# AN01711

## RHF0M003 Design Guide

**V2.0**

## Document information

| Info | Content |
|---|---|
| **Keywords** | *LoRa, LoRaWAN, UART, Low Power, RHF0M003* |
| **Abstract** | RHF0M003 design guide |

# Content

# Figures



# Tables

# 1 Preface

RHF0M003 is an UART interface LoRaWAN AT Modem with auto ultra-low power feature.
Below is a typical reference design using RHF0M003 to quick start with LoRaWAN Application.
Just connect the USART and NRST to their host MCU and send AT command.
Besides, Pin7 of the module could be used to show status of the modem. The LED1 would blink if uplink or downlink operates. Let it float if not used.
Pin8 of the module could be used to trigger the bootloader mode. It's useful to upgrade the FW of the AT command. Let it float if not used.

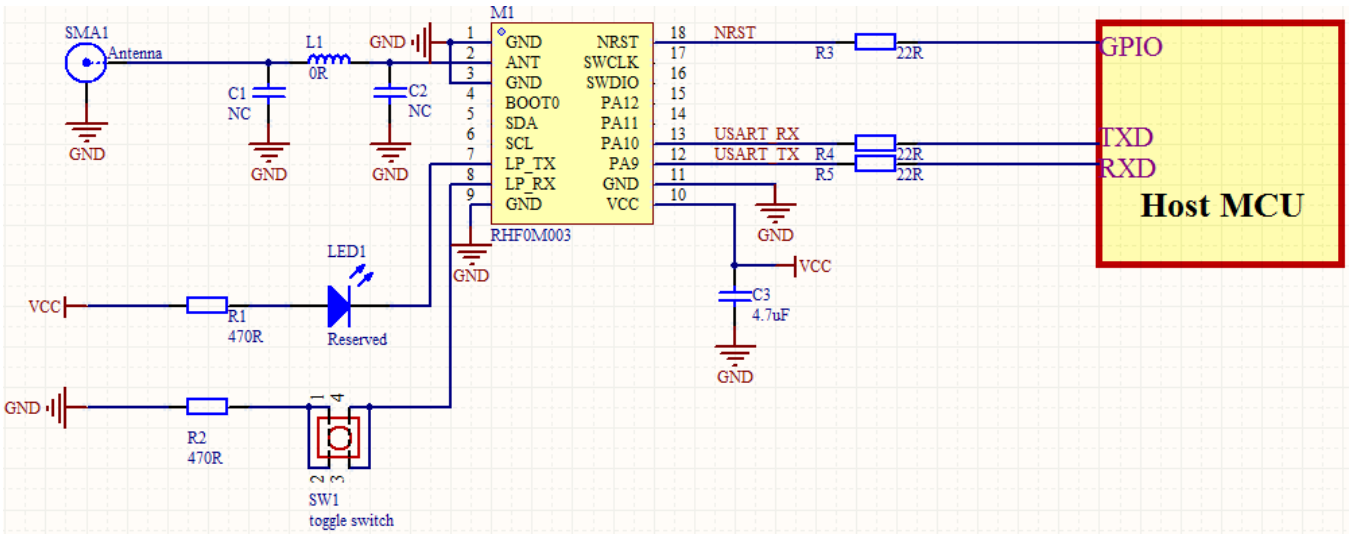



**Figure 1 RHF0M003 reference design**

# 2 Application Information

## 2.1 Pin Definition

### 2.1.1 RHF0M003 external pin definition

| Pin | RHF0M003AM | Comments |
|---|---|---|
| **1** | GND | |
| **2** | RFIO | RF input/output Port |
| **3** | GND | |
| **4** | BOOT0 | Boot mode, internal pulled low with 10kΩ by default |
| **5** | PB7/ I2C_SDA | SCL of I2C from MCU; or GPIO from MCU, PB7 |
| **6** | PB6/ I2C_SCL | SDA of I2C from MCU; or GPIO from MCU, PB6 |
| **7** | LED status/ PA2 | Indication for LoRaWan status, or PA2 |
| **8** | BLEN /PA3 | Press Low to enable bootloader during power up, or PA3 |
| **9** | GND | GND |
| **10** | VCC | Supply voltage for the module |
| **11** | GND | GND |
| **12** | UART_TX/PA9 | AT command interface UART TX (9600,8,n,1), Or UART TX (115200,8,n,1) when Bootloader mode |
| **13** | UART_RX/PA10 | AT command interface UART RX (9600,8,n,1), Or UART RX (115200,8,n,1) when Bootloader mode |
| **14** | PA11/USB_DM | GPIO from MCU, PA11 |
| **15** | PA12/USB_DP | GPIO from MCU, PA12 |
| **16** | SWDIO | SWD interface, SWDIO |
| **17** | SWCLK | SWD interface, SWCLK |
| **18** | NRST | Reset trigger input for MCU, Low to reset |

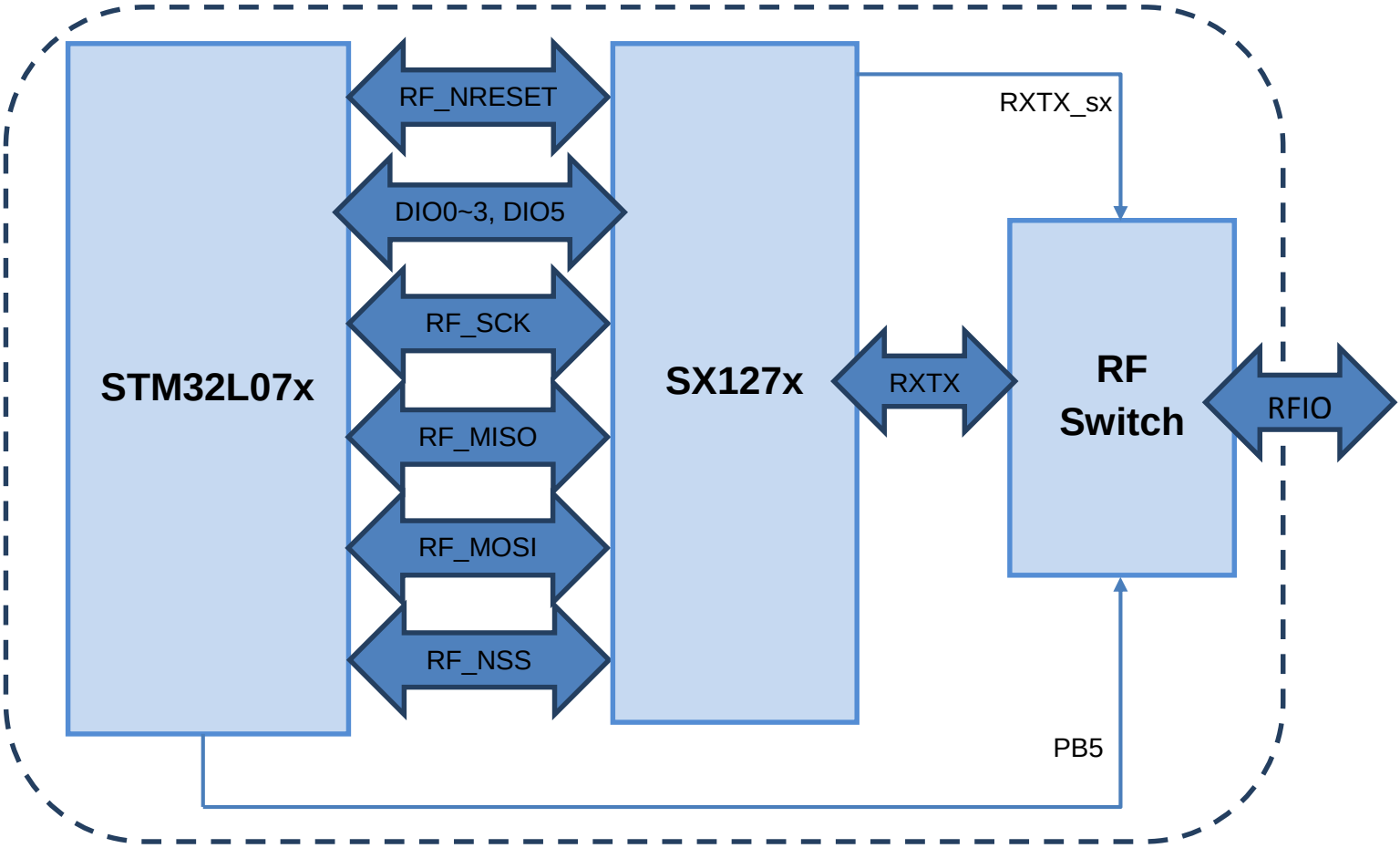
**Table 1 RHF0M003 pin definition**

### 2.1.2 RHF0M003 internal circuit

SX127x connect to STML07x as follows:

| SX127x | STM32L071 | RF Switch |
|---|---|---|
| RF_NRESET | PA-0 | |
| DIO 0 | PB-4 | |
| DIO 1 | PA-8 | |
| DIO 2 | PB-1 | |
| DIO 3 | PB-0 | |
| DIO 5 | PA-1 | |
| RF_SCK | PA-5 | |
| RF_MISO | PA-6 | |
| RF_MOSI | PA-7 | |
| RF_NSS | PA-4 | |
| RXTX/RFMOD | | RXTX_sx |
| | PB-5 | Switch_CTL |

RHF0M003 Module:



## 2.2 UART Configuration

### 2.2.1 AT Mode

Interface: Pin12-PA9 (TX) / Pin13-PA10 (RX)
Baud Rate: 9600
Data Bits: 8
Stop Bits: 1
Parity: None

### 2.2.2 Bootloader Mode

Interface: Pin12-PA9 (TX) / Pin13-PA10 (RX)
Baud Rate: 115200
Data Bits: 8
Stop Bits: 1
Parity: None

For more information about how to upgrade firmware refer to UM01518

## 2.3 Reset

If the application requires the disconnection of VCC from the RHF0M003 module or send AT+RESET command, user should wait for 2s from of the end of the POR cycle before commencing communications over the UART port. RHF0M003 use this 2s to initial itself.

Example: (AT+RESET)

```
printf("AT+RESET");
delay_ms(2000);
```

Example: (Power On)

```
// Wait until RHF0M003 is ready
delay_ms(2000);
```

### 2.3.1 NRST

Two millisecond (2ms) negative pulses on NRST pin will trigger a reset of RHF0M003 module, it performs the same as “AT+RESET”.
This pin is optional to connect with host MCU, if it is connected the pin of host MCU is expected to set to floating after reset is triggered.

## 2.4 Sleep Mode

While RHF0M003 is in sleep mode, RHF0M003 TX/RX pins status will be changed to save power. Details:

| RHF0M003 | Status |
|---|---|
| UART TX | Floating |
| UART RX | Input weak pull up |

**Table 2 RHF0M003 UART pin sleep mode status**

Host controller is expected to set HOST TX pin floating. As RHF0M003 TX pin is set to floating, so HOST RX status is not mandatory, customer could decide it on demand.

| Host MCU | Status |
|---|---|
| **UART TX** | Floating |
| **UART RX** | Depends on HOST MCU |

**Table 3 Expected Host MCU UART pin sleep mode status**

For more information about how to use auto low power mode refer to PS01509 V3.1+.

# 3 ATCTL API

RisingHF ATCTL API package is available for customers who integrate RHF0m003 in their design.

To use ATCTL software package user need implement all function in hal-atctl-template.c. And there is detailed description in hal-atctl.h to explain usage of each function and macro.

## 3.1 DBG

```
/*!
 * main.c debug log information
 */
#define DBG(x...)                               printf(x)
```

## 3.2 MUTEX

```
/*!
 * Enable protection of critical code, usually be disable global interrupt
 */
#define HAL_ATCTL_ENTER_MUTEX()                 __disable_irq()

/*!
 * End protection of critical code, recove the interrupt
 */
#define HAL_ATCTL_EXIT_MUTEX()                  __enable_irq()
```

## 3.3 Callback Function Prototype

```
/*!
 * atctl driver callback function pointer type
 */
typedef void (*atctl_cb_t)(uint8_t);
```

## 3.4 UART TX

hal_atctl_printf and hal_atctl_putchar will be called by ATCTL to send data to RHF0M003.

```
/*!
 * atctl driver call this function to send data to UART
 * printf output must be redirected to UART, refer compiler manual about how
 * to redirect printf
 */
#define hal_atctl_printf(x...)                  printf(x)

/*!
 * UART low level TX one byte function
 */
void hal_atctl_putchar(uint8_t c);
```

## 3.5 UART RX

User need implement hal_atctl_init function and supply interface for atctl driver to register atctl rx_byte function.

```
/*!
 * atctl driver HAL layer initialize, this function will register atctl callback
 * function to the UART driver, UART driver must receive all data returned by
 * RHF0M003 module
 */
void hal_atctl_init(atctl_cb_t cb);
```

The way it is called by atctl driver:

```
void atctl_init(bool wakeup)
{
    hal_atctl_init(atctl_rx_byte);

    atctl_reset();

    atctl_rx_tmp_wr_index = 0;
    atctl_rx_tmp_rd_index = 0;
    atctl_wakeup = wakeup;
}
```

## 3.6 System Time

These low level functions are mandatory for ATCTL driver.

```
/*!
 * return system tick, if the HOST MCU tick is not 32bits,
 * user should make hal_get_timeout function to adapt it.
 */
uint32_t hal_get_tick(void);

/*!
 * return difference between current time and an elapsed time
 * old_time     - an elapsed time
 */
uint32_t hal_get_timeout(uint32_t etime);

/*!
 * delay some time, unit is millisecond (ms)
 * old_time     - an elapsed time
 */
void hal_delay_ms(uint32_t ms);
```

## 3.7 Optional

```
/*!
 * HOST mcu initialize, it is optional to implement this function
 */
void hal_sys_init(void);

/*!
 * HOST mcu lowpower event, need to be polled in main loop,
 * it is optional to implement this function, user could local lowpower
 * management APIs to replace this one
 */
void hal_sys_lowpower(void);

/*!
 * This function is used to generate a periodic event, main.c will detect
 * this event, to use this event to trigger OTAA JOIN or send message periodicly
 */
bool hal_evt_flag(void);
```

# 4 FAQ

1. How to check configuration?

Each configuration except the keys (AppKey/NWKSKEY/APPSKEY) could be queried by user through query format command. Like:

```
AT+ID
AT+CH
AT+RXWIN1
AT+RXWIn2
AT+DR
AT+LW=ULDL                // Check uplink and downlink counter
AT+UART=TIEMOUT
```

# Revision

V2.0 2018-05-30
+ Update reference design and correct some error description.

V1.0 2017-01-11
+ Creation

## Please Read Carefully:

Information in this document is provided solely in connection with RisingHF products. RisingHF reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All RisingHF products are sold pursuant to RisingHF's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the RisingHF products and services described herein, and RisingHF assumes no liability whatsoever relating to the choice, selection or use of the RisingHF products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by RisingHF for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN RISINGHF'S TERMS AND CONDITIONS OF SALE RisingHF DISCLAIMS ANY EXPRESS OR IMPLIEDWARRANTY WITH RESPECT TO THE USE AND/OR SALE OF RisingHF PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIEDWARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWSOF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

RISINGHF PRODUCTS ARE NOT DESIGNED OR AUTHORIZED FOR USE IN: (A) SAFETY CRITICAL APPLICATIONS SUCH AS LIFE SUPPORTING, ACTIVE IMPLANTED DEVICES OR SYSTEMS WITH PRODUCT FUNCTIONAL SAFETY REQUIREMENTS; (B) AERONAUTIC APPLICATIONS; (C) AUTOMOTIVE APPLICATIONS OR ENVIRONMENTS, AND/OR (D) AEROSPACE APPLICATIONS OR ENVIRONMENTS. WHERE RISINGHF PRODUCTS ARE NOT DESIGNED FOR SUCH USE, THE PURCHASER SHALL USE PRODUCTS AT PURCHASER'S SOLE RISK, EVEN IF RISINGHF HAS BEEN INFORMED IN WRITING OF SUCH USAGE, UNLESS A PRODUCT IS EXPRESSLY DESIGNATED BY RISINGHF AS BEING INTENDED FOR "AUTOMOTIVE, AUTOMOTIVE SAFETY OR MEDICAL" INDUSTRY DOMAINS ACCORDING TO RISINGHF PRODUCT DESIGN SPECIFICATIONS. PRODUCTS FORMALLY ESCC, QML OR JAN QUALIFIED ARE DEEMED SUITABLE FOR USE IN AEROSPACE BY THE CORRESPONDING GOVERNMENTAL AGENCY.

Resale of RisingHF products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by RisingHF for the RisingHF product or service described herein and shall not create or extend in any manner whatsoever, any liability of RisingHF.

RisingHF and the RisingHF logo are trademarks or registered trademarks of RisingHF in various countries.
Information in this document supersedes and replaces all information previously supplied.
The RisingHF logo is a registered trademark of RisingHF. All other names are the property of their respective owners.



http://www.risinghf.com