NVRAM 驱动

V1.0

制作者 Doxygen 1.9.6

1 模块索引 **1**
1.1 模块 . . . 1

2 结构体索引 **3**
2.1 结构体 . . . 3

3 文件索引 **5**
3.1 文件列表 . . . 5

4 模块说明 **7**
4.1 NVSRAM驱动 . . . 7
4.1.1 详细描述 . . . 7
4.1.2 变量说明 . . . 7
4.1.2.1 _nvsram . . . 7
4.1.2.2 _nvsram_rxbuf . . . 8
4.1.2.3 _nvsram_txbuf . . . 8
4.2 NVSRAM外部枚举与结构体类型 . . . 8
4.2.1 详细描述 . . . 8
4.2.2 枚举类型说明 . . . 8
4.2.2.1 NVSRAM_BPL . . . 8
4.2.2.2 NVSRAM_RET . . . 9
4.3 NVSRAM外部宏 . . . 9
4.3.1 详细描述 . . . 9
4.3.2 宏定义说明 . . . 9
4.3.2.1 Drv_Nvsram_TSET . . . 9
4.4 NVSRAM外部函数 . . . 9
4.4.1 详细描述 . . . 10
4.4.2 函数说明 . . . 10
4.4.2.1 Nvsram_Burst_Read_Buff() . . . 10
4.4.2.2 Nvsram_Burst_Write_Buff() . . . 11
4.4.2.3 Nvsram_Get_Bp() . . . 11
4.4.2.4 Nvsram_Get_SN() . . . 12
4.4.2.5 Nvsram_Get_Struct() . . . 12
4.4.2.6 Nvsram_Init() . . . 12
4.4.2.7 Nvsram_Set_Bp() . . . 13
4.4.2.8 Nvsram_Set_SN() . . . 13
4.4.2.9 Nvsram_Soft_Recall() . . . 14
4.4.2.10 Nvsram_Soft_Store() . . . 14
4.4.2.11 Nvsram_Test() . . . 15
4.4.3 变量说明 . . . 15
4.4.3.1 test_rx . . . 15
4.4.3.2 test_tx . . . 16
4.5 设备驱动 . . . 16

4.5.1 详细描述 . . . 16
4.6 NVSRAM静态函数宏 . . . 16
4.6.1 详细描述 . . . 16
4.6.2 宏定义说明 . . . 16
4.6.2.1 Nvsram_Delay_Ms . . . 16
4.6.2.2 Nvsram_Get_tick . . . 17
4.6.2.3 Nvsram_Spi_TR . . . 17
4.7 NVSRAM设备参数 . . . 17
4.7.1 详细描述 . . . 18
4.7.2 宏定义说明 . . . 18
4.7.2.1 NVSRAM_BUF_SIZE . . . 18
4.7.2.2 NVSRAM_DEV_ID . . . 18
4.7.2.3 NVSRAM_MAX_ADDR . . . 18
4.7.2.4 NVSRAM_SPEC_FRDID . . . 18
4.7.2.5 NVSRAM_SPEC_FRDSN . . . 19
4.7.2.6 NVSRAM_SPEC_NV_ASDISB . . . 19
4.7.2.7 NVSRAM_SPEC_NV_ASENB . . . 19
4.7.2.8 NVSRAM_SPEC_NV_RECALL . . . 19
4.7.2.9 NVSRAM_SPEC_NV_STORE . . . 19
4.7.2.10 NVSRAM_SPEC_RDID . . . 19
4.7.2.11 NVSRAM_SPEC_RDSN . . . 19
4.7.2.12 NVSRAM_SPEC_SLEEP . . . 19
4.7.2.13 NVSRAM_SPEC_WRSN . . . 20
4.7.2.14 NVSRAM_SRAM_FREAD . . . 20
4.7.2.15 NVSRAM_SRAM_READ . . . 20
4.7.2.16 NVSRAM_SRAM_WRITE . . . 20
4.7.2.17 NVSRAM_STATUS_BP0 . . . 20
4.7.2.18 NVSRAM_STATUS_BP1 . . . 20
4.7.2.19 NVSRAM_STATUS_FRDSR . . . 20
4.7.2.20 NVSRAM_STATUS_RDSR . . . 20
4.7.2.21 NVSRAM_STATUS_RDY . . . 21
4.7.2.22 NVSRAM_STATUS_SNL . . . 21
4.7.2.23 NVSRAM_STATUS_WEN . . . 21
4.7.2.24 NVSRAM_STATUS_WPEN . . . 21
4.7.2.25 NVSRAM_STATUS_WRDI . . . 21
4.7.2.26 NVSRAM_STATUS_WREN . . . 21
4.7.2.27 NVSRAM_STATUS_WRSR . . . 21
4.8 NVSRAM静态函数 . . . 22
4.8.1 详细描述 . . . 22
4.8.2 函数说明 . . . 22
4.8.2.1 nvsram_check_id() . . . 22
4.8.2.2 nvsram_get_ready() . . . 22

4.8.2.3 nvsram_get_snw() . . . 23
4.8.2.4 nvsram_verify_set_wen() . . . 23

5 结构体说明 **25**
5.1 Nvsram_Struct_t结构体 参考 . . . 25
5.1.1 详细描述 . . . 25
5.1.2 结构体成员变量说明 . . . 25
5.1.2.1 enabled . . . 25
5.1.2.2 error . . . 25
5.1.2.3 id . . . 26
5.1.2.4 snw . . . 26

6 文件说明 **27**
6.1 E:/STM32SVN/Doxy实验/NVSRAM驱动/Drv_Nvsram.c 文件参考 . . . 27
6.1.1 详细描述 . . . 29
6.2 E:/STM32SVN/Doxy实验/NVSRAM驱动/Drv_Nvsram.h 文件参考 . . . 29
6.2.1 详细描述 . . . 30
6.3 Drv_Nvsram.h . . . 31

Index **33**

Chapter 1

模块索引

1.1 模块

这里列出了所有模块:

设备驱动 16
NVSRAM驱动 7
NVSRAM外部枚举与结构体类型 8
NVSRAM外部宏 9
NVSRAM外部函数 9
NVSRAM静态函数宏 16
NVSRAM设备参数 17
NVSRAM静态函数 22

Chapter 2

结构体索引

2.1 结构体

这里列出了所有结构体，并附带简要说明:

Nvsram_Struct_t
NVSRAM结构体 25

Chapter 3

文件索引

3.1 文件列表

这里列出了所有文件，并附带简要说明:

E:/STM32SVN/Doxy实验/NVSRAM驱动/ **Drv_Nvsram.c**
非易失性RAM驱动 27
E:/STM32SVN/Doxy实验/NVSRAM驱动/ **Drv_Nvsram.h**
非易失性RAM驱动 29

Chapter 4

模块说明

4.1 NVSRAM驱动

CY14B101Q2A-SXI.

模块

- **NVSRAM**外部枚举与结构体类型
- **NVSRAM**外部宏
- **NVSRAM**外部函数
- **NVSRAM**静态函数宏
- **NVSRAM**设备参数
- **NVSRAM**静态函数

变量

- static **Nvsram_Struct_t** **_nvsram** = {0}
- static uint8_t **_nvsram_txbuf** [**NVSRAM_BUF_SIZE**+4] = {0}
- static uint8_t **_nvsram_rxbuf** [**NVSRAM_BUF_SIZE**+4] = {0}

4.1.1 详细描述

CY14B101Q2A-SXI.

4.1.2 变量说明

4.1.2.1 _nvsram

Nvsram_Struct_t _nvsram = {0} [static]

4.1.2.2 _nvsram_rxbuf

```
uint8_t _nvsram_rxbuf[ NVSRAM_BUF_SIZE+4] = {0}  [static]
```

前4个字节 1个字节命令码+3个字节地址

4.1.2.3 _nvsram_txbuf

```
uint8_t _nvsram_txbuf[ NVSRAM_BUF_SIZE+4] = {0}  [static]
```

前4个字节 1个字节命令码+3个字节地址

4.2 NVSRAM外部枚举与结构体类型

结构体

- struct **Nvsram_Struct_t**
  *NVSRAM*结构体

枚举

- enum **NVSRAM_BPL** { **NVSRAM_BPL0** = 0x00 , **NVSRAM_BPL1** , **NVSRAM_BPL2** , **NVSRAM_BPL3** }
  内存写保护等级
- enum **NVSRAM_RET** { **NVSRAM_SUCCEED** , **NVSRAM_PRAM_ERRO** , **NVSRAM_Busy** }
  *NVSRAM*返回值说明

4.2.1 详细描述

4.2.2 枚举类型说明

4.2.2.1 NVSRAM_BPL

```
enum NVSRAM_BPL
```

内存写保护等级

枚举值

NVSRAM_BPL0	内存不保护
NVSRAM_BPL1	内存保护 0x18000–0x1FFFF
NVSRAM_BPL2	内存保护 0x10000–0x1FFFF
NVSRAM_BPL3	内存保护 0x00000–0x1FFFF

4.2.2.2 NVSRAM_RET

enum **NVSRAM_RET**

NVSRAM返回值说明

枚举值

NVSRAM_SUCCEED	成功
NVSRAM_PRAM_ERRO	传入参数错误
NVSRAM_Busy	设备忙

4.3 NVSRAM外部宏

宏定义

- #define **Drv_Nvsram_TSET**

4.3.1 详细描述

4.3.2 宏定义说明

4.3.2.1 Drv_Nvsram_TSET

#define Drv_Nvsram_TSET

设备测试 开关

4.4 NVSRAM外部函数

函数

- uint8_t **Nvsram_Set_Bp** (uint8_t Level)
  NVSRAM 设置写保护.
- uint8_t **Nvsram_Get_Bp** (void)
  NVSRAM 获取写保护块.
- uint32_t **Nvsram_Burst_Write_Buff** (uint32_t nvsram_addr, uint8_t *w_buf, uint32_t len)
  NVSRAM 写内存函数
- uint32_t **Nvsram_Burst_Read_Buff** (uint32_t nvsram_addr, uint8_t *r_buf, uint32_t len)

NVSRAM 读内存函数

- uint8_t **Nvsram_Soft_Store** (void)
  NVSRAM 软件存储
- uint8_t **Nvsram_Soft_Recall** (void)
  NVSRAM 软件加载数据
- uint8_t **Nvsram_Set_SN** (uint8_t *sn_buff)
  NVSRAM 设置序列号
- void **Nvsram_Get_SN** (uint8_t *sn_buff)
  NVSRAM 获取序列号
- uint8_t **Nvsram_Init** (void)
  NVSRAM 设备初始化
- void **Nvsram_Get_Struct** (**Nvsram_Struct_t** *nvsram_st)
  NVSRAM 获取设备状态
- void **Nvsram_Test** (void)
  NVSRAM 驱动测试

变量

- uint8_t **test_tx** [**NVSRAM_BUF_SIZE**] = {0}
- uint8_t **test_rx** [**NVSRAM_BUF_SIZE**] = {0}

4.4.1 详细描述

4.4.2 函数说明

4.4.2.1 Nvsram_Burst_Read_Buff()

```
uint32_t Nvsram_Burst_Read_Buff (
            uint32_t nvsram_addr,
            uint8_t * r_buf,
            uint32_t len )
```

NVSRAM 读内存函数

注解

数据进行读操作时,如果地址大于0x1FFFF,地址将自动在0x00000中依次增加读取
缓存区大小为 256Byte, SPI通信速率为 12.5MHZ 读速率为 262.83KByte/S

参数

in	*nvsram_addr*	NVSRAM内存地址 0x00000–0x1FFFF
out	*r_buf:*读缓存区地址	
in	*len:*读字节长度	

返回值

读出字节数量	

4.4.2.2 Nvsram_Burst_Write_Buff()

```
uint32_t Nvsram_Burst_Write_Buff (
            uint32_t nvsram_addr,
            uint8_t * w_buf,
            uint32_t len )
```

NVSRAM 写内存函数

注解

1) 数据进行突发写操作时，当突发写到达一个受保护的块地址时，它将地址增
量继续到受保护空间，但是不向受保护内存写入任何数据。如果地址增加到
突发写操作带到未受保护的空间，然后恢复写入
2) 数据进行突发写操作时，如果地址大于0x1FFFF，地址将自动在0x00000处依次增加写入

3) 缓存区大小为 256Byte, SPI通信速率为 12.5MHZ 写速率为 262.83KByte/S

参数

in	*nvsram_addr:NVSRAM*内存地址	0x00000–0x1FFFF
in	*w_buf:*写缓存区地址	
in	*len:*写入字节长度	

返回值

写入字节数量	

4.4.2.3 Nvsram_Get_Bp()

```
uint8_t Nvsram_Get_Bp (
            void  )
```

NVSRAM 获取写保护块.

返回值

NVSRAM_BPL0	0 内存不保护

返回值

NVSRAM_BPL1	1/4 内存保护 0x18000–0x1FFFF
NVSRAM_BPL2	1/2 内存保护 0x10000–0x1FFFF
NVSRAM_BPL3	ALL 内存保护 0x00000–0x1FFFF

4.4.2.4 Nvsram_Get_SN()

```
void Nvsram_Get_SN (
            uint8_t * sn_buff )
```

NVSRAM 获取序列号

参数

out	*sn_buff*	序列号缓存区保存地址 该缓存区要大于等于8个字节

4.4.2.5 Nvsram_Get_Struct()

```
void Nvsram_Get_Struct (
             Nvsram_Struct_t * nvsram_st )
```

NVSRAM 获取设备状态

参数

out	*nvsram_st*	设备状态结构体地址

4.4.2.6 Nvsram_Init()

```
uint8_t Nvsram_Init (
            void  )
```

NVSRAM 设备初始化

注解

硬件初始化 等待设备就绪,加载ID号,加载序列号写入标志

返回值

NVSRAM_SUCCEED:成功	
NVSRAM_PRAM_ERRO:传入参数错误	
NVSRAM_Busy:设备忙	

4.4.2.7 Nvsram_Set_Bp()

```
uint8_t Nvsram_Set_Bp (
            uint8_t Level )
```

NVSRAM 设置写保护.

参数

in	*Level*	设置写保护等级 • **NVSRAM_BPL0** (p. 8) 0 内存不保护 • **NVSRAM_BPL1** (p. 8) 1/4 内存保护 0x18000–0x1FFFF • **NVSRAM_BPL2** (p. 8) 1/2 内存保护 0x10000–0x1FFFF • **NVSRAM_BPL3** (p. 8) ALL 内存保护 0x00000–0x1FFFF

返回值

NVSRAM_SUCCEED:成功	
NVSRAM_PRAM_ERRO:传入参数错误	
NVSRAM_Busy:设备忙	

4.4.2.8 Nvsram_Set_SN()

```
uint8_t Nvsram_Set_SN (
            uint8_t * sn_buff )
```

NVSRAM 设置序列号

注解

内部有8个字节的序列号空间，但只能被写一次(保存到量子陷阱之后),
写一次之后,状态寄存器中SNL bit 为1,并且永远不能擦除该标志位,序列号
永远不能再被更改,由于需要存储功能,将会耗时6ms左右,不建议将此函数放入线程中使用

参数

in	*sn_buff*	序列号缓存区保存地址 该缓存区要大于等于8个字节

返回值

NVSRAM_SUCCEED:成功	
NVSRAM_PRAM_ERRO:传入参数错误	
NVSRAM_Busy:设备忙	

4.4.2.9 Nvsram_Soft_Recall()

```
uint8_t Nvsram_Soft_Recall (
            void  )
```

NVSRAM 软件加载数据

注解

将量子陷阱中数据加载到内部SDRAM数据中,由于存储需要一定的时间,如果设备可以上电加载,
不建议使用该函数

返回值

NVSRAM_SUCCEED:成功	
NVSRAM_PRAM_ERRO:传入参数错误	
NVSRAM_Busy:设备忙	

4.4.2.10 Nvsram_Soft_Store()

```
uint8_t Nvsram_Soft_Store (
            void  )
```

NVSRAM 软件存储

注解

将内部SDRAM数据存储到量子陷阱中,由于存储需要一定的时间,如果设备可以掉电保存,
不建议使用该函数

返回值

NVSRAM_SUCCEED:成功	
NVSRAM_PRAM_ERRO:传入参数错误	
NVSRAM_Busy:设备忙	

4.4.2.11 Nvsram_Test()

```
void Nvsram_Test (
             void  )
```

NVSRAM 驱动测试

注解

1)设备初始化测试,函数运行时间,设备ID,设备序列号是否写入
2)设备软件保存指令检测 测试运行时间
3)设备软件数据加载指令检测 测试运行时间
4)设备写函数测试（一次写一个缓存区）
5)设备读函数测试（一次读一个缓存区）
6)设备写函数测试（128K内存）测试写入运行时间
7)设备读函数测试（128K内存）测试写入读出数据的正确性

8)设备读函数测试（128K内存）测试读出运行时间
9)设备掉电数据保存测试 调试状态将内存中写入数据 0x55
断电重启,设备初始化之后,读出数据，判断数据全为0x55

10)设备块保护测试 NVSRAM_BPL0 （128K内存）写入0x11 读出数据判断是否一致
NVSRAM_BPL1 （128K内存）写入0x22 读出数据判断是否一致
NVSRAM_BPL2 （128K内存）写入0x33 读出数据判断是否一致
NVSRAM_BPL3 （128K内存）写入0x44 读出数据判断是否一致
解除块保护 （128K内存）写入0x55 读出数据判断是否一致

4.4.3 变量说明

4.4.3.1 test_rx

```
uint8_t test_rx[ NVSRAM_BUF_SIZE] = {0}
```

4.4.3.2 test_tx

```
uint8_t test_tx[ NVSRAM_BUF_SIZE] = {0}
```

4.5 设备驱动

模块

- **NVSRAM**驱动
  CY14B101Q2A-SXI.

4.5.1 详细描述

4.6 NVSRAM静态函数宏

宏定义

- #define **Nvsram_Delay_Ms**(NMS) HAL_Delay(NMS)
  NVSRAM 毫秒延时.
- #define **Nvsram_Get_tick**() HAL_GetTick()
  NVSRAM 获取系统时间戳.
- #define **Nvsram_Spi_TR**(txbuf, rxbuf, len) SPI_TransmitReceive(SPI_CH1, txbuf, rxbuf, len)
  *NVSRAM SPI*通信接口

4.6.1 详细描述

4.6.2 宏定义说明

4.6.2.1 Nvsram_Delay_Ms

```
#define Nvsram_Delay_Ms(
              NMS ) HAL_Delay(NMS)
```

NVSRAM 毫秒延时.

参数

in	*MS*	延时 N ms

4.6.2.2 Nvsram_Get_tick

```
#define Nvsram_Get_tick( ) HAL_GetTick()
```

NVSRAM 获取系统时间戳.

返回值

系统时间戳	毫秒

4.6.2.3 Nvsram_Spi_TR

```
#define Nvsram_Spi_TR(
              txbuf,
              rxbuf,
              len ) SPI_TransmitReceive(SPI_CH1, txbuf, rxbuf, len)
```

NVSRAM SPI通信接口

参数

in	*txbuf*	发送缓存区地址
out	*rxbuf*	接收缓存区地址
in	*len*	发送字节数

返回值

RET_OK: 成功	;
RET_ERR:失败	

4.7 NVSRAM设备参数

宏定义

- #define **NVSRAM_STATUS_RDSR** 0x05
- #define **NVSRAM_STATUS_FRDSR** 0x0A
- #define **NVSRAM_STATUS_WRSR** 0x01
- #define **NVSRAM_STATUS_WREN** 0x06
- #define **NVSRAM_STATUS_WRDI** 0x04
- #define **NVSRAM_SRAM_READ** 0x03
- #define **NVSRAM_SRAM_FREAD** 0x0B
- #define **NVSRAM_SRAM_WRITE** 0x02
- #define **NVSRAM_SPEC_NV_STORE** 0x3C
- #define **NVSRAM_SPEC_NV_RECALL** 0x60

- #define **NVSRAM_SPEC_NV_ASENB** 0x59
- #define **NVSRAM_SPEC_NV_ASDISB** 0x19
- #define **NVSRAM_SPEC_SLEEP** 0xB9
- #define **NVSRAM_SPEC_WRSN** 0xC2
- #define **NVSRAM_SPEC_RDSN** 0xC3
- #define **NVSRAM_SPEC_FRDSN** 0xC9
- #define **NVSRAM_SPEC_RDID** 0x9F
- #define **NVSRAM_SPEC_FRDID** 0x99
- #define **NVSRAM_STATUS_RDY** 0x01
- #define **NVSRAM_STATUS_WEN** 0x02
- #define **NVSRAM_STATUS_BP0** 0x04
- #define **NVSRAM_STATUS_BP1** 0x08
- #define **NVSRAM_STATUS_SNL** 0x40
- #define **NVSRAM_STATUS_WPEN** 0x80
- #define **NVSRAM_DEV_ID** 0x06818820
- #define **NVSRAM_MAX_ADDR** 0x1FFFF
- #define **NVSRAM_BUF_SIZE** 256

4.7.1 详细描述

4.7.2 宏定义说明

4.7.2.1 NVSRAM_BUF_SIZE

```
#define NVSRAM_BUF_SIZE 256
```

设备一次发送最大字节数

4.7.2.2 NVSRAM_DEV_ID

```
#define NVSRAM_DEV_ID 0x06818820
```

设备ID

4.7.2.3 NVSRAM_MAX_ADDR

```
#define NVSRAM_MAX_ADDR 0x1FFFF
```

设备最大地址

4.7.2.4 NVSRAM_SPEC_FRDID

```
#define NVSRAM_SPEC_FRDID 0x99
```

快读设备ID

4.7.2.5 NVSRAM_SPEC_FRDSN

```
#define NVSRAM_SPEC_FRDSN 0xC9
```

快读序列号

4.7.2.6 NVSRAM_SPEC_NV_ASDISB

```
#define NVSRAM_SPEC_NV_ASDISB 0x19
```

自动储存禁止

4.7.2.7 NVSRAM_SPEC_NV_ASENB

```
#define NVSRAM_SPEC_NV_ASENB 0x59
```

自动储存使能

4.7.2.8 NVSRAM_SPEC_NV_RECALL

```
#define NVSRAM_SPEC_NV_RECALL 0x60
```

内存数据加载

4.7.2.9 NVSRAM_SPEC_NV_STORE

```
#define NVSRAM_SPEC_NV_STORE 0x3C
```

内存数据存储

4.7.2.10 NVSRAM_SPEC_RDID

```
#define NVSRAM_SPEC_RDID 0x9F
```

读设备ID

4.7.2.11 NVSRAM_SPEC_RDSN

```
#define NVSRAM_SPEC_RDSN 0xC3
```

读序列号

4.7.2.12 NVSRAM_SPEC_SLEEP

```
#define NVSRAM_SPEC_SLEEP 0xB9
```

睡眠模式使能

4.7.2.13 NVSRAM_SPEC_WRSN

```
#define NVSRAM_SPEC_WRSN 0xC2
```

写序列号

4.7.2.14 NVSRAM_SRAM_FREAD

```
#define NVSRAM_SRAM_FREAD 0x0B
```

快读内存

4.7.2.15 NVSRAM_SRAM_READ

```
#define NVSRAM_SRAM_READ 0x03
```

读内存

4.7.2.16 NVSRAM_SRAM_WRITE

```
#define NVSRAM_SRAM_WRITE 0x02
```

写内存

4.7.2.17 NVSRAM_STATUS_BP0

```
#define NVSRAM_STATUS_BP0 0x04
```

状态寄存器 块保护 0位

4.7.2.18 NVSRAM_STATUS_BP1

```
#define NVSRAM_STATUS_BP1 0x08
```

状态寄存器 块保护 1位

4.7.2.19 NVSRAM_STATUS_FRDSR

```
#define NVSRAM_STATUS_FRDSR 0x0A
```

快读状态寄存器

4.7.2.20 NVSRAM_STATUS_RDSR

```
#define NVSRAM_STATUS_RDSR 0x05
```

读状态寄存器

4.7.2.21 NVSRAM_STATUS_RDY

```
#define NVSRAM_STATUS_RDY 0x01
```

状态寄存器 就绪位

4.7.2.22 NVSRAM_STATUS_SNL

```
#define NVSRAM_STATUS_SNL 0x40
```

状态寄存器 序列号锁存位

4.7.2.23 NVSRAM_STATUS_WEN

```
#define NVSRAM_STATUS_WEN 0x02
```

状态寄存器 写使能位

4.7.2.24 NVSRAM_STATUS_WPEN

```
#define NVSRAM_STATUS_WPEN 0x80
```

状态寄存器 写保护引脚使能位

4.7.2.25 NVSRAM_STATUS_WRDI

```
#define NVSRAM_STATUS_WRDI 0x04
```

复位写使能

4.7.2.26 NVSRAM_STATUS_WREN

```
#define NVSRAM_STATUS_WREN 0x06
```

写使能

4.7.2.27 NVSRAM_STATUS_WRSR

```
#define NVSRAM_STATUS_WRSR 0x01
```

写转态寄存器

4.8 NVSRAM静态函数

函数

- static int8_t **nvsram_check_id** (void)
  NVSRAM 检测设备*ID*
- static int8_t **nvsram_get_ready** (void)
  NVSRAM 获取就绪状态（上电等待）
- static int8_t **nvsram_get_snw** (void)
  NVSRAM 序列号是否写入*(*上电检测*)*
- static int8_t **nvsram_verify_set_wen** (void)
  NVSRAM 设置写使能

4.8.1 详细描述

4.8.2 函数说明

4.8.2.1 nvsram_check_id()

```
static int8_t nvsram_check_id (
             void  )  [static]
```

NVSRAM 检测设备ID

返回值

*0:*成功	
*-1:*失败	

4.8.2.2 nvsram_get_ready()

```
static int8_t nvsram_get_ready (
             void  )  [static]
```

NVSRAM 获取就绪状态（上电等待）

返回值

*0:*成功	
*-1:*失败	

4.8.2.3 nvsram_get_snw()

```
static int8_t nvsram_get_snw (
             void  )  [static]
```

NVSRAM 序列号是否写入(上电检测)

返回值

*0:*成功	
*-1:*失败	

4.8.2.4 nvsram_verify_set_wen()

```
static int8_t nvsram_verify_set_wen (
             void  )  [static]
```

NVSRAM 设置写使能

注解

每次写操作之前都要调用
状态寄存器两个方面判断 : 设备就绪位为0 写使能位为1

返回值

*0:*成功	
*-1:*失败	

Chapter 5

结构体说明

5.1 Nvsram_Struct_t结构体 参考

NVSRAM结构体

`#include <Drv_Nvsram.h>`

成员变量

- uint8_t **error**
- uint8_t **enabled**
- uint32_t **id**
- uint8_t **snw**

5.1.1 详细描述

NVSRAM结构体

5.1.2 结构体成员变量说明

5.1.2.1 enabled

`uint8_t enabled`

设备使能

5.1.2.2 error

`uint8_t error`

设备不正常

5.1.2.3 id

```
uint32_t id
```

设备ID

5.1.2.4 snw

```
uint8_t snw
```

是否写入设备序列号

该结构体的文档由以下文件生成:

- E:/STM32SVN/Doxy实验/NVSRAM驱动/ **Drv_Nvsram.h**

Chapter 6

文件说明

6.1 E:/STM32SVN/Doxy实验/NVSRAM驱动/Drv_Nvsram.c 文件参考

非易失性RAM驱动.

```
#include "Drv_Nvsram.h"
#include "Drv_Spi.h"
#include <stdio.h>
#include <string.h>
```

宏定义

- #define **Nvsram_Delay_Ms**(NMS) HAL_Delay(NMS)
  NVSRAM 毫秒延时.
- #define **Nvsram_Get_tick**() HAL_GetTick()
  NVSRAM 获取系统时间戳.
- #define **Nvsram_Spi_TR**(txbuf, rxbuf, len) SPI_TransmitReceive(SPI_CH1, txbuf, rxbuf, len)
  *NVSRAM SPI*通信接口
- #define **NVSRAM_STATUS_RDSR** 0x05
- #define **NVSRAM_STATUS_FRDSR** 0x0A
- #define **NVSRAM_STATUS_WRSR** 0x01
- #define **NVSRAM_STATUS_WREN** 0x06
- #define **NVSRAM_STATUS_WRDI** 0x04
- #define **NVSRAM_SRAM_READ** 0x03
- #define **NVSRAM_SRAM_FREAD** 0x0B
- #define **NVSRAM_SRAM_WRITE** 0x02
- #define **NVSRAM_SPEC_NV_STORE** 0x3C
- #define **NVSRAM_SPEC_NV_RECALL** 0x60
- #define **NVSRAM_SPEC_NV_ASENB** 0x59
- #define **NVSRAM_SPEC_NV_ASDISB** 0x19
- #define **NVSRAM_SPEC_SLEEP** 0xB9
- #define **NVSRAM_SPEC_WRSN** 0xC2
- #define **NVSRAM_SPEC_RDSN** 0xC3
- #define **NVSRAM_SPEC_FRDSN** 0xC9
- #define **NVSRAM_SPEC_RDID** 0x9F
- #define **NVSRAM_SPEC_FRDID** 0x99

- #define **NVSRAM_STATUS_RDY** 0x01
- #define **NVSRAM_STATUS_WEN** 0x02
- #define **NVSRAM_STATUS_BP0** 0x04
- #define **NVSRAM_STATUS_BP1** 0x08
- #define **NVSRAM_STATUS_SNL** 0x40
- #define **NVSRAM_STATUS_WPEN** 0x80
- #define **NVSRAM_DEV_ID** 0x06818820
- #define **NVSRAM_MAX_ADDR** 0x1FFFF
- #define **NVSRAM_BUF_SIZE** 256

函数

- static int8_t **nvsram_check_id** (void)
  NVSRAM 检测设备*ID*
- static int8_t **nvsram_get_ready** (void)
  NVSRAM 获取就绪状态（上电等待）
- static int8_t **nvsram_get_snw** (void)
  NVSRAM 序列号是否写入*(*上电检测*)*
- static int8_t **nvsram_verify_set_wen** (void)
  NVSRAM 设置写使能
- uint8_t **Nvsram_Set_Bp** (uint8_t Level)
  NVSRAM 设置写保护.
- uint8_t **Nvsram_Get_Bp** (void)
  NVSRAM 获取写保护块.
- uint32_t **Nvsram_Burst_Write_Buff** (uint32_t nvsram_addr, uint8_t ∗w_buf, uint32_t len)
  NVSRAM 写内存函数
- uint32_t **Nvsram_Burst_Read_Buff** (uint32_t nvsram_addr, uint8_t ∗r_buf, uint32_t len)
  NVSRAM 读内存函数
- uint8_t **Nvsram_Soft_Store** (void)
  NVSRAM 软件存储
- uint8_t **Nvsram_Soft_Recall** (void)
  NVSRAM 软件加载数据
- uint8_t **Nvsram_Set_SN** (uint8_t ∗sn_buff)
  NVSRAM 设置序列号
- void **Nvsram_Get_SN** (uint8_t ∗sn_buff)
  NVSRAM 获取序列号
- uint8_t **Nvsram_Init** (void)
  NVSRAM 设备初始化
- void **Nvsram_Get_Struct** (**Nvsram_Struct_t** ∗nvsram_st)
  NVSRAM 获取设备状态
- void **Nvsram_Test** (void)
  NVSRAM 驱动测试

变量

- static **Nvsram_Struct_t** **_nvsram** = {0}
- static uint8_t **_nvsram_txbuf** [**NVSRAM_BUF_SIZE**+4] = {0}
- static uint8_t **_nvsram_rxbuf** [**NVSRAM_BUF_SIZE**+4] = {0}
- uint8_t **test_tx** [**NVSRAM_BUF_SIZE**] = {0}
- uint8_t **test_rx** [**NVSRAM_BUF_SIZE**] = {0}

6.1.1 详细描述

非易失性RAM驱动.

作者

YZH

```
  ==========================================================================
                    ##### 芯片介绍 #####
  ==========================================================================
[..]
   (+) 芯片型号    :  CY14B101Q2A-SXI
   (+) 内存        :  128K * 8 bit
   (+) 地址        :  0x00000-0x1FFFF
   (+) 工作电压    :  3V
   (+) 工作温度    : -40~85°C
   (+) 设备ID      :  0x06818820
   (+) 数据保存时间（环境温度85°C） :  20年
   (+) 通信方式    :  SPI 模式0或模式3
   (+) 时钟频率    :  正常读写 最高40MHZ  快读方式最高 104MHZ
   (+) 写保护      :  支持 1/4 1/2 1/1 0 内存保护
   (+) 工作方式    :  其内部有两个模块组成 SRAM （掉电丢失）和 量子陷阱（掉电保存）
                      正常读写在SRAM中，掉电的过程中，外部连接电容剩余的电量，将SRAM
                      中的数据保存到量子陷阱单元，实现掉电数据保护，上电之后，会自动
                      将量子陷阱单元的数据搬运到SRAM中，实现上电加载，同时也可以利用
                      软件命令实现数据保存。
```

注意

(+) 数据进行突发写操作时，当突发写到达一个受保护的块地址时，它将地址增量继续到受保护空间，但是不向受保护内存写入任何数据。如果地址滚转将突发写操作带到未受保护的空间，然后恢复写入。
(+) 数据进行突发写操作或者读操作时，如果地址大于0x1FFFF，地址将自动在0x00000处依次增加写入或者读取;
(+) 在进行写内存，写寄存器，或者保存，加载、自动保存使能，自动保存不使能操作之后，写使能将被禁止，同时状态寄存中WEN bit为0，下一次进行写操作时必须将开启写使能。
(+) 芯片内部有8个字节的序列号空间，但只能被写一次（保存到量子陷阱之后），写一次之后，状态寄存器中SNL bit 为1，并且永远不能擦除该标志位，序列号永远不能再被更改。
(+) 提高设备驱动读写速率有两个方法 : 提高SPI通信速率
增大NVSRAM_BUF_SIZE缓存区

6.2 E:/STM32SVN/Doxy实验/NVSRAM驱动/Drv_Nvsram.h 文件参考

非易失性RAM驱动.

```
#include "stm32h7xx.h"
#include "Data_Type.h"
```

结构体

- struct **Nvsram_Struct_t**

  *NVSRAM*结构体

宏定义

- #define **Drv_Nvsram_TSET**

枚举

- enum **NVSRAM_BPL** { **NVSRAM_BPL0** = 0x00 , **NVSRAM_BPL1** , **NVSRAM_BPL2** , **NVSRAM_BPL3** }
  内存写保护等级
- enum **NVSRAM_RET** { **NVSRAM_SUCCEED** , **NVSRAM_PRAM_ERRO** , **NVSRAM_Busy** }
  *NVSRAM*返回值说明

函数

- uint8_t **Nvsram_Init** (void)
  NVSRAM 设备初始化
- uint8_t **Nvsram_Set_Bp** (uint8_t Level)
  NVSRAM 设置写保护.
- uint8_t **Nvsram_Get_Bp** (void)
  NVSRAM 获取写保护块.
- uint32_t **Nvsram_Burst_Write_Buff** (uint32_t nvsram_addr, uint8_t *w_buf, uint32_t len)
  NVSRAM 写内存函数
- uint32_t **Nvsram_Burst_Read_Buff** (uint32_t nvsram_addr, uint8_t *r_buf, uint32_t len)
  NVSRAM 读内存函数
- uint8_t **Nvsram_Soft_Store** (void)
  NVSRAM 软件存储
- uint8_t **Nvsram_Soft_Recall** (void)
  NVSRAM 软件加载数据
- uint8_t **Nvsram_Set_SN** (uint8_t *sn_buff)
  NVSRAM 设置序列号
- void **Nvsram_Get_SN** (uint8_t *sn_buff)
  NVSRAM 获取序列号
- void **Nvsram_Get_Struct** (**Nvsram_Struct_t** *nvsram_st)
  NVSRAM 获取设备状态
- void **Nvsram_Test** (void)
  NVSRAM 驱动测试

6.2.1 详细描述

非易失性RAM驱动.

作者

YZH

6.3 Drv_Nvsram.h

浏览该文件的文档.

```
00001
00008 #ifndef _DRV_NVSRAM_H
00009 #define _DRV_NVSRAM_H
00010
00011 #ifdef __cplusplus
00012 extern "C" {
00013 #endif
00014 /* Includes ------------------------------------------------------------------*/
00015 #include "stm32h7xx.h"
00016 #include "Data_Type.h"
00017
00018
00027 /* Exported types ------------------------------------------------------------*/
00035  typedef enum
00036  {
00037
00038    NVSRAM_BPL0 = 0x00,
00039    NVSRAM_BPL1,
00040    NVSRAM_BPL2,
00041    NVSRAM_BPL3,
00043  }NVSRAM_BPL;
00044
00048  typedef enum
00049  {
00050
00051    NVSRAM_SUCCEED,
00052    NVSRAM_PRAM_ERRO,
00053    NVSRAM_Busy,
00055  }NVSRAM_RET;
00056
00060  typedef struct
00061 {
00062   uint8_t error;
00063   uint8_t enabled;
00064   uint32_t id;
00065   uint8_t snw;
00066 } Nvsram_Struct_t;
00067
00075 #define Drv_Nvsram_TSET
00081  /* Exported function ---------------------------------------------------------*/
00085 uint8_t Nvsram_Init(void);
00086 uint8_t Nvsram_Set_Bp(uint8_t Level);
00087 uint8_t Nvsram_Get_Bp(void);
00088 uint32_t Nvsram_Burst_Write_Buff(uint32_t nvsram_addr, uint8_t *w_buf, uint32_t len);
00089 uint32_t Nvsram_Burst_Read_Buff(uint32_t nvsram_addr, uint8_t *r_buf, uint32_t len);
00090 uint8_t Nvsram_Soft_Store(void);
00091 uint8_t Nvsram_Soft_Recall(void);
00092 uint8_t Nvsram_Set_SN(uint8_t *sn_buff);
00093 void Nvsram_Get_SN(uint8_t *sn_buff);
00094 void Nvsram_Get_Struct(Nvsram_Struct_t *nvsram_st);
00095
00096 #ifdef Drv_Nvsram_TSET
00097   void Nvsram_Test(void);
00098 #endif
00113 #ifdef __cplusplus
00114 }
00115 #endif
00116
00117 #endif
00118
00119 /*****************************END OF FILE**************************************/
00120
00121
00122
00123
00124
00125
00126
00127
00128
00129
00130
00131
00132
00133
00134
00135
00136
00137
00138
00139
```

Index

_nvsram
NVSRAM驱动, 7
_nvsram_rxbuf
NVSRAM驱动, 7
_nvsram_txbuf
NVSRAM驱动, 8

Drv_Nvsram_TSET
NVSRAM外部宏, 9

E:/STM32SVN/Doxy实验/NVSRAM驱动/Drv_Nvsram.c, 27
E:/STM32SVN/Doxy实验/NVSRAM驱动/Drv_Nvsram.h, 29, 31
enabled
Nvsram_Struct_t, 25
error
Nvsram_Struct_t, 25

id
Nvsram_Struct_t, 25

NVSRAM_BPL
NVSRAM外部枚举与结构体类型, 8
NVSRAM_BPL0
NVSRAM外部枚举与结构体类型, 8
NVSRAM_BPL1
NVSRAM外部枚举与结构体类型, 8
NVSRAM_BPL2
NVSRAM外部枚举与结构体类型, 8
NVSRAM_BPL3
NVSRAM外部枚举与结构体类型, 8
NVSRAM_BUF_SIZE
NVSRAM设备参数, 18
Nvsram_Burst_Read_Buff
NVSRAM外部函数, 10
Nvsram_Burst_Write_Buff
NVSRAM外部函数, 11
NVSRAM_Busy
NVSRAM外部枚举与结构体类型, 9
nvsram_check_id
NVSRAM静态函数, 22
Nvsram_Delay_Ms
NVSRAM静态函数宏, 16
NVSRAM_DEV_ID
NVSRAM设备参数, 18
Nvsram_Get_Bp
NVSRAM外部函数, 11
nvsram_get_ready
NVSRAM静态函数, 22
Nvsram_Get_SN
NVSRAM外部函数, 12
nvsram_get_snw
NVSRAM静态函数, 22
Nvsram_Get_Struct
NVSRAM外部函数, 12
Nvsram_Get_tick
NVSRAM静态函数宏, 16
Nvsram_Init
NVSRAM外部函数, 12
NVSRAM_MAX_ADDR
NVSRAM设备参数, 18
NVSRAM_PRAM_ERRO
NVSRAM外部枚举与结构体类型, 9
NVSRAM_RET
NVSRAM外部枚举与结构体类型, 9
Nvsram_Set_Bp
NVSRAM外部函数, 13
Nvsram_Set_SN
NVSRAM外部函数, 13
Nvsram_Soft_Recall
NVSRAM外部函数, 14
Nvsram_Soft_Store
NVSRAM外部函数, 14
NVSRAM_SPEC_FRDID
NVSRAM设备参数, 18
NVSRAM_SPEC_FRDSN
NVSRAM设备参数, 18
NVSRAM_SPEC_NV_ASDISB
NVSRAM设备参数, 19
NVSRAM_SPEC_NV_ASENB
NVSRAM设备参数, 19
NVSRAM_SPEC_NV_RECALL
NVSRAM设备参数, 19
NVSRAM_SPEC_NV_STORE
NVSRAM设备参数, 19
NVSRAM_SPEC_RDID
NVSRAM设备参数, 19
NVSRAM_SPEC_RDSN
NVSRAM设备参数, 19
NVSRAM_SPEC_SLEEP
NVSRAM设备参数, 19
NVSRAM_SPEC_WRSN
NVSRAM设备参数, 19
Nvsram_Spi_TR
NVSRAM静态函数宏, 17
NVSRAM_SRAM_FREAD
NVSRAM设备参数, 20
NVSRAM_SRAM_READ

NVSRAM设备参数, 20
NVSRAM_SRAM_WRITE
NVSRAM设备参数, 20
NVSRAM_STATUS_BP0
NVSRAM设备参数, 20
NVSRAM_STATUS_BP1
NVSRAM设备参数, 20
NVSRAM_STATUS_FRDSR
NVSRAM设备参数, 20
NVSRAM_STATUS_RDSR
NVSRAM设备参数, 20
NVSRAM_STATUS_RDY
NVSRAM设备参数, 20
NVSRAM_STATUS_SNL
NVSRAM设备参数, 21
NVSRAM_STATUS_WEN
NVSRAM设备参数, 21
NVSRAM_STATUS_WPEN
NVSRAM设备参数, 21
NVSRAM_STATUS_WRDI
NVSRAM设备参数, 21
NVSRAM_STATUS_WREN
NVSRAM设备参数, 21
NVSRAM_STATUS_WRSR
NVSRAM设备参数, 21
Nvsram_Struct_t, 25
enabled, 25
error, 25
id, 25
snw, 26
NVSRAM_SUCCEED
NVSRAM外部枚举与结构体类型, 9
Nvsram_Test
NVSRAM外部函数, 15
nvsram_verify_set_wen
NVSRAM静态函数, 23
NVSRAM外部函数, 9
Nvsram_Burst_Read_Buff, 10
Nvsram_Burst_Write_Buff, 11
Nvsram_Get_Bp, 11
Nvsram_Get_SN, 12
Nvsram_Get_Struct, 12
Nvsram_Init, 12
Nvsram_Set_Bp, 13
Nvsram_Set_SN, 13
Nvsram_Soft_Recall, 14
Nvsram_Soft_Store, 14
Nvsram_Test, 15
test_rx, 15
test_tx, 15
NVSRAM外部宏, 9
Drv_Nvsram_TSET, 9
NVSRAM外部枚举与结构体类型, 8
NVSRAM_BPL, 8
NVSRAM_BPL0, 8
NVSRAM_BPL1, 8
NVSRAM_BPL2, 8
NVSRAM_BPL3, 8
NVSRAM_Busy, 9
NVSRAM_PRAM_ERRO, 9
NVSRAM_RET, 9
NVSRAM_SUCCEED, 9
NVSRAM设备参数, 17
NVSRAM_BUF_SIZE, 18
NVSRAM_DEV_ID, 18
NVSRAM_MAX_ADDR, 18
NVSRAM_SPEC_FRDID, 18
NVSRAM_SPEC_FRDSN, 18
NVSRAM_SPEC_NV_ASDISB, 19
NVSRAM_SPEC_NV_ASENB, 19
NVSRAM_SPEC_NV_RECALL, 19
NVSRAM_SPEC_NV_STORE, 19
NVSRAM_SPEC_RDID, 19
NVSRAM_SPEC_RDSN, 19
NVSRAM_SPEC_SLEEP, 19
NVSRAM_SPEC_WRSN, 19
NVSRAM_SRAM_FREAD, 20
NVSRAM_SRAM_READ, 20
NVSRAM_SRAM_WRITE, 20
NVSRAM_STATUS_BP0, 20
NVSRAM_STATUS_BP1, 20
NVSRAM_STATUS_FRDSR, 20
NVSRAM_STATUS_RDSR, 20
NVSRAM_STATUS_RDY, 20
NVSRAM_STATUS_SNL, 21
NVSRAM_STATUS_WEN, 21
NVSRAM_STATUS_WPEN, 21
NVSRAM_STATUS_WRDI, 21
NVSRAM_STATUS_WREN, 21
NVSRAM_STATUS_WRSR, 21
NVSRAM静态函数, 22
nvsram_check_id, 22
nvsram_get_ready, 22
nvsram_get_snw, 22
nvsram_verify_set_wen, 23
NVSRAM静态函数宏, 16
Nvsram_Delay_Ms, 16
Nvsram_Get_tick, 16
Nvsram_Spi_TR, 17
NVSRAM驱动, 7
_nvsram, 7
_nvsram_rxbuf, 7
_nvsram_txbuf, 8

snw
Nvsram_Struct_t, 26

test_rx
NVSRAM外部函数, 15
test_tx
NVSRAM外部函数, 15

设备驱动, 16