HAL\_GTIMER\_GetStatus**( GTIMER\_type\* GTIMERx )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

**Описание**

Функция возвращает статусный регистр таймера.

## HAL\_GTIMER\_GetFlagStatus

**Резюме**

State\_type **HAL\_GTIMER\_GetFlagStatus**( GTIMER\_type\* GTIMERx, uint32\_t Flag )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

uint32\_t Flag ­– интересующие флаги таймера

**Описание**

Функция возвращает состояние интересующих флагов таймера. Флаги могут быть:

GTIMER\_EVENT\_CNT\_ZERO – событие совпадения CNT с нулем

GTIMER\_EVENT\_CNT\_ARR – событие совпадения CNT с ARR

GTIMER\_EVENT\_ETR\_RISING\_EDGE – событие переднего фронта на входе ETR

GTIMER\_EVENT\_ETR\_FALLING\_EDGE – событие заднего фронта на входе ETR

GTIMER\_EVENT\_BRK – триггерированное по PCLK состояние входа BRK

GTIMER\_EVENT\_CCR\_CAP\_CH1 – событие настроенного фронта на входе CHxi каналов таймера

GTIMER\_EVENT\_CCR\_CAP\_CH2

GTIMER\_EVENT\_CCR\_CAP\_CH3

GTIMER\_EVENT\_CCR\_CAP\_CH4

GTIMER\_EVENT\_CCR\_REF\_CH1 – событие переднего фронта на выходе REF каналов таймера

GTIMER\_EVENT\_CCR\_REF\_CH2

GTIMER\_EVENT\_CCR\_REF\_CH3

GTIMER\_EVENT\_CCR\_REF\_CH4

GTIMER\_EVENT\_CCR\_CAP1\_CH1 – событие настроенного фронта на входе CHxi каналов таймера

GTIMER\_EVENT\_CCR\_CAP1\_CH2

GTIMER\_EVENT\_CCR\_CAP1\_CH3

GTIMER\_EVENT\_CCR\_CAP1\_CH4

Допускается передавать несколько событий через логическое ИЛИ:

**HAL\_GTIMER\_GetFlagStatus**( LX\_GTIMER0, GTIMER\_EVENT\_CNT\_ZERO | GTIMER\_EVENT\_CNT\_ARR | GTIMER\_EVENT\_BRK )

Возвращаемое значение:

*RESET* = 0

*SET* = 1

## HAL\_GTIMER\_ClearFlag

**Резюме**

**void** **HAL\_GTIMER\_ClearFlag**( GTIMER\_type\* GTIMERx, uint32\_t Flags )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

uint32\_t Flags ­– флаги таймера

**Описание**

Функция производит сброс заданных флагов выбранного таймера. Параметр Flags подробно расписан в описании функции **HAL\_GTIMER\_GetFlagStatus.**

## HAL\_GTIMER\_DMACmd

**Резюме**

**void** **HAL\_GTIMER\_DMACmd**( GTIMER\_type\* GTIMERx, uint32\_t GTIMER\_DMASource, GTIMER\_DMA\_REQ\_type GTIMER\_DMA\_Request, FunctionalState\_type NewState )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

uint32\_t GTIMER\_DMASource – источник запросов (флаг таймера)

GTIMER\_DMA\_REQ\_type GTIMER\_DMA\_Request – запрос к DMA

FunctionalState\_type NewState – состояние (*ENABLE, DISABLE*)

**Описание**

Функция настройки источника запросов к DMA от выбранного таймера. В качестве источника запросов указывается флаг таймера, подробно описанный в описании функции **HAL\_GTIMER\_GetFlagStatus**.GTIMER\_DMA\_Request отвечает за выбор запроса к DMA (настройка выбранного запроса к DMA выполняется в соответствующем регистре DMA\_RE):

*GTIMER\_DMA\_REQ1*

*GTIMER\_DMA\_REQ2*

*GTIMER\_DMA\_REQ3*

*GTIMER\_DMA\_REQ4*

*GTIMER\_DMA\_REQ*

## HAL\_GTIMER\_ITConfig

**Резюме**

**void** **HAL\_GTIMER\_ITConfig**( GTIMER\_type\* GTIMERx, uint32\_t GTIMER\_IT, FunctionalState\_type NewState )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

uint32\_t GTIMER\_IT – источник запросов на прерывание (флаг таймера)

FunctionalState\_type NewState – состояние (*ENABLE, DISABLE*)

**Описание**

Функция настройки формирования прерываний от выбранного таймера. В качестве GTIMER\_IT указывается флаг таймера, подробно описанный в описании функции **HAL\_GTIMER\_GetFlagStatus.**

## HAL\_GTIMER\_GetITStatus

**Резюме**

State\_type **HAL\_GTIMER\_GetITStatus**( GTIMER\_type\* GTIMERx, uint32\_t GTIMER\_IT )

GTIMER\_type\* GTIMERx – указатель на структуру с регистрами таймера

uint32\_t GTIMER\_IT – интересующее прерывание (флаг таймера)

**Описание**

Функция возвращает статус интересующих прерываний выбранного таймера. В качестве GTIMER\_IT указывается флаг таймера, подробно описанный в описании функции **HAL\_GTIMER\_GetFlagStatus.**

Возвращаемое значение:

*RESET* = 0

*SET* = 1

# Пример работы

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

**#define** CoreClk\_kHz 100000

**#define** GTIMER\_DMA\_CH 5

**#define** PWM\_DATA\_LEN 6

**#define** GTIMER\_CCR1\_offset 0x4

uint32\_t ulPwm[ PWM\_DATA\_LEN ] = { 1, 3, 10, 30, 100, 300 };

**static** \_\_builtin\_quad pwm\_tcb;

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

{

uint64\_t ullData;

uint32\_t \*ptr = ( uint32\_t\* ) &pwm\_tcb;

GTIMER\_CntInit\_type xTimerCnt;

GTIMER\_ChnInit\_type xTimerCh;

GTIMER\_ChnOutInit\_type xTimerChOut;

HAL\_PLL\_CoreSetup( CoreClk\_kHz );

// Разрешаем работу таймера 1

LX\_CMU->CFG1.b.GTMR1\_EN = 1;

// Настраиваем таймер 1

HAL\_GTIMER\_CntStructInit( &xTimerCnt );

xTimerCnt.Period = 500;

xTimerCnt.Prescaler = 999;

HAL\_GTIMER\_CntInit( LX\_GTIMER1, &xTimerCnt );

// Настраиваем 1 канал таймера 1

HAL\_GTIMER\_ChnStructInit( &xTimerCh );

xTimerCh.ChNumber = *GTIMER\_CH1*;

xTimerCh.REF\_Format = *GTIMER\_CH\_REF\_Format6*;

xTimerCh.CCR\_UpdateMode = *GTIMER\_Update\_On\_CNT\_Overflow*;

HAL\_GTIMER\_ChnInit( LX\_GTIMER1, &xTimerCh );

// Настраиваем выход 1 канала таймера 1

HAL\_GTIMER\_ChnOutStructInit( &xTimerChOut );

xTimerChOut.ChNumber = *GTIMER\_CH1*;

xTimerChOut.DirOut\_Mode = *GTIMER\_CH\_OutMode\_Output*;

xTimerChOut.DirOut\_Source = *GTIMER\_CH\_OutSrc\_REF*;

HAL\_GTIMER\_ChnOutInit( LX\_GTIMER1, &xTimerChOut );

// Настраиваем вывод, используемый 1 прямым каналом таймера 1

HAL\_GPIO\_Init( LX\_GPIO\_PB, *GPIO\_PIN\_8*, *GPIO\_PinMode\_Alt* );

/\* ---------------------------- Настройка DMA -------------------------------------- \*/

HAL\_DMA\_Stop( GTIMER\_DMA\_CH );

\*( ptr + 0 ) = ( uint32\_t ) ulPwm; // Источник информации

\*( ptr + 1 ) = ( PWM\_DATA\_LEN << 16 ) | 1; // Количество слов + инкремент адреса

\*( ptr + 2 ) = 0; // Количество попыток передачи по X + инкремент адреса X

\*( ptr + 3 ) = TCB\_NORMAL | // Длина передаваемых данных (операнда) в одном цикле обмена

TCB\_INTMEM; // Источник во внутренней памяти

HAL\_DMA\_CreateChannelDest( GTIMER\_DMA\_CH, &pwm\_tcb, &pwm\_tcb );

HAL\_DMA\_RqstSet( GTIMER\_DMA\_CH, *dmaGTIMER1\_REQ1* );

// С помощью регистра DCA указываем в какой регистр таймера загружать новые данные

ullData = DMA\_DCA\_INT\_MEM | ( base\_GTIMER1 + GTIMER\_CCR1\_offset );

HAL\_DMA\_WriteDCA( GTIMER\_DMA\_CH, &ullData );

HAL\_DMA\_WriteDC( GTIMER\_DMA\_CH, &pwm\_tcb );

// Разрешаем запросы к DMA от таймера 1

HAL\_GTIMER\_DMACmd( LX\_GTIMER1, GTIMER\_EVENT\_CNT\_ARR, *GTIMER\_DMA\_REQ1*, *ENABLE* );

HAL\_GTIMER\_Cmd( LX\_GTIMER1, *ENABLE* );

**while**( 1 )

{

HAL\_SYS\_WaitMs( 10 );

}

}

В данном примере показан вывод ШИМ с помощью DMA через 1 прямой канал таймера 1. Запрос к DMA возникает в момент совпадения CNT с ARR таймера.

Запросы от таймеров:

*dmaGTIMERX\_REQ*  – запрос REQ, настраиваемый в регистре DMA\_RE

*dmaGTIMERX\_REQ1* – запрос REQ1, настраиваемый в регистре DMA\_RE1

*dmaGTIMERX\_REQ2* – запрос REQ2, настраиваемый в регистре DMA\_RE2

*dmaGTIMERX\_REQ3* – запрос REQ3, настраиваемый в регистре DMA\_RE3

*dmaGTIMERX\_REQ4* – запрос REQ4, настраиваемый в регистре DMA\_RE4