AN2701

应用笔记

UM2011A API 使用指南

版本：V1.0

UNICMICRO

广芯微电子

广芯微电子（广州）股份有限公司

http://www.unicmicro.com/

条款协议

本文档的所有部分，其著作产权归广芯微电子（广州）股份有限公司（以下简称广芯微电子）所有，未经广芯微电子授权许可，任何个人及组织不得复制、转载、仿制本文档的全部或部分组件。本文档没有任何形式的担保、立场表达或其他暗示，若有任何因本文档或其中提及的产品所有资讯所引起的直接或间接损失，广芯微电子及所属员工恕不为其担保任何责任。除此以外，本文档所提到的产品规格及资讯仅供参考，内容亦会随时更新，恕不另行通知。

1. 本文档中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文档中的电路、软件和相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，广芯微电子不承担任何责任。
2. 在准备本文档所记载的信息的过程中，广芯微电子已尽量做到合理注意，但是，广芯微电子并不保证这些信息都是准确无误的。用户因本文档中所记载的信息的错误或遗漏而遭受的任何损失，广芯微电子不承担任何责任。
3. 对于因使用本文档中的广芯微电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，广芯微电子不承担任何责任。本文档所记载的内容不应视为对广芯微电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 使用本文档中记载的广芯微电子产品时，应在广芯微电子指定的范围内，特别是在最大额定值、电源工作电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用广芯微电子产品而产生的故障或损失，广芯微电子不承担任何责任。
5. 虽然广芯微电子一直致力于提高广芯微电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，广芯微电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当广芯微电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。

目录

1 UM2011A 移植

UM2011A 的示例代码已封装相关驱动接口，在移植过程中仅需要移植实现 um2011A_hal 硬件接口，其他文件无需修改。

注：所有的示例代码下的 UM2011A 驱动文件接口都一样，仅可能存在 rfconfig.h 不一样，rfconfig.h 文件可由 RFOCT 上位机导出，可直接替换。

1.1 文件准备

1. 公共文件

路径：/ COMMON/type.h

描述：type.h 文件为公共定义类型，UM2011A 中的驱动使用到相关的定义

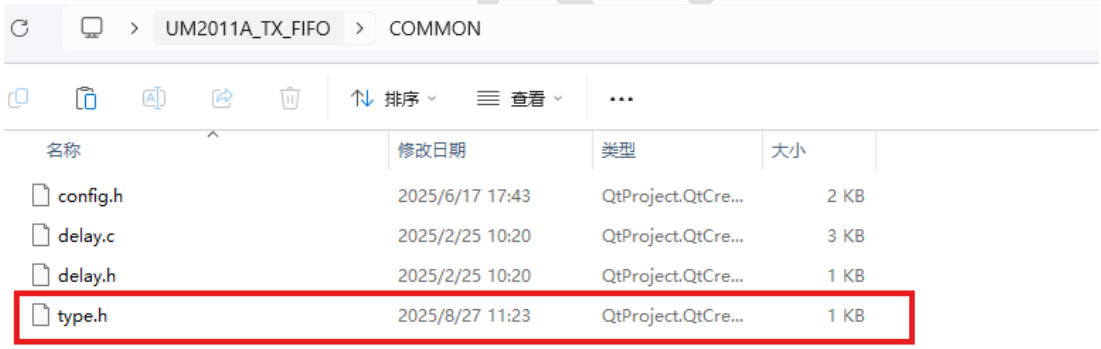

图 1-1：公共定义类型文件

2. UM2011A 驱动文件

路径：/ UM2011A/*

描述：UM2011A 文件夹包含以下文件：

- radio.h ：包括初始化、发射数据包、接收数据包（Mode2）
- radio.c ：实现初始化、发射数据包、接收数据包（Mode2）
- um2011A.h ：UM2011A 驱动文件接口
- um2011A.c ：实现 UM2011A 驱动

- rfconfig.h ：配置文件，修改配置直接替换，由上位机 RFOCT 导出
- um2011A_defs.h ：UM2011A 的宏定义
- um2011A_hal.h ：UM2011A 硬件接口
- um2011A_hal.c ：UM2011A 硬件接口实现

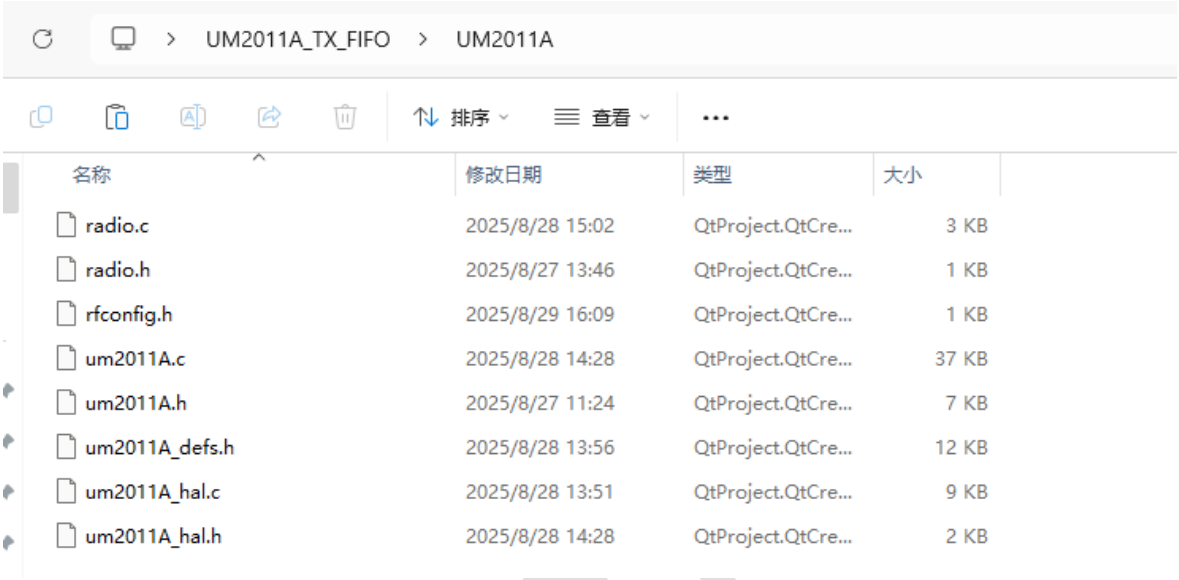

图 1-2：UM2011A 驱动文件夹的文件

1.2 硬件接口

1.2.1 um2011A_hal 介绍

移植文件（um2011A_hal）的硬件接口介绍：

1. void um2011A_hal_init(void)

   硬件初始化，实现以下功能：

   - 实现配置中断输出连接的 GPIO 初始化，并使能中断，采用单边沿的上升沿中断。
   - 实现控制 SDN 的 GPIO 的初始化功能，并拉低 SDN 启动让芯片处于工作中。

2. void um2011A_hal_set_sdn(em_level_t level)

   SDN 控制，实现以下功能：

   - 实现 SDN 的拉高关断芯片和拉低使芯片正常工作。

3. void um2011A_delay_us(uint16_t us);

   UM2011A 延时，实现以下功能：

   - 延迟函数，实现μs 延时。

4. void spi_init(void)

   SPI 初始化，实现以下功能：

   - SPI 的 IO 的硬件初始化，采用模拟三线 SPI，需初始化与芯片 CS、CLK、SDA 连接 IO 的初始化。

5. void spi_cs_enable(void)

   SPI CS 使能，实现以下功能：

   - 实现 CS 的拉低功能。

6. void spi_cs_disable(void)

   SPI CS 不使能，实现以下功能：

   - 实现 CS 的拉高功能。

7. void spi_write_byte(uint8_t byte)

   SPI 写一个字节，实现以下功能：

   - 实现 SPI 时序写一个字节。

8. uint8_t spi_read_byte(void)

   SPI 读一个字节，实现以下功能：

   - 实现 SPI 时序读一个字节。

1.2.2 um2011A_hal 移植

UM2011A 的移植仅需要移植 um2011A_hal 文件，um2011A_hal 移植过程中仅需要实现以下功能：

1. 宏定义

```
#define um2011A_HAL_SDN_LOW                    // 拉低 SDN 电平
```



```
#define um2011A_HAL_SDN_HIGH                    // 拉高 SDN 电平
#define um2011A_HAL_SPI_CS_ENABLE               // 拉低 CS 电平
#define um2011A_HAL_SPI_CS_DISABLE              // 拉高 CS 电平
#define um2011A_HAL_SPI_CLK_LOW                 // 拉低 CLK 电平
#define um2011A_HAL_SPI_CLK_HIGH                // 拉高 CLK 电平
#define um2011A_HAL_SPI_SDA_INPUT               // SDA 设置为输入
#define um2011A_HAL_SPI_SDA_OUTPUT              // SDA 设置为输出
#define um2011A_HAL_SPI_SDA_LOW                 // 拉低 SDA 电平
#define um2011A_HAL_SPI_SDA_HIGH                // 拉高 SDA 电平
#define um2011A_HAL_SPI_SDA_GET                 // 获取 SDA 电平
#define um2011A_DELAY_US(us)                    // 延迟（μs）μs
```

2. 硬件初始化

```
void um2011A_hal_init(void)
{
    /* 发射完成或接收完成中断的 GPIO 设置为输入下拉，单边上升沿，中断调用
um2011A_hal_irq */
    /* SDN 设置为输出，并输出低电平，等待 2ms 使芯片上电稳定 */
}
```

3. SPI 初始化

```
void spi_init(void)
{
    /* CS 输出，默认输出高电平 */
    /* CLK 输出，默认输出低电平 */
    /* SDA 输出，使能输入，默认输出高电平 */
}
```

1.3 TX 流程

UM2011A 发射数据流程：

- radio_init 初始化
- radio_send_data 发射数据

详细流程图如下：

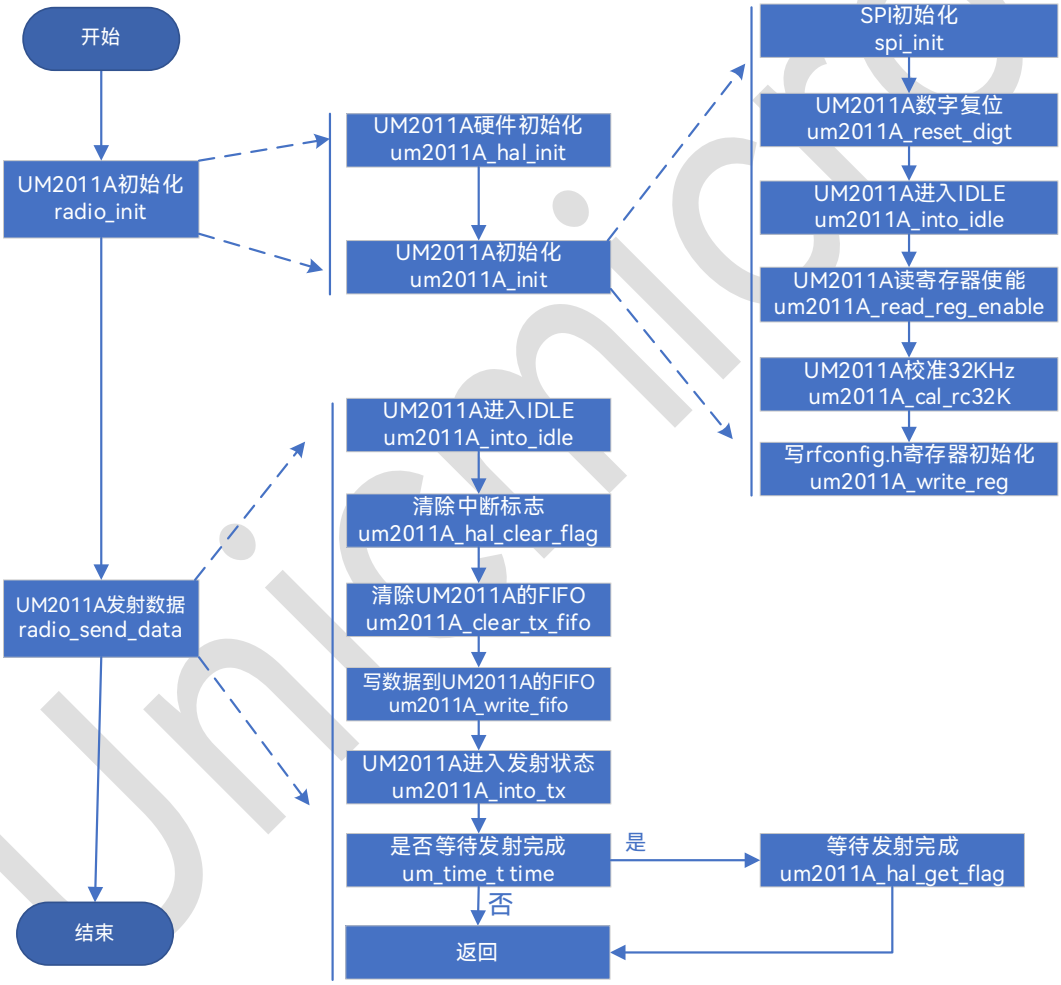

图 1-3：发射数据流程图

1.4 RX 流程

UM2011A 接收数据流程：

- radio_init 初始化
- um2011A_into_rx 进入接收
- radio_recv_data 接收数据

详细接收流程图如下:

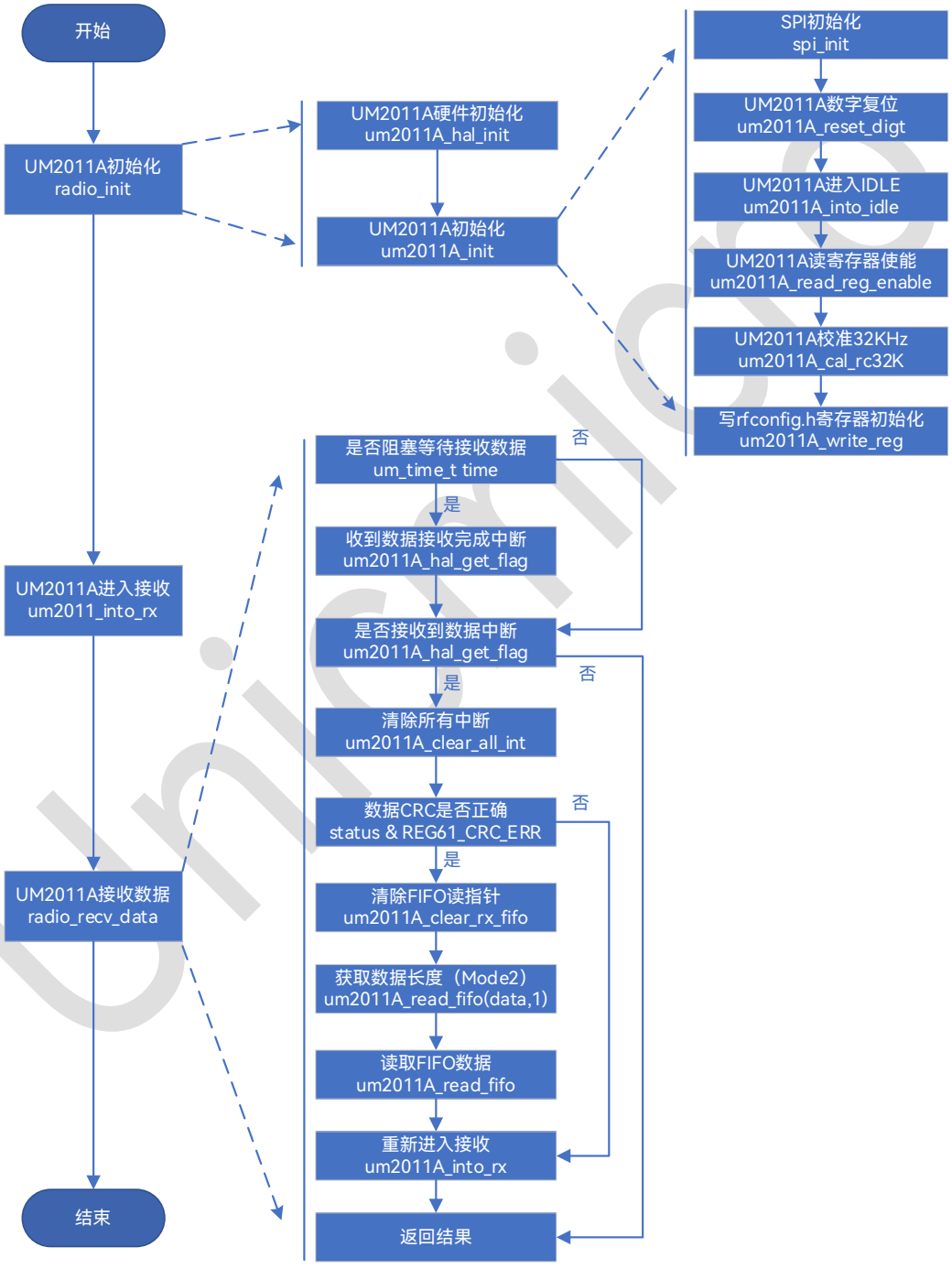

图 1-4：接收数据流程图

2 Radio 接口定义

#include “radio.h”

2.1 radio_init

函数功能	radio_init
函数描述	Radio 初始化
函数定义	em_ret_t radio_init(void);
输入参数	None
输出参数	None
函数返回	em_ret_t 返回初始化结果： ➢ SUCCESS：返回成功 ➢ FAILED：返回失败

2.2 radio_send_data

函数功能	radio_send_data
函数描述	radio 发射一包数据
函数定义	em_ret_t radio_send_data(uint8_t *data,uint8_t len,um_time_t time);
输入参数	uint8_t *data：发射的数据 uint8_t len：发射数据长度（长度不能超过 FIFO 长度，如超过 FIFO 长度需要使用 FIFO Empty 配合边发射边写入 FIFO） um_time_t time：发射是否阻塞 ➢ UM_TIME_WAITTING_FOREVER：阻塞方式 ➢ UM_TIME_WAITTING_NO：非阻塞方式
输出参数	None
函数返回	em_ret_t 返回发射数据结果： ➢ SUCCESS：返回成功 ➢ FAILED：返回失败

2.3 radio_recv_data

函数功能	radio_recv_data
函数描述	radio 接收一包数据
函数定义	em_ret_t radio_recv_data(uint8_t *data,uint8_t *len,um_time_t time);
输入参数	um_time_t time：是否阻塞接收数据 ➢ UM_TIME_WAITTING_FOREVER：阻塞方式 ➢ UM_TIME_WAITTING_NO：非阻塞方式
输出参数	uint8_t *data：接收到的数据 uint8_t *len：接收到数据的长度
函数返回	em_ret_t 返回接收数据结果： ➢ SUCCESS：返回成功 ➢ FAILED：返回失败

3 UM2011A 接口定义

#include “um2011A.h”

3.1 um2011A_init

函数功能	um2011A_init
函数描述	初始化 UM2011A，对 UM2011A 进行寄存器配置
函数定义	void um2011A_init(void)
输入参数	None
输出参数	None
函数返回	None

3.2 um2011A_write_reg

函数功能	um2011A_write_reg
函数描述	写单个寄存器
函数定义	void um2011A_write_reg(uint8_t addr,uint8_t value)
输入参数	uint8_t addr：寄存器地址 uint8_t value：寄存器值
输出参数	None
函数返回	None

3.3 um2011A_write_regs

函数功能	um2011A_write_regs
函数描述	连续写地址寄存器（连续写寄存器不允许跨页）
函数定义	void um2011A_write_regs(uint8_t addr,uint8_t *dat,uint8_t len)
输入参数	uint8_t addr：寄存器地址 uint8_t *dat：连续寄存器值 uint8_t len：连续写寄存器长度
输出参数	None
函数返回	None

3.4 um2011A_read_reg_enable

函数功能	um2011A_read_reg_enable
函数描述	SPI 读寄存器使能，寄存器需要先使能
函数定义	em_ret_t um2011A_read_reg_enable(void)
输入参数	None
输出参数	None
函数返回	em_ret_t 返回读使能结果 ➢ SUCCESS：返回成功 ➢ FAILED：返回失败

3.5 um2011A_read_reg

函数功能	um2011A_read_reg
函数描述	读寄存器
函数定义	uint8_t um2011A_read_reg(uint8_t addr)
输入参数	uint8_t addr：寄存器地址
输出参数	None
函数返回	uint8_t 返回寄存器值

3.6 um2011A_read_regs

函数功能	um2011A_read_regs
函数描述	读连续地址寄存器
函数定义	void um2011A_read_regs(uint8_t addr,uint8_t *dat,uint8_t len)
输入参数	uint8_t addr：读连续寄存器首地址（连续读地址不跨页） uint8_t len：读寄存器长度
输出参数	uint8_t *dat：连续寄存器的值
函数返回	None

3.7 um2011A_clear_tx_fifo

函数功能	um2011A_clear_tx_fifo
函数描述	清空 FIFO 写指针
函数定义	void um2011A_clear_tx_fifo(void);
输入参数	None
输出参数	None
函数返回	None

3.8 um2011A_write_fifo

函数功能	um2011A_write_fifo
函数描述	写 FIFO 数据
函数定义	void um2011A_write_fifo(uint8_t *dat,uint8_t len)
输入参数	uint8_t *dat：写 FIFO 的数据 uint8_t len：写 FIFO 的长度
输出参数	None
函数返回	None

3.9 um2011A_clear_rx_fifo

函数功能	um2011A_clear_rx_fifo
函数描述	清除 FIFO 读指针
函数定义	void um2011A_clear_rx_fifo(void)
输入参数	None
输出参数	None
函数返回	None

3.10 um2011A_read_fifo

函数功能	um2011A_read_fifo
函数描述	读 FIFO 数据

函数定义	void um2011A_read_fifo(uint8_t *dat,uint8_t len)
输入参数	uint8_t len：读 FIFO 的长度
输出参数	uint8_t *dat：读 FIFO 的数据
函数返回	None

3.11 um2011A_fifo_merge

函数功能	um2011A_fifo_merge
函数描述	RX_FIFO 和 TX_FIFO 共享使用，在 RX 状态或 TX 状态作为 128 字节 FIFO 使用
函数定义	void um2011A_fifo_merge(void)
输入参数	None
输出参数	None
函数返回	None

3.12 um2011A_fifo_notmerge

函数功能	um2011A_fifo_notmerge
函数描述	RX_FIFO 和 TX_FIFO 独立使用，分别为 64 个字节
函数定义	void um2011A_fifo_notmerge(void)
输入参数	None
输出参数	None
函数返回	None

3.13 um2011A_set_fifo_full_thres

函数功能	um2011A_set_fifo_full_thres
函数描述	FIFO 满门限，芯片接收数据写入 FIFO 时，FIFO 剩余的数据空间低于 value 个字节时产生 fifo_flag 中断
函数定义	void um2011A_set_fifo_full_thres(uint8_t value)
输入参数	uint8_t value：FIFO 剩余的数据空间
输出参数	None
函数返回	None

3.14 um2011A_set_fifo_empty_thres

函数功能	um2011A_set_fifo_empty_thres
函数描述	FIFO 空门限，芯片发射数据时，FIFO 剩余的数据低于 value 个字节时产生 fifo_flag 中断
函数定义	void um2011A_set_fifo_empty_thres(uint8_t value)
输入参数	uint8_t value：FIFO 剩余的数据空间
输出参数	None
函数返回	None

3.15 um2011A_fifo_save_data

函数功能	um2011A_fifo_save_data
函数描述	使能 UM2011A 在 sleep 状态下，FIFO 数据保存；不使能，sleep 状态下，FIFO 数据丢失
函数定义	void um2011A_fifo_save_data(em_enable_t enable)
输入参数	em_enable_t enable 使能选项： ➢ ENABLE：使能 ➢ DISABLE：失能
输出参数	None
函数返回	None

3.16 um2011A_into_sleep

函数功能	um2011A_into_sleep
函数描述	进入 Sleep 模式
函数定义	void um2011A_into_sleep(void)
输入参数	None
输出参数	None
函数返回	None

3.17 um2011A_into_idle

函数功能	um2011A_into_idle
函数描述	进入 IDLE 模式
函数定义	em_ret_t um2011A_into_idle(void)
输入参数	None
输出参数	None
函数返回	em_ret_t 返回结果： ➢ SUCCESS：返回成功 ➢ FAILED：返回失败

3.18 um2011A_into_tx

函数功能	um2011A_into_tx
函数描述	进入 TX 模式
函数定义	em_ret_t um2011A_into_tx(void)
输入参数	None
输出参数	None
函数返回	em_ret_t 返回结果： ➢ SUCCESS：返回成功 ➢ FAILED：返回失败

3.19 um2011A_into_rx

函数功能	um2011A_into_rx
函数描述	进入 RX 模式
函数定义	em_ret_t um2011A_into_rx(void)
输入参数	None
输出参数	None
函数返回	em_ret_t 返回结果： ➢ SUCCESS：返回成功 ➢ FAILED：返回失败

3.20 um2011A_into_tx_fson

函数功能	um2011A_into_tx_fson
函数描述	进入 TX FSON 模式
函数定义	em_ret_t um2011A_into_tx_fson(void)
输入参数	None
输出参数	None
函数返回	em_ret_t 返回结果： ➢ SUCCESS：返回成功 ➢ FAILED：返回失败

3.21 um2011A_into_rx_fson

函数功能	um2011A_into_rx_fson
函数描述	进入 RX FSON 模式
函数定义	em_ret_t um2011A_into_rx_fson(void)
输入参数	None
输出参数	None
函数返回	em_ret_t 返回结果： ➢ SUCCESS：返回成功 ➢ FAILED：返回失败

3.22 um2011A_cal_rc32K

函数功能	um2011A_cal_rc32K
函数描述	校准 RC32K
函数定义	em_ret_t um2011A_cal_rc32K(void)
输入参数	None
输出参数	None
函数返回	em_ret_t 返回结果： ➢ SUCCESS：返回成功 ➢ FAILED：返回失败

3.23 um2011A_into_tx_carrier

函数功能	um2011A_into_tx_carrier
函数描述	发射载波
函数定义	void um2011A_into_tx_carrier(void)
输入参数	None
输出参数	None
函数返回	None

3.24 um2011A_reset_digt

函数功能	um2011A_reset_digt
函数描述	数字复位
函数定义	void um2011A_reset_digt(void)
输入参数	None
输出参数	None
函数返回	None

3.25 um2011A_set_freq

函数功能	um2011A_set_freq
函数描述	设置频点
函数定义	void um2011A_set_freq(float freq,float xtal)
输入参数	float freq：信道频率 float xtal：晶振频率
输出参数	None
函数返回	None

3.26 um2011A_set_modulation

函数功能	um2011A_set_modulation
函数描述	设置模式
函数定义	void um2011A_set_modulation(em_mod_t mod)
输入参数	em_mod_t mod：调制模式 ➢ GFSK ➢ FSK ➢ OOK
输出参数	None
函数返回	None

3.27 um2011A_set_ch_step

函数功能	um2011A_set_ch_step
函数描述	设置信道步进
函数定义	void um2011A_set_ch_step(float step,float xtal)
输入参数	float step：信道步进，单位 MHz float xtal：晶振频率，单位 MHz
输出参数	None
函数返回	None

3.28 um2011A_set_ch_index

函数功能	um2011A_set_ch_index
函数描述	设置信道序号
函数定义	void um2011A_set_ch_index(uint8_t ch)
输入参数	uint8_t ch：信道号
输出参数	None
函数返回	None

3.29 um2011A_nirq_enable

函数功能	um2011A_nirq_enable
函数描述	设置 nIRQ 的中断使能
函数定义	void um2011A_nirq_enable(em_nirq_sel_t nirq)
输入参数	em_nirq_sel_t nirq：中断向量 ➢ RSSI_PJD_INT ：RSSI 有效中断，且可以选择组合输出，RSSI_PJD_INT_SET ➢ PREAMBLE_INT：Preamble 中断 ➢ SYNC_INT：SyncWord 中断 ➢ RX_PKT_INT：接收完成中断 ➢ TX_PKT_INT：发射完成中断 ➢ FIFO_INT：FIFO 中断 ➢ LBD_WARN_INT：LBD 报警中断
输出参数	None
函数返回	None

3.30 um2011A_set_gpio_dir

函数功能	um2011A_set_gpio_dir
函数描述	设置芯片 GPIO0、GPIO1、GPIO2 引脚输出方向
函数定义	void um2011A_set_gpio_dir(em_pin_sel_t pin, em_io_dir_t dir)
输入参数	em_pin_sel_t pin：引脚 ➢ PIN_SEL_GPIO0：GPIO0 ➢ PIN_SEL_GPIO1：GPIO1 ➢ PIN_SEL_GPIO2：GPIO2 em_io_dir_t dir：方向 ➢ RFIO_DIR_OUT：输出 ➢ RFIO_DIR_IN：输入
输出参数	None
函数返回	None

3.31 um2011A_get_gpio_dir

函数功能	um2011A_get_gpio_dir
函数描述	获取芯片 GPIO0、GPIO1、GPIO2 引脚输出方向
函数定义	em_io_dir_t um2011A_get_gpio_dir(em_pin_sel_t pin)
输入参数	em_pin_sel_t pin：引脚 ➢ PIN_SEL_GPIO0：GPIO0 ➢ PIN_SEL_GPIO1：GPIO1 ➢ PIN_SEL_GPIO2：GPIO2
输出参数	None
函数返回	em_io_dir_t dir：返回 GPIO 方向 ➢ RFIO_DIR_OUT：输出 ➢ RFIO_DIR_IN：输入

3.32 um2011A_set_gpio0_sel

函数功能	um2011A_set_gpio0_sel
函数描述	GPIO0 引脚的输出信号选择
函数定义	void um2011A_set_gpio0_sel(em_gpio_out_sel_t mode)
输入参数	em_gpio_out_sel_t mode：GPIO 输出设置 ➢ GPIO_OUT_NIRQ：中断信号，包括 rx_pkt_int、tx_pkt_int、preamble_int、syncword_int、fifo_int、vco_lock_int、rssi_pjd_int、lbd_warn_int，且每个中断分别有中断使能控制，通过使能控制可以选择一个中断，或者多个中断或运算后输出 ➢ GPIO_OUT_RX_PKT_FLAG：接收包完成标志 ➢ GPIO_OUT_PREAMBLE_FLAG：Preamble match 标志，接收 preamble 的数量达到目标后锁定，需要写寄存器清除，或重新进入接收后自动清除 ➢ GPIO_OUT_SYNSWORD_FLAG：同步字匹配标志，接收 syncword 匹配后锁定，需要写寄存器清除，或重新进入接收后自动清除 ➢ GPIO_OUT_FIFO_FLAG：RXFIFO 快满标志或 TXFIFO 快空标志 ➢ GPIO_OUT_CLKO_SEL：调试时钟输出，由寄存器 brclk_sel 选择 ➢ GPIO_OUT_TRXDATA：接收时输出解调 BIT 数据 RX DATA，发射输出调制 BIT 数据 TX DATA ➢ GPIO_OUT_VCO_LOCK_FLAG：VCO 校准异常标志 ➢ GPIO_OUT_CRC_ERROR：接收 CRC 错误标志 ➢ GPIO_OUT_RSSI_PJD_FLAG：RSSI 有效标志，由寄存器 rssi_pjd_int_sel 选择

	rx_rssi_valid、rssi_pjd_valid 或 rssi_valid 和 preamble_match 的组合输出 ➢ GPIO_OUT_LBD_WARN：LBD 报警输出，由 LBD 电路产生 ➢ GPIO_OUT_TR_SW：TX 切换标志 ➢ GPIO_OUT_TR_SW_INV：TX 切换标志反向 ➢ GPIO_OUT_RX_FIFO_WRBYTE：指示 RX FIFO 每写入一个 BYTE 的中断，脉冲信号，宽度为一个 symbol 长度 ➢ GPIO_OUT_HIGH：高电平 ➢ GPIO_OUT_LOW：低电平
输出参数	None
函数返回	None

3.33 um2011A_set_gpio1_sel

函数功能	um2011A_set_gpio1_sel
函数描述	GPIO1 引脚的输出信号选择
函数定义	void um2011A_set_gpio1_sel(em_gpio_out_sel_t mode)
输入参数	em_gpio_out_sel_t mode：GPIO 输出设置 ➢ GPIO_OUT_NIRQ：中断信号，包括 rx_pkt_int、tx_pkt_int、preamble_int、syncword_int、fifo_int、vco_lock_int、rssi_pjd_int、lbd_warn_int，且每个中断分别有中断使能控制，通过使能控制可以选择一个中断，或者多个中断或运算后输出 ➢ GPIO_OUT_RX_PKT_FLAG：接收包完成标志 ➢ GPIO_OUT_PREAMBLE_FLAG：Preamble match 标志，接收 preamble 的数量达到目标后锁定，需要写寄存器清除，或重新进入接收后自动清除 ➢ GPIO_OUT_SYNSWORD_FLAG：同步字匹配标志，接收 syncword 匹配后锁定，需要写寄存器清除，或重新进入接收后自动清除 ➢ GPIO_OUT_FIFO_FLAG：RXFIFO 快满标志或 TXFIFO 快空标志 ➢ GPIO_OUT_CLKO_SEL：调试时钟输出，由寄存器 brclk_sel 选择 ➢ GPIO_OUT_TRXDATA：接收时输出解调 BIT 数据 RX DATA，发射输出调制 BIT 数据 TX DATA ➢ GPIO_OUT_VCO_LOCK_FLAG：VCO 校准异常标志 ➢ GPIO_OUT_CRC_ERROR：接收 CRC 错误标志 ➢ GPIO_OUT_RSSI_PJD_FLAG：RSSI 有效标志，由寄存器 rssi_pjd_int_sel 选择 rx_rssi_valid、rssi_pjd_valid 或 rssi_valid 和 preamble_match 的组合输出 ➢ GPIO_OUT_LBD_WARN：LBD 报警输出，由 LBD 电路产生 ➢ GPIO_OUT_TR_SW：TX 切换标志 ➢ GPIO_OUT_TR_SW_INV：TX 切换标志反向 ➢ GPIO_OUT_RX_FIFO_WRBYTE：指示 RX FIFO 每写入一个 BYTE 的中断，脉冲

	信号，宽度为一个 symbol 长度 ➢ GPIO_OUT_HIGH：高电平 ➢ GPIO_OUT_LOW：低电平
输出参数	None
函数返回	None

3.34 um2011A_set_gpio2_sel

函数功能	um2011A_set_gpio2_sel
函数描述	GPIO2 引脚的输出信号选择
函数定义	void um2011A_set_gpio2_sel(em_gpio_out_sel_t mode)
输入参数	em_gpio_out_sel_t mode：GPIO 输出设置 ➢ GPIO_OUT_NIRQ：中断信号，包括 rx_pkt_int、tx_pkt_int、preamble_int、syncword_int、fifo_int、vco_lock_int、rssi_pjd_int、lbd_warn_int，且每个中断分别有中断使能控制，通过使能控制可以选择一个中断，或者多个中断或运算后输出 ➢ GPIO_OUT_RX_PKT_FLAG：接收包完成标志 ➢ GPIO_OUT_PREAMBLE_FLAG：Preamble match 标志，接收 preamble 的数量达到目标后锁定，需要写寄存器清除，或重新进入接收后自动清除 ➢ GPIO_OUT_SYNSWORD_FLAG：同步字匹配标志，接收 syncword 匹配后锁定，需要写寄存器清除，或重新进入接收后自动清除 ➢ GPIO_OUT_FIFO_FLAG：RXFIFO 快满标志或 TXFIFO 快空标志 ➢ GPIO_OUT_CLKO_SEL：调试时钟输出，由寄存器 brclk_sel 选择 ➢ GPIO_OUT_TRXDATA：接收时输出解调 BIT 数据 RX DATA，发射输出调制 BIT 数据 TX DATA ➢ GPIO_OUT_VCO_LOCK_FLAG：VCO 校准异常标志 ➢ GPIO_OUT_CRC_ERROR：接收 CRC 错误标志 ➢ GPIO_OUT_RSSI_PJD_FLAG：RSSI 有效标志，由寄存器 rssi_pjd_int_sel 选择 rx_rssi_valid、rssi_pjd_valid 或 rssi_valid 和 preamble_match 的组合输出 ➢ GPIO_OUT_LBD_WARN：LBD 报警输出，由 LBD 电路产生 ➢ GPIO_OUT_TR_SW：TX 切换标志 ➢ GPIO_OUT_TR_SW_INV：TX 切换标志反向 ➢ GPIO_OUT_RX_FIFO_WRBYTE：指示 RX FIFO 每写入一个 BYTE 的中断，脉冲信号，宽度为一个 symbol 长度 ➢ GPIO_OUT_HIGH：高电平 ➢ GPIO_OUT_LOW：低电平
输出参数	None
函数返回	None

3.35 um2011A_event_output

函数功能	um2011A_event_output
函数描述	输出 WOR EVENT 信号
函数定义	void um2011A_event_output(em_event_outpin_t pin)
输入参数	em_event_outpin_t pin：选择输出 WOR EVENT 引脚，优先级高于 GPIO 设置 ➢ NONE：不输出 ➢ EVENT_OUTPIN_SEL_GPIO0：GPIO0 输出 WOR EVENT 信号 ➢ EVENT_OUTPIN_SEL_GPIO1：GPIO1 输出 WOR EVENT 信号
输出参数	None
函数返回	None

3.36 um2011A_set_lbd_enable

函数功能	um2011A_set_lbd_enable
函数描述	设置 LBD 使能
函数定义	void um2011A_set_lbd_enable(em_enable_t enable)
输入参数	em_enable_t enable：使能选项 ➢ ENABLE：使能 ➢ DISABLE：不使能
输出参数	None
函数返回	None

3.37 um2011A_set_lbd_volt

函数功能	um2011A_set_lbd_volt
函数描述	设置 LBD 报警电压
函数定义	void um2011A_set_lbd_volt(float value)
输入参数	float value：报警电压，电压范围：2.0V~3.1V
输出参数	None
函数返回	None

3.38 um2011A_scan_lbd_volt

函数功能	um2011A_scan_lbd_volt
函数描述	扫描 LBD 电压一次
函数定义	void um2011A_scan_lbd_volt(void)
输入参数	None
输出参数	None
函数返回	None

3.39 um2011A_read_lbd_value

函数功能	um2011A_read_lbd_value
函数描述	读 LBD 电压值
函数定义	float um2011A_read_lbd_value(void)
输入参数	None
输出参数	None
函数返回	float：返回电源电压，电压范围 2.0~3.1

3.40 um2011A_postamble_en

函数功能	um2011A_postamble_en
函数描述	设置后导码使能
函数定义	void um2011A_postamble_en(em_enable_t enable)
输入参数	em_enable_t enable：使能选择 ➢ ENABLE：使能 ➢ DISABLE：不使能
输出参数	None
函数返回	None

3.41 um2011A_set_postamble

函数功能	um2011A_set_postamble
函数描述	设置后导码值与字节长度
函数定义	void um2011A_set_postamble(uint8_t value,uint8_t len)
输入参数	uint8_t value：发射后导码值 uint8_t len：发射后导码长度
输出参数	None
函数返回	None

3.42 um2011A_set_txpkt_num

函数功能	um2011A_set_txpkt_num
函数描述	每个发射命令重复发射的数据包数量（需要同时使能后导码）
函数定义	void um2011A_set_txpkt_num(uint8_t num)
输入参数	uint8_t num：数据包重复发射次数
输出参数	None
函数返回	None

3.43 um2011A_set_rssi_thr

函数功能	um2011A_set_rssi_thr
函数描述	设置 rssi 门限
函数定义	void um2011A_set_rssi_thr(double rssi)
输入参数	double rssi：RSSI 门限
输出参数	None
函数返回	None

3.44 um2011A_set_packet_mode

函数功能	um2011A_set_packet_mode
函数描述	设置包模式

函数定义	void um2011A_set_packet_mode(em_packet_mode_t mode)
输入参数	em_packet_mode_t mode：数据包模式 ➢ PACKET_MODE_0：模式 0 ➢ PACKET_MODE_1：模式 1 ➢ PACKET_MODE_2：模式 2 ➢ PACKET_MODE_3：模式 3
输出参数	None
函数返回	None

3.45 um2011A_set_payload_bit_order

函数功能	um2011A_set_payload_bit_order
函数描述	设置包模式下 payload 的 bit 顺序
函数定义	void um2011A_set_payload_bit_order(em_bit_order_t order)
输入参数	em_bit_order_t order：bit 顺序 ➢ BIT_LSB：从低位开始发射 ➢ BIT_MSB：从高位开始发射
输出参数	None
函数返回	None

3.46 um2011A_set_direct_tx

函数功能	um2011A_set_direct_tx
函数描述	使能直通发射并选择数据输入引脚
函数定义	void um2011A_set_direct_tx(em_enable_t enable, em_pin_sel_t pin)
输入参数	em_enable_t enable：使能选择 ➢ ENABLE：使能 ➢ DISABLE：不使能 em_pin_sel_t pin：引脚 ➢ PIN_SEL_GPIO0：GPIO0 ➢ PIN_SEL_GPIO1：GPIO1 ➢ PIN_SEL_GPIO2：GPIO2
输出参数	None
函数返回	None

3.47 um2011A_set_preamble

函数功能	um2011A_set_preamble
函数描述	设置 preamble 使能及字节长度
函数定义	void um2011A_set_preamble(em_enable_t enable,uint16_t len,em_unit_t unit)
输入参数	em_enable_t enable：使能选择 ➢ ENABLE：使能 ➢ DISABLE：不使能 uint16_t len：Preamble 长度 em_unit_t unit：Preamble 单位 ➢ UNIT_BYTE：单位 byte ➢ UNIT_BIT：单位 bit
输出参数	None
函数返回	None

3.48 um2011A_set_preamble_match_len

函数功能	um2011A_set_preamble_match_len
函数描述	设置 RX 时，preamble match 的字节长度
函数定义	void um2011A_set_preamble_match_len(uint8_t bit)
输入参数	uint8_t bit：匹配长度
输出参数	None
函数返回	None

3.49 um2011A_set_syncword_enable

函数功能	um2011A_set_syncword_enable
函数描述	设置同步字使能
函数定义	void um2011A_set_syncword_enable(em_enable_t enable)
输入参数	em_enable_t enable：使能选择 ➢ ENABLE：使能 ➢ DISABLE：不使能
输出参数	None
函数返回	None

3.50 um2011A_set_sync_bit_order

函数功能	um2011A_set_sync_bit_order
函数描述	设置同步字 bit 顺序
函数定义	void um2011A_set_sync_bit_order(em_bit_order_t order)
输入参数	em_bit_order_t order：bit 顺序 ➢ BIT_LSB：从低位开始发射 ➢ BIT_MSB：从高位开始发射
输出参数	None
函数返回	None

3.51 um2011A_set_sync_id

函数功能	um2011A_set_sync_id
函数描述	设置同步字长度及 ID
函数定义	void um2011A_set_sync_id(uint8_t* syncbuff,uint8_t sync_len)
输入参数	uint8_t* syncbuff：同步字值 uint8_t sync_len：同步字长度
输出参数	None
函数返回	None

3.52 um2011A_get_rssi

函数功能	um2011A_get_rssi
函数描述	获取 RSSI 值
函数定义	float um2011A_get_rssi(void)
输入参数	None
输出参数	None
函数返回	返回 RSSI 值

3.53 um2011A_clear_all_int

函数功能	um2011A_clear_all_int
函数描述	清除所有中断，包括 pjd_vaild、syncword_int、preamble_int、pkt_flag_int
函数定义	void um2011A_clear_all_int(void)
输入参数	None
输出参数	None
函数返回	None

3.54 um2011A_get_status

函数功能	um2011A_get_status
函数描述	返回状态,包括 fifo 空或满中断、包接收/发射完成中断、CRC error、preamble 中断、syncword 中断
函数定义	uint8_t um2011A_get_status(void)
输入参数	None
输出参数	None
函数返回	uint8_t：返回状态 0x80：SyncWord 中断 0x40：Preamble 中断 0x20：CRC 错误 0x10：接收数据包或发射数据包中断 0x08：FIFO 的空或满中断 0x04 ： RSSI_PJD 中断 0x02 ： VCO 校准失效标志 0x01 ： 包发射完成标志

4 UM2011A 硬件接口

#include “um2011A_hal.h”

4.1 um2011A_hal_init

函数功能	um2011A_hal_init
函数描述	UM2011A 硬件初始化
函数定义	void um2011A_hal_init(void)
输入参数	None
输出参数	None
函数返回	None

4.2 um2011A_hal_set_sdn

函数功能	um2011A_hal_set_sdn
函数描述	UM2011A SDN 控制电平，拉高关断芯片
函数定义	void um2011A_hal_set_sdn(em_level_t level)
输入参数	em_level_t level，SDN 输出电平： ➢ LOW：低电平 ➢ HIGH：高电平
输出参数	None
函数返回	None

4.3 um2011A_hal_get_flag

函数功能	um2011A_hal_get_flag
函数描述	获取中断标志
函数定义	uint8_t um2011A_hal_get_flag(void)
输入参数	None
输出参数	None
函数返回	返回中断标志 ➢ 0：无中断 ➢ 1：有中断产生

4.4 um2011A_hal_clear_flag

函数功能	um2011A_hal_clear_flag
函数描述	清除中断标志
函数定义	void um2011A_hal_clear_flag(void)
输入参数	None
输出参数	None
函数返回	None

4.5 spi_init

函数功能	spi_init
函数描述	SPI 初始化
函数定义	void spi_init(void)
输入参数	None
输出参数	None
函数返回	None

4.6 spi_cs_enable

函数功能	spi_cs_enable
函数描述	SPI CS 使能
函数定义	void spi_cs_enable(void);
输入参数	None
输出参数	None
函数返回	None

4.7 spi_cs_disable

函数功能	spi_cs_disable
函数描述	SPI CS 不使能
函数定义	void spi_cs_disable(void)
输入参数	None
输出参数	None
函数返回	None

4.8 spi_write_byte

函数功能	spi_write_byte
函数描述	SPI 写一个字节数据
函数定义	void spi_write_byte(uint8_t byte)
输入参数	uint8_t byte：写入的数据
输出参数	None
函数返回	None

4.9 spi_read_byte

函数功能	spi_read_byte
函数描述	SPI 读一个字节数据
函数定义	uint8_t spi_read_byte(void)
输入参数	None
输出参数	None
函数返回	SPI 读回的数据

5 版本修订

版本	日期	描述
V1.0	2025.08.15	初始版