1. **HAL\_PUx\_Write**

**Резюме**

**void** **HAL\_PUx\_Write** (PUx\_num PU\_reg, uint32\_t \*InitStruct) – функция записи в регистры группы PU(0-7) и PU\_CR.

**Описание**

Данная функция позволяет запрограммировать любой регистр группы PU. Первым аргументом в функцию должно передаваться имя регистра, в который требуется записать значение. Имя регистра выбирается из определенного списка:

**typedef** **enum** {

*PU0*,

*PU1*,

*PU2*,

*PU3*,

*PU4*,

*PU5*,

*PU6*,

*PU7*,

*PU\_CR*

}PUx\_num;

Вторым аргументом передается адрес структуры, в которой задаются все необходимые параметры. С помощью структуры каждого регистра можно обратиться как к отдельным битам, так и к регистру в целом.

Для того, чтобы получить доступ к регистрам данной группы, необходимо убедиться, что 30 бит (EXT\_ MODE) в регистре SQSTAT имеет значение 1. Если данный бит будет иметь значение 0, регистры группы PU будут недоступны. Как установить SQSTAT[30] в значение 1, см. документ «[Спецификация на микросхему 1967ВН028](https://ic.milandr.ru/upload/iblock/1f7/1f7be64124ace23c2f892e54622d6858.pdf)», пункт 7.3.9.6 «Регистр включения расширенных функций».

**Пример**

**#include** "hal\_PU.h"

**#include** <stdint.h>

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

PU\_CR\_reg PU\_CR\_reg;

PUx\_reg PUx;

uint32\_t \*ptr\_pu\_cr, \*ptr\_pux;

uint32\_t PU\_0, PU\_1, PU\_2, PU\_3, PU\_4, PU\_5, PU\_6, PU\_7, PUCR;

//Init struct PUx

PUx.b.STA = 0x1d0;

PUx.b.ENDA = 0x1df;

PUx.b.JK\_AP = 1;

PUx.b.I\_AP = 2;

PUx.b.H\_AP = 3;

ptr\_pux = (uint32\_t \*)&PUx;

//Init struct PU\_CR

PU\_CR\_reg.b.DC\_ON = 1;

PU\_CR\_reg.b.IC\_ON = 1;

PU\_CR\_reg.b.EN\_2dQW = 1;

PU\_CR\_reg.b.EN\_2iQW = 1;

PU\_CR\_reg.b.SOC\_speed\_up = 0;

PU\_CR\_reg.b.PSD\_fun = 1;

PU\_CR\_reg.b.SP\_byte = 1;

PU\_CR\_reg.b.MIN\_MAX = 1;

PU\_CR\_reg.b.LC\_sup = 0;

PU\_CR\_reg.b.PU0\_EN = 1;

PU\_CR\_reg.b.PU1\_EN = 1;

PU\_CR\_reg.b.PU2\_EN = 1;

PU\_CR\_reg.b.PU3\_EN = 1;

PU\_CR\_reg.b.PU4\_EN = 1;

PU\_CR\_reg.b.PU5\_EN = 1;

PU\_CR\_reg.b.PU6\_EN = 1;

PU\_CR\_reg.b.PU7\_EN = 1;

PU\_CR\_reg.b.USOC\_EN = 3;

ptr\_pu\_cr = (uint32\_t \*)&PU\_CR\_reg;

HAL\_PUx\_Write(*PU\_CR*, ptr\_pu\_cr);

HAL\_PUx\_Write(*PU0*, ptr\_pux);

HAL\_PUx\_Write(*PU1*, ptr\_pux);

HAL\_PUx\_Write(*PU2*, ptr\_pux);

HAL\_PUx\_Write(*PU3*, ptr\_pux);

HAL\_PUx\_Write(*PU4*, ptr\_pux);

HAL\_PUx\_Write(*PU5*, ptr\_pux);

HAL\_PUx\_Write(*PU6*, ptr\_pux);

HAL\_PUx\_Write(*PU7*, ptr\_pux);

**return** 0;

}

1. **HAL\_PUx\_Read**

**Резюме**

uint32\_t **HAL\_PUx\_Read** (PUx\_num PU\_reg)- функция чтения регистра группы PU.

**Описание**

Данная функция позволяет выполнить чтение любого регистра группы PU. В качестве аргумента функции передается имя регистра, значение которого необходимо прочитать. Функция возвращает полученное из регистра значение. Имя регистра должно выбираться из определенного списка:

**typedef** **enum** {

*PU0*,

*PU1*,

*PU2*,

*PU3*,

*PU4*,

*PU5*,

*PU6*,

*PU7*,

*PU\_CR*

}PUx\_num;

**Пример**

**#include** "hal\_PU.h"

**#include** <stdint.h>

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

uint32\_t PU0\_value;

PU0\_value = HAL\_PUx\_Read(*PU0*);

//Данная функция вернет в переменную PU0\_value

//значение регистра PU0

**return** 0;

}