



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

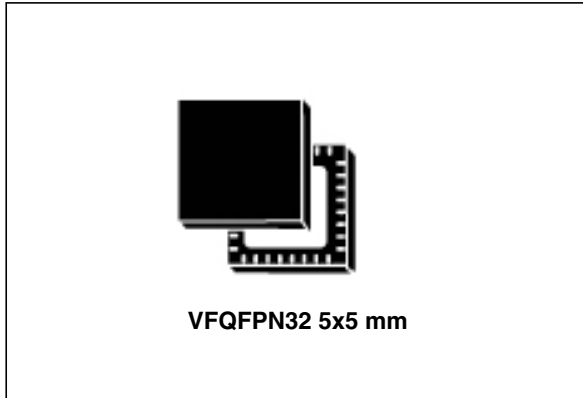
Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



Near field communication transceiver for tag emulation

Datasheet - production data



Applications

Typical protocols supported:

- ISO/IEC 14443-3 Type A tag emulation

Typical RX95HF applications include:

- Keyboard, laptop, set top box, printer, TV, etc.
- Identification, item pairing and data exchange

Features

- Operating mode supported:
 - Tag Emulation for passive peer-to-peer communication
- Hardware features
 - Dedicated internal frame controller
 - Highly integrated Analog Front End (AFE) for RF communications
 - Transmission and reception modes in Tag Emulation mode
 - Optimized power management
 - Field Detection mode
- RF communication @13.56 MHz
 - ISO/IEC 14443 Type A in Tag Emulation mode
- Communication interfaces with a Host Controller
 - Serial peripheral interface (SPI) Slave interface
 - Up to 256-byte command/reception buffer (FIFO)
- 32-lead, 5x5 mm, very thin fine pitch quad flat (VFQFPN) ECOPACK®2 package

Contents

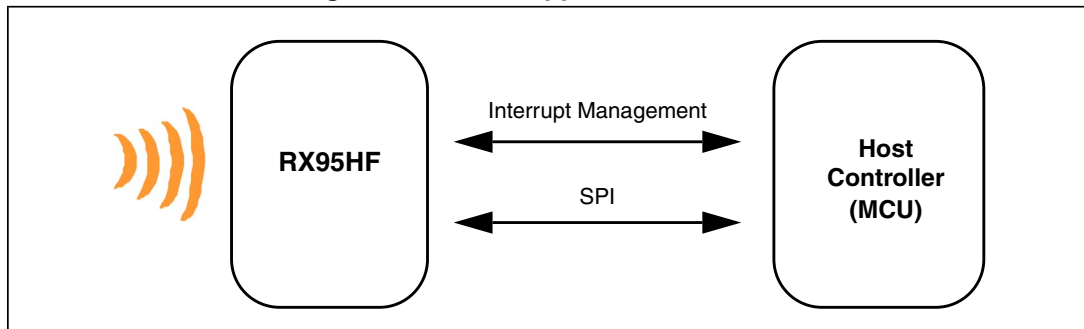
- 1 Description 4**
 - 1.1 Block diagram 4
 - 1.2 List of terms 5
- 2 Pin and signal descriptions 6**
- 3 Power management and operating modes 8**
 - 3.1 Operating modes 8
 - 3.2 Startup sequence 9
- 4 Communication protocols 11**
 - 4.1 Serial peripheral interface (SPI) 11
 - 4.1.1 Polling mode 11
 - 4.1.2 Interrupt mode 13
- 5 Commands 14**
 - 5.1 Command format 14
 - 5.2 List of commands 14
 - 5.3 IDN command (0x01) description 15
 - 5.4 Protocol Select command (0x02) description 15
 - 5.5 Pollfield command (0x03) description 17
 - 5.6 Listen command (0x05) description 17
 - 5.7 Send command (0x06) description 19
 - 5.8 Idle command (0x07) description 20
 - 5.8.1 Idle command parameters 21
 - 5.8.2 Optimizing wake-up conditions 21
 - 5.8.3 Using various techniques to return to Ready state 21
 - 5.9 Read Register (RdReg) command (0x08) description 22
 - 5.10 Write Register (WrReg) command (0x09) description 23
 - 5.10.1 Improving RF performance 23
 - 5.11 AcFilter command (0x0D) description 25
 - 5.12 Echo command (0x55) description 27

6	Electrical characteristics	28
6.1	Absolute maximum ratings	28
6.2	DC characteristics	28
6.3	Power consumption characteristics	29
6.4	SPI characteristics	30
6.5	RF characteristics	32
6.6	Oscillator characteristics	33
7	Package mechanical data	34
8	Part numbering	36
	Appendix A Card emulation communication	37
	Revision history	43

1 Description

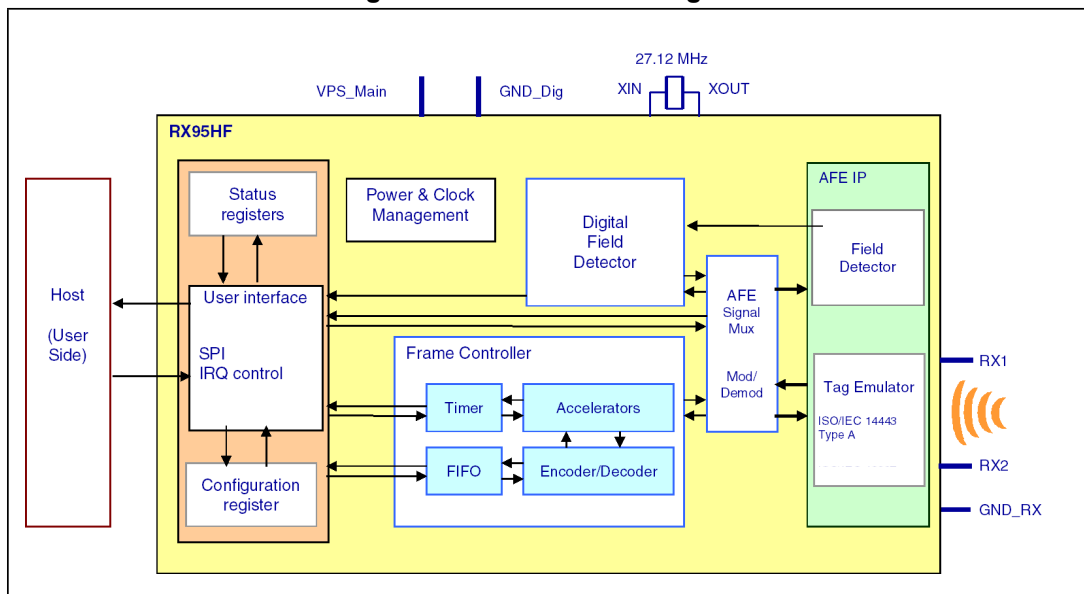
The RX95HF is an integrated transceiver IC for contactless applications. The RX95HF embeds an Analog Front End to provide the 13.56 MHz Air Interface. The RX95HF supports ISO/IEC 14443 Type A communication in Tag Emulation mode.

Figure 1. RX95HF application overview



1.1 Block diagram

Figure 2. RX95HF block diagram



1.2 List of terms

Table 1. List of terms

Term	Meaning
GND	Ground
HFO	High frequency oscillator
LFO	Low frequency oscillator
MCU	Microcontroller unit
NFC	Near Field Communication
RFID	Radio Frequency Identification
RFU	Reserved for future use
SPI	Serial peripheral interface
t_L	Low frequency period
t_{REF}	Reference time
WFE	Wait For Event

2 Pin and signal descriptions

Figure 3. RX95HF pinout description

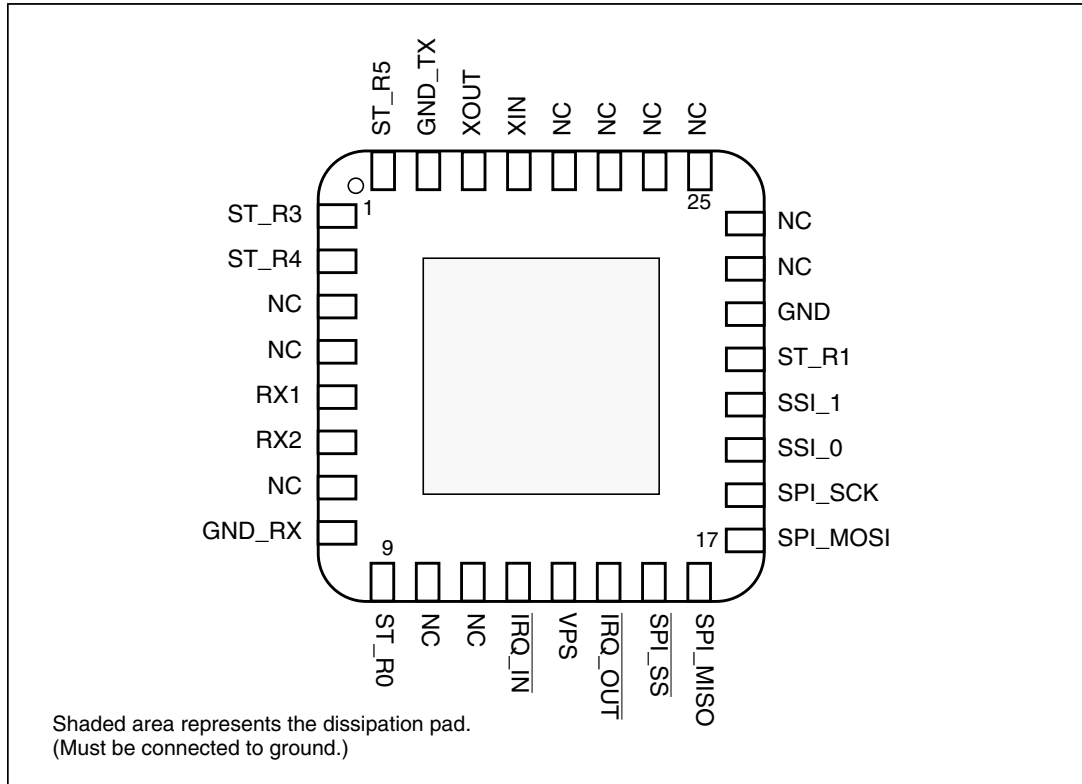


Table 2. RX95HF pin descriptions

Pin	Pin name	Type ⁽¹⁾	Main function	Alternate function
1	ST_R3		ST Reserved	
2	ST_R4		ST Reserved	
3	NC		Not connected	
4	NC		Not connected	
5	RX1	I	Receiver input 1	
6	RX2	I	Receiver input 2	
7	NC		Not connected	
8	GND_RX	P	Ground (analog)	
9	ST_R0	O	ST Reserved ⁽²⁾	
10	NC		Not connected	
11	NC		Not connected	
12	$\overline{\text{IRQ_IN}}$	I ⁽³⁾	Interrupt input	
13	VPS	P	Main power supply	

Table 2. RX95HF pin descriptions (continued)

Pin	Pin name	Type ⁽¹⁾	Main function	Alternate function
14	IRQ_OUT	O ⁽⁴⁾	Interrupt output	
15	SPI_SS	I ⁽⁵⁾	SPI Slave Select (active low)	
16	SPI_MISO	O ⁽⁵⁾	SPI Data, Slave Output	
17	SPI_MOSI	I ⁽⁵⁾	SPI Data, Slave Input ⁽⁵⁾	
18	SPI_SCK	I ⁽⁶⁾	SPI serial clock	
19	SSI_0	I ⁽⁵⁾	Select serial communication interface	
20	SSI_1	I ⁽⁵⁾	Select serial communication interface	
21	ST_R1	I ⁽⁷⁾	ST Reserved	
22	GND	P	Ground (digital)	
23	NC		Not connected	
24	NC		Not connected	
25	NC		Not connected	
26	NC		Not connected	
27	NC		Not connected	
28	NC		Not connected	
29	XIN		Crystal oscillator input	
30	XOUT		Crystal oscillator output	
31	GND_TX	P	Ground (RF drivers)	
32	ST_R5		ST Reserved	

1. I: Input, O: Output, and P: Power
2. Must add a capacitor to ground (~1 nF).
3. Pad internally connected to a Very Weak Pull-up to VPS.
4. Pad internally connected to a Weak Pull-up to VPS.
5. Must not be left floating.
6. Pad internally connected to a Weak Pull-down to GND.
7. Pad input in High Impedance. Must be connected to VPS.

3 Power management and operating modes

3.1 Operating modes

The RX95HF has 2 operating modes: Wait for Event (WFE) and Active. In Active mode, the RX95HF communicates actively with an NFC reader or an external host (an MCU, for example).

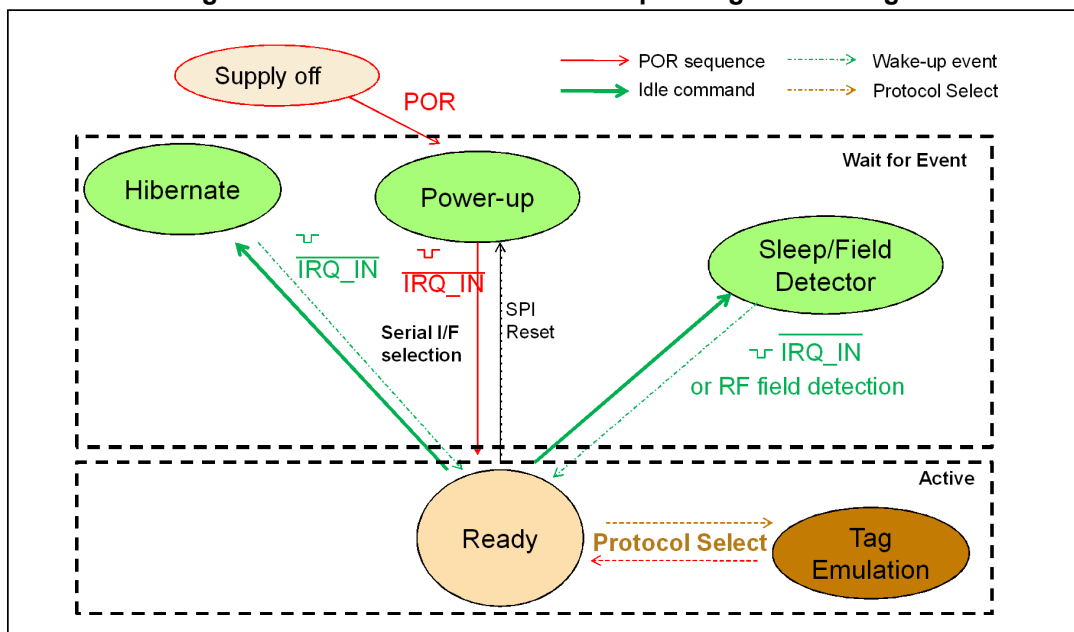
The RX95HF can switch from one mode to another.

Table 3. RX95HF operating modes and states

Mode	State	Description
Wait For Event (WFE)	Power-up	This mode is accessible directly after POR. Low level on $\overline{\text{IRQ_IN}}$ pin (longer than 10 μs) is the only wakeup source. LFO (low-frequency oscillator) is running in this state.
	Hibernate	Lowest power consumption state. The RX95HF has to be woken-up in order to communicate. Low level on $\overline{\text{IRQ_IN}}$ pin (longer than 10 μs) is the only wakeup source.
	Sleep/Field Detector	Low power consumption state. Wakeup source is configurable: – $\overline{\text{IRQ_IN}}$ pin – Field Detector LFO (low-frequency oscillator) is running in this state.
Active	Ready	In this mode, the RX95HF waits for a command (PROTOCOLSELECT, ...) from the external host via the serial interface (SPI).
	Tag Emulation	The RX95HF can communicate as a tag with an external reader. The tag application is located in the Host and communicates with the RX95HF via the serial interface (SPI).

Hibernate, and Sleep/Field Detector states can only be activated by a command from the external host. As soon as any of these three states are activated, the RX95HF can no longer communicate with the external host. It can only be woken up.

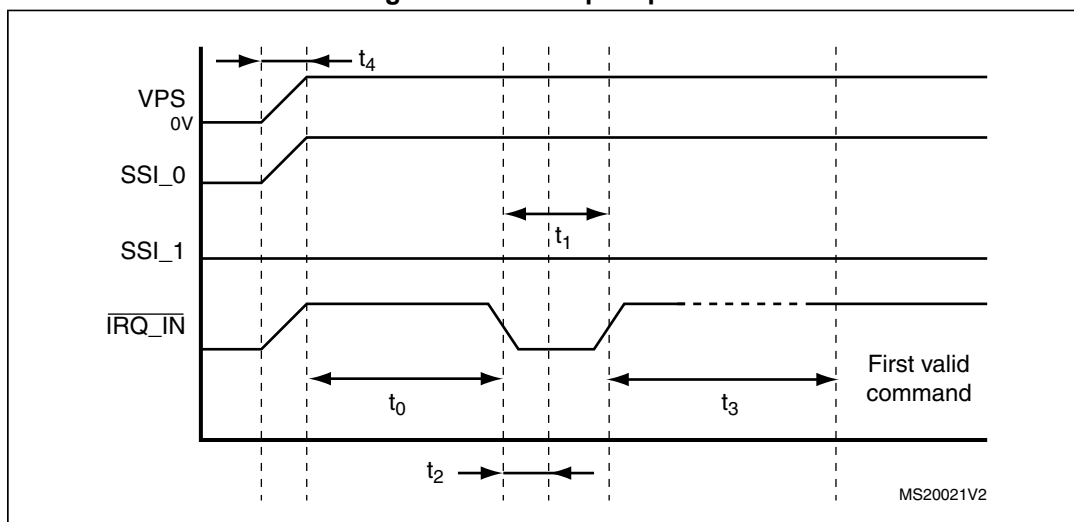
Figure 4. RX95HF initialization and operating state change



3.2 Startup sequence

After the power supply is established at power-on, the RX95HF waits for a low pulse on the pin $\overline{\text{IRQ_IN}}$ (t_1) before automatically selecting the external interface (SPI) and entering Ready state after a delay (t_3).

Figure 5. Power-up sequence



1. Pin $\overline{\text{IRQ_IN}}$ low level < 0.2 VPS_{Main}.

Note: When RX95HF leaves WFE mode (from Power-up, Hibernate, or Sleep/Field Detector) following an $\overline{\text{IRQ_IN}}$ low level pulse.

Figure 5 shows the power-up sequence for a RX95HF device; where,

- t_0 is the initial wake-up delay 100 μ s (minimum)
- t_1 is the minimum interrupt width 10 μ s (minimum)
- t_2 is the delay for the serial interface selection 250 ns (typical)
- t_3 is the HFO setup time ($t_{SU(HFO)}$) 10 ms (maximum)
- t_4 is the V_{PS} ramp-up time from 0V to V_{PS} 200 μ s (minimum) and 10 ms (max. by design validation)

Note: *VPS must be 0V before executing the start-up sequence.*

The serial interface is selected after the following falling edge of pin $\overline{TRQ_IN}$ when leaving from POR or Hibernate state.

Table 4 lists the signal configuration used to select the serial communication interface.

Table 4. Select serial communication interface selection table

Pin	Serial interface (SPI)
SSI_0	1
SSI_1	0

4 Communication protocols

4.1 Serial peripheral interface (SPI)

4.1.1 Polling mode

In order to send commands and receive replies, the application software has to perform 3 steps.

1. Send the command to the RX95HF.
2. Poll the RX95HF until it is ready to transmit the response.
3. Read the response.

The application software should never read data from the RX95HF without being sure that the RX95HF is ready to send the response.

The maximum allowed SPI communication speed is f_{SCK} .

A Control byte is used to specify a communication type and direction:

- 0x00: Send command to the RX95HF
- 0x03: Poll the RX95HF
- 0x02: Read data from the RX95HF
- 0x01: Reset the RX95HF

The $\overline{SPI_SS}$ line is used to select a device on the common SPI bus. The $\overline{SPI_SS}$ pin is active low.

When the $\overline{SPI_SS}$ line is inactive, all data sent by the Master device is ignored and the MISO line remains in High Impedance state.

Figure 6. Sending command to RX95HF

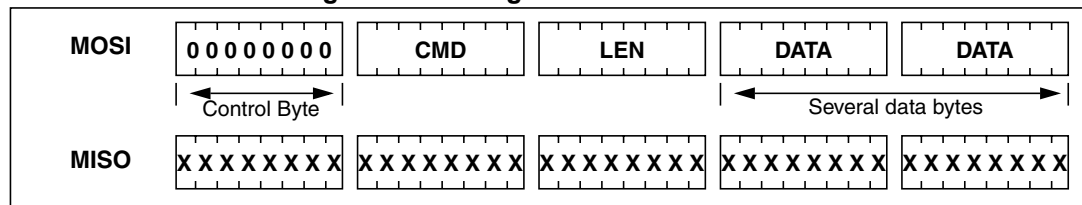


Figure 7. Polling the RX95HF until it is ready

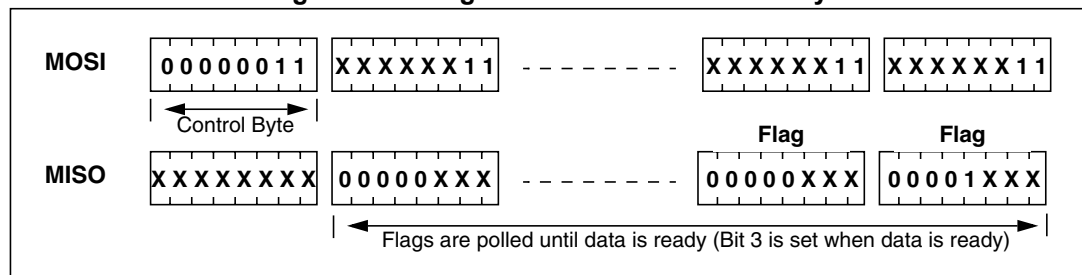
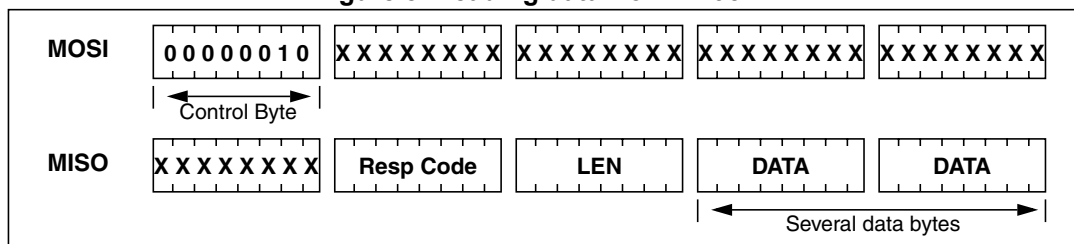


Table 5. Interpretation of flags

Bit	Meaning (Application point of view)
[7:4]	Not significant
3	Data can be read from the RX95HF when set.
2	Data can be sent to the RX95HF when set.
[1:0]	Not significant

Figure 8. Reading data from RX95HF



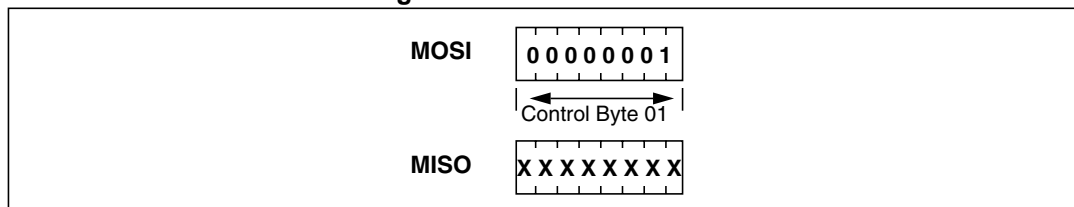
Data must be sampled at the rising edge of the SCK signal.

'Sending', 'Polling' and 'Reading' commands must be separated by a high level of the SPI_SS line. For example, when the application needs to wait for data from the RX95HF, it asserts the SPI_SS line low and issues a 'Polling' command. Keeping the SPI_SS line low, the Host can read the Flags Waiting bit which indicates that the RX95HF can be read. Then, the application has to assert the SPI_SS line high to finish the polling command. The Host asserts the SPI_SS line low and issues a 'Reading' command to read data. When all data is read, the application asserts the SPI_SS line high.

The application is not obliged to keep reading Flags using the Polling command until the RX95HF is ready in one command. It can issue as many 'Polling' commands as necessary. For example, the application asserts SPI_SS low, issues 'Polling' commands and reads Flags. If the RX95HF is not ready, the application can assert SPI_SS high and continue its algorithm (measuring temperature, communication with something else). Then, the application can assert SPI_SS low again and again issue 'Polling' commands, and so on, as many times as necessary, until the RX95HF is ready.

Note that at the beginning of communication, the application does not need to check flags to start transmission. The RX95HF is assumed to be ready to receive a command from the application.

Figure 9. Reset the RX95HF



To reset the RX95HF using the SPI, the application sends the SPI Reset command (Control Byte 01, see Figure 9) which starts the internal controller reset process and puts the RX95HF into Power-up state. The RX95HF will wake up when pin IRQ_IN goes low. The RX95HF reset process only starts when the SPI_SS pin returns to high level.

Caution: SPI communication is MSB first.

4.1.2 Interrupt mode

When the RX95HF is configured to use the SPI serial interface, pin $\overline{\text{IRQ_OUT}}$ is used to give additional information to user. When the RX95HF is ready to send back a reply, it sends an Interrupt Request by setting a low level on pin $\overline{\text{IRQ_OUT}}$, which remains low until the host reads the data.

The application can use the Interrupt mode to skip the polling stage.

Caution: SPI communication is MSB first.

5 Commands

5.1 Command format

- The frame from the Host to the RX95HF has the following format:
 <CMD><Len><Data>
- The frame from the RX95HF to Host has the following format:
 <RespCode><Len><Data>

These two formats are available in SPI mode.

Fields <Cmd>, <RespCode> and <Len> are always 1 byte long. <Data> can be from 0 to 253 bytes.

Note: The ECHO command is an exception as it has only one byte (0x55).

The following symbols correspond to:

- >>> Frame sent by the Host to RX95HF
- <<< Frame sent by the RX95HF to the Host

5.2 List of commands

Table 6 summarizes the available commands.

Table 6. List of RX95HF commands

Code	Command	Description
0x01	IDN	Requests short information about the RX95HF and its revision.
0x02	PROTOCOLSELECT	Selects the RF communication protocol and specifies certain protocol-related parameters.
0x03	POLLFIELD	Returns the current value of the FieldDet flag.
0x05	LISTEN	Listens for data using previously selected protocol.
0x06	SEND	Sends data using previously selected protocol.
0x07	IDLE	Switches the RX95HF into a low consumption Wait for Event (WFE) mode (Power-up, Hibernate, or Sleep/Field Detector), specifies the authorized wake-up sources and waits for an event to exit to Ready state.
0x08	RdREG	Reads Wake-up event register or the Analog Register Configuration (ACC_A) register.
0x09	WRREG	Writes Analog Register Configuration (ACC_A) register or writes index of ACC_A register address.
0x0D	ACFILTER	Enables or disables the anti-collision filter for ISO/IEC 14443 Type A protocol.
0x55	ECHO	RX95HF performs a serial interface ECHO command (reply data 0x55 or stops the Listening state when a listen command has been sent without error).
Other codes		ST Reserved

5.3 IDN command (0x01) description

The IDN command (0x01) gives brief information about the RX95HF and its revision.

Table 7. IDN command description

Direction	Data	Comments	Example
Host to RX95HF	0x01	Command code	>>>0x0100
	0x00	Length of data	
RX95HF to Host	0x00	Result code	<<<0x000F4E4643204653324A41535434002ACE In this example, <<<0x4E4643204653324A4153543400 : 'NFC FS2JAST4', #4 (Last Character of NFC FS2JAST4 means ROM code revision 4.) 0x2ACE: CRC of ROM (real CRC may differ from this example)
	<Len>	Length of data	
	<Device ID>	Data in ASCII format (13 bytes)	
	<ROM CRC>	CRC calculated for ROM content (2 bytes)	

It takes approximately 6 ms to calculate the CRC for the entire ROM. The application must allow sufficient time for waiting for a response for this command.

5.4 Protocol Select command (0x02) description

This command selects the RF communication protocol and prepares the RX95HF for communication with a reader.

Table 8. PROTOCOLSELECT command description

Direction	Data	Comments	Example
Host to RX95HF	0x02	Command code	See Table 9: List of <Parameters> values for different protocols (Tag Emulation) for a detailed example. >>>0x02021208
	<Len>	Length of data	
	<Protocol>	Protocol codes: 12: ISO/IEC 14443-A	
	<Parameters>	Each protocol has a different set of parameters. See Table 9 .	
RX95HF to Host	0x00	Result code	<<<0x0000
	0x00	Length of data	Protocol is successfully selected
RX95HF to Host	0x82	Error code	<<<0x8200
	0x00	Length of data	Invalid command length
RX95HF to Host	0x83	Error code	<<<0x8300
	0x00	Length of data	Invalid protocol

When the application selects a protocol, the RX95HF performs all necessary settings: it will choose the appropriate reception and transmission chains.

Table 9. List of <Parameters> values for different protocols (Tag Emulation)

Protocol (Card)	Code	Parameters			Examples of commands Comments
		Byte	Bit	Function	
ISO/IEC 14443 Type A	0x12	0	7:6	Transmission data rate 00: 106 Kbps 01: 212 Kbps ⁽¹⁾ 10: 424 Kbps ⁽¹⁾ 11: RFU	>>>0x02021208 <<<0x0000 Tag Emulation for ISO/IEC 14443 Type A, Data rate is 106 Kbps for both up- and down-links.
			5:4	Reception data rate 00: 106 Kbps 01: 212 Kbps ⁽¹⁾ 10: 424 Kbps ⁽¹⁾ 11: RFU	
			3	0: Return an error, if no RF field 1: Wait for RF field	
			2	RFU	
			1	0: HFO 1: ClkRec	
			0	RFU	

1. Not qualified for this version.

5.5 Pollfield command (0x03) description

This command returns the current value of the FieldDet flag.

Table 10. POLLFIELD command description

Direction	Data	Comments	Example
Host to RX95HF	03	Command code	>>>0x0300 Check if RF Field is ON or OFF >>> 0x0303010FFF - Wait for RF Field appearance for (16*256)/13.56 μs Flags, Presc and Timer parameters are optional. They must be specified if the application has to wait for RF field appearance or disappearance. The time to wait is (Presc+1)*(Timer+1)/13.56 μs.
	<Len>	Length of data	
	<Flags>	RF field presence (Optional): 01: Wait for RF field appearance 00: Wait for RF field disappearance	
	<Presc>	Timer prescaler (Optional)	
	<Timer>	Timer time-out (Optional)	
RX95HF to Host	00	Result code	<<<0x0000 or 0x000100 (No RF field detected) or 0x000101 (RF field detected)
	01	Length of data	
	<FieldDet>	01, if FieldDet is set. Otherwise, 00.	

Table 11. Response for <POLLFIELD> command

Function	Explanation	Response example	Comments
Pollfield	Response	00 01	This command returns the current state of the RF field. <<<0x0000 or 0x000100 (No RF field detected) or 0x000101 (RF field detected)
	Result code	01 or 00	
	Length of data field		
	Bits [7:1]: RFU Bit 0: Field detected (if set)		

5.6 Listen command (0x05) description

In Tag Emulation mode, this command waits for a command from an external reader.

Before sending this command, the application must select a protocol.

Table 12. LISTEN command description

Direction	Data	Comments	Example
Host to RX95HF	05	Command code	0x0500: Enters a Listening mode where the RX95HF waits for a command from an external reader.
	00	Length of data	
RX95HF to Host	00	Result code	0x0000: No error. Confirmation that RX95HF now is in Listening mode.
	00	Length of data	

Table 12. LISTEN command description (continued)

Direction	Data	Comments	Example
RX95HF to Host	82	Error code	0x8200: Invalid command length
	00	Length of data	
RX95HF to Host	83	Error code	0x8300: Invalid protocol or protocol is not supported.
	00	Length of data	
RX95HF to Host	8F	Error code	0x8F00: No field. Command cannot be executed because there is no external field.
	00	Length of data	

After reception of the LISTEN command and the return of a 'No error' confirmation, the RX95HF enters Listening mode. The host controller has to use SPI Poll mode or IRQ_OUT pin to detect when a receive frame is available in the RX95HF buffer. The RX95HF will exit Listening mode as soon it receives the ECHO command (0x55) from the Host Controller (MCU) or a command from an external reader (not including commands supported by the ACFILTER command listed in [Table 29](#)).

In all cases, the RX95HF will send data or an error code to the Host controller (MCU).

The ECHO command (0x55) allows exiting Listening mode. In response to the ECHO command, the RX95HF sends 0x55 + 0x8500 (error code of the Listening state cancelled by the MCU).

Possible return codes are listed in [Table 13](#).

Table 13. Response codes from the RX95HF in Listening mode

Direction	Data	Comments	Example
RX95HF to Host	80	Result code	<<<0x8005E050BCA508 The request from the Reader is decoded. This is an example of a response in ISO/IEC 14443 Type A protocol.
	<Len>	Length of data	
	<Data>	Data received. Interpretation depends on protocol	
RX95HF to Host	85	Error code	<<<0x8500 Listening mode was cancelled by the application
	00	Length of data	
RX95HF to Host	86	Error code	<<<0x8600 Communication error
	00	Length of data	
RX95HF to Host	88	Error code	<<<0x8800 Invalid SOF
	00	Length of data	
RX95HF to Host	89	Error code	<<<0x8900 Receive buffer overflow (too many bytes received)
	00	Length of data	
RX95HF to Host	8A	Error code	<<<0x8A00 Framing error (start bit=0, stop bit=1)
	00	Length of data	

Table 13. Response codes from the RX95HF in Listening mode (continued)

Direction	Data	Comments	Example
RX95HF - Host	8E	Error code	<<<0x8E00 Reception lost without EOF received
	00	Length of data	

If the request from the Reader was received and decoded correctly, the RX95HF will send data back to the Host (Tag Emulation application). This is explained in [Table 14](#).

Table 14. Data format sent to the Host in Listening mode

Protocol	Explanation	Response example						Comments
ISO/IEC-14443 Type A	Request example	80	0A	9370800F8C8E	8D	4E01	08	<<<0x80 0A 9370800F8C8E 8D 4E01 08
	Result code							
	Length of entire data field							
	Data received from reader							
	Received value of BCC (if any)							
	Received value of CRC (if any)							
	7: RFU 6: RFU 5: CRC error 4: Parity error 3:0: number of significant bits in last byte							

5.7 Send command (0x06) description

This command immediately sends data to the reader using the Load Modulation method without waiting for a reply.

Before sending this command, the application must select a communication protocol.

Table 15. SEND command description

Direction	Data	Comments	Example
Host to RX95HF	06	Command code	Depends on protocol previously selected! >>>0x0606D07387080028: Emulation of TAG response in ISO/IEC 14443 Type A protocol
	<Len>	Length of data	
	<Data>	Data and additional parameter to be sent	
RX95HF to Host	00	Result code	<<<0x0000
	00	Length of data	Data was successfully sent
RX95HF to Host	82	Error code	<<<0x8200 Invalid length (for example, Length=0 where it is not possible)
	00	Length of data	
RX95HF to Host	83	Error code	<<<0x8300 Invalid protocol previously selected by Select Protocol command
	00	Length of data	

Table 16. Format of data to be sent using SEND command

Protocol	Explanation	Response example				Comments
ISO/IEC-14443 Type A	Send example	06	03	0400	08	>>>0x0603040008
	Command code					
	Length of entire data field					
	Data					
	Parameter: 7:6: RFU 5: Append CRC 4: RFU (Do not append parity) 3:0: Number of significant bits in first byte					

5.8 Idle command (0x07) description

This command switches the RX95HF into low consumption mode and defines the way to return to Ready state.

The Result code contains the Wake-up flag register value indicating to the application the wake-up event that caused the device to exit WFE mode.

Table 17. IDLE command description

Direction	Data	Comments	Example
Host to RX95HF	07	Command code	Example of switch from Active mode to Hibernate state (wake-up by low pulse on IRQ_IN pin): >>>0x07 0E 08 04 00 04 00 18 00 00 00 00 00 00 00 00
	0E	Length of data	
	<WU Source>	Specifies authorized wake-up sources and the LFO frequency	
	EnterCtrlL	Settings to enter WFE mode: 0x0400, 0x0100 or 0x0142	Example of switch from Active to WFE mode (Sleep) (wake-up by low pulse on IRQ_IN pin): >>>0x07 0E 08 01 00 38 00 18 00 00 60 00 00 00 00 00
	EnterCtrlH		
	WUCtrlL	Settings to wake-up from WFE mode: 0x0400 or 0x3800	Example of switch from Active mode to WFE mode (Field Detector) with wake-up by Field Detection: >>> 0x07 0E 04 01 42 38 00 18 00 00 60 00 00 00 00 00
	WUCtrlH		
	LeaveCtrlL	Settings to leave WFE mode (Default value = 0x1800)	
LeaveCtrlH			
RX95HF to Host	0x00	Result code	This response is sent only when RX95HF exits WFE mode. <<<0x000104 Wake-up by field detect <<<0x000108 Wake-up by low pulse on IRQ_IN pin
	0x01	Length of data	
	<Data>	Data (Wake-up source): 0x04: Field detect 0x08: Low pulse on IRQ_IN pin	

Table 17. Idle command description (continued)

Direction	Data	Comments	Example
RX95HF to Host	0x82	Error code	<<<0x8200 Invalid command length
	0x00	Length of data	

5.8.1 Idle command parameters

The Idle command (Host to RX95HF) has the following structure (all values are hexadecimal):

Table 18. Idle command structure

07	0E	xx	YY ZZ	YY ZZ	YY ZZ
Command code	Data length	WU source	Enter Control	WU Control	Leave Control

Table 19. Summary of Idle command parameters

Parameter	Description
Command code	This byte is the command code. '07' represents the Idle command. This command switches the device from Active mode to WFE mode.
Data length	This byte is the length of the command in bytes. Its value depends on the following parameter values.
WU Source	This byte defines the authorized wake-up sources in the Wake-up source register. Predefined values are: 0x04: Field Detector 0x08: Low pulse on $\overline{\text{IRQ_IN}}$
Enter Control	These two bytes (EnterCtrlL and EnterCtrlH) define the resources when entering WFE mode. 0x0400: Hibernate 0x0142: Sleep (if Field Detector source is enabled)
WU Control	These two bytes (WuCtrlL and WuCtrlH) define the wake-up resources. 0x0400: Hibernate 0x3800: Sleep
Leave Control	These two bytes (LeaveCtrlL and LeaveCtrlH) define the resources when returning to Ready state. 0x1800: Hibernate 0x1800: Sleep

5.8.2 Optimizing wake-up conditions

Using the Wake-up source register, it is possible to cumulate sources for a wake-up event. It is strongly recommended to always set an external event as a possible wake-up source.

To cumulate wake-up sources, simply set the corresponding bits in the Wake-up source register. For example, to enable a wake-up when an RF field is detected (bit 1 set to '1') or on a low pulse on pin $\overline{\text{IRQ_IN}}$ (bit 3 set to '1'), set the register to 0x0A.

5.8.3 Using various techniques to return to Ready state

The Idle command and reply set offers several benefits to users by enabling various methods to return the RX95HF to Ready state. Some methods are nearly automatic, such as waiting for a timer overflow or a field detection, but others consume more power compared to the ones requesting a host action. A description of each method follows below.

Default setting: from POR to Ready state

After power-on, the RX95HF enters Power-up state.

To wake up the RX95HF and set it to Ready state, the user must send a low pulse on the $\overline{\text{IRQ_IN}}$ pin. The RX95HF then automatically selects the external interface (SPI) and enters Ready state and is able to accept commSands after a delay of approximately 6 ms (t_3).

From Ready state to Hibernate state and back to Ready state

In Hibernate state, most resources are switched off to achieve an ultra-low power consumption.

The only way the RX95HF can wake-up from Hibernate state is by an external event (low pulse on pin $\overline{\text{IRQ_IN}}$).

A basic Idle command is:

```
>>>0x07 0E 08 04 00 04 00 18 00 00 00 00 00 00 00
```

Note: The Wake-up flag value is NOT significant when returning to Ready state from Hibernate state or after a POR.

From Ready state to Sleep state and back to Ready state

Wake-up by external event (low pulse on $\overline{\text{IRQ_IN}}$)

In Sleep or Power-up states, operating resources are limited in function of the selected wake-up source to achieve a moderate power consumption level.

An Idle command example when wake-up source is pin $\overline{\text{IRQ_IN}}$:

```
>>>0x07 0E 08 01 00 38 00 18 00 00 60 00 00 00 00
```

5.9 Read Register (RdReg) command (0x08) description

This command is used to read the Wakeup register.

Table 20. RdREG command description

Direction	Data	Comments	Example
Host to RX95HF	0x08	Command code	Ex 1. >>>0x0803690100 Reads the ACC_A register. ⁽¹⁾
	0x03	Length of data	
	0x62 or 0x69	Register address	
	0x01	Register size	
	0x00	ST Reserved	
RX95HF to Host	0x00	Result code	<<<0x000127
	<Len>	Length of data	See Section 5.10: Write Register (WrReg) command (0x09) description for more information on received register data.
	<RegData>	Register data	
RX95HF to Host	0x82	Error code	<<<0x8200 Invalid command length
	0x00	Length of data	

1. This command must be preceded by the setting of the ACC_A register index (0x0903680004) .

Note: The Management of the Analog Register Configuration register (ACC_A) is described in Section 5.10: Write Register (WrReg) command (0x09) description.

5.10 Write Register (WrReg) command (0x09) description

The Write Register (WRREG) command (0x09) is used to:

- set the Analog Register Configuration address index value before reading or overwriting the Analog Register Configuration register (ACC_A) value
- set the Analog Register Configuration register (ACC_A) value
 - Bits [7:6] are ST Reserved and must be set to '0'
 - Bits [5:4] set the Demodulator Sensitivity ('00' = AM 10% or '10' = AM 100%)
 - Bits [3:0] set the Load Modulation Index ('0001' = minimum load modulation, '0111' = default value, '1111' = maximum load modulation). The user can adjust this value to better fit his application.

5.10.1 Improving RF performance

Adjusting the Load Modulation Index and Demodulator Sensitivity parameters in Card Emulation mode helps improve application behavior.

The default value of these parameters ([Table 21](#)) is set by the PROTOCOLSELECT command, but they can be overwritten using the Write Register (WRREG) command (0x09). [Table 22](#) and [Table 23](#) list possible values for the Load Modulation Index and Demodulator Sensitivity parameters respectively.

This new configuration is valid until a new PROTOCOLSELECT or Write Register (of register ACC_A) command is executed. Register values are cleared at power off.

Table 21. WRREG command description (Load Modulation Index and Demodulator Sensitivity)

Direction	Data	Comments	Example
Host to RX95HF	0x09	Command code	
	0x03 or 0x04	Length of data	
	0x68	Analog Register Configuration address index	>>>0x0903680004 Set Analog Register Index to 0x04 (ACC_A) ⁽¹⁾
	0x00 or 0x01	Flag Increment address or not after Write command	
	0x04	Index pointing to the Demodulator Sensitivity and Load Modulation values in ACC_A register (0x04) (See Section 5.10.1)	>>>0x090468010425 Update ACC_A to 0x25
	0xXX	New value for Demodulator Sensitivity and Load Modulation (See Section 5.10.1)	
RX95HF to Host	0x00	Result code	<<<0x0000
	0x00	Length of data (= RegCount)	Register written

1. This command must be executed before reading the ACC_A register (0x0903680004).

How to modify Analog Register Configuration register (ACC_A) values

1. Use the `PROTOCOLSELECT` command (0x02) to select the correct communication protocol.

For example, to select the ISO/IEC 14443-A card emulation protocol:

```
Send PROTOCOLSELECT command: >>>0x02021208
RX95HF reply: <<<0x0000
```

2. Read the Analog Register Configuration register (ACC_A) value.

```
a) Write the ACC_A register index at 0x04: >>>0x0903680004
RX95HF reply: <<<0x0000
b) Read the ACC_A register value: >>>0x0803690100
RX95HF reply: <<<0x0127
```

In this example, the ACC_A register value is 0x27, where “2” is the Demodulator Sensitivity and “7” is the Load Modulation Index.

3. Modify the Load Modulation Index values with 0x25.

```
Write the ACC_A register index: >>>0x090468010425
RX95HF reply: <<<0x0000
```

4. Read the Analog Configuration register (ACC_A) value.

```
a) Write the ACC_A register index at 0x04: >>>0x0903680004
RX95HF reply: <<<0x0000
b) Read the ACC_A register value: >>>0x0803690100
RX95HF reply: <<<0x0125
```

Load Modulation Index and Demodulator Sensitivity values

Table 22. Possible Load Modulation Index values

Code (bits [3:0] of ACC_A register)	1	2	...	7	...	E	F
Load Modulation Index	Min.			Default			Max.

Table 23. Possible Demodulator Sensitivity values

Code (bits [5:4] of ACC_A register)	0x1	0x2
Demodulator Sensitivity ⁽¹⁾	10%	100%

1. Characterized by design simulation.

Note: Bits [7:6] of ACC_A register are RFU and must set to '0'.

5.11 AcFilter command (0x0D) description

This command activates/deactivates the anti-collision filter in Type A Tag Emulation mode. The following tables define the command and response fields depending on the AC filter command length.

Table 24. AC filter command description (Command <Len> > 0x02)

Direction	Data	Comments	Example
Host to RX95HF	0D	Command code	General case (LEN ≠ 0x00, 0x01 or 0x02)
	<Len>	Length of data	
	<ATQA> (2 bytes)	Unused and proprietary bits of SAK (protocol bits will be handled by firmware)	>>>0D0B440320880251744AEF2280 Activate filter for 2-cascade anti-collision
	<SAK>	Unused and proprietary bits of SAK (protocol bits will be handled by firmware)	Note that the command length can be 7, 11 or 15 bytes depending on UID length: 7: for 1-cascade level filter 11: for 2-cascade levels filter 15: for 3-cascade levels filter
	<UID part 1>	UID for cascade level 1 (Mandatory)	All other values greater than 2 will result in an 'Invalid command length' error. Values 0, 1 and 2 are defined in the following tables.
	<UID part 2>	UID for cascade level 2 (Optional)	
	<UID part 3>	UID for cascade level 3 (Optional)	*** Special cases of command *** LEN = 0x00 >>>0x0D00 De-activate AC Filter (LEN = 0x00) <<< 0x0001XX0000 AC Filter is successfully deactivated (XX: Table 28) LEN = 0x01 >>>0x0D01XX - Set AC state (see table below) <<< 0x0000 AC state successfully set LEN = 0x02 >>> 0x0D020000 - Returns AC state without deactivating filter <<< 0x0001XX AC state (XX: Table 28)
RX95HF to Host	00	Result code	AC Filter is successfully activated. <<<0x0000
	00	Length of data	AC Filter is successfully deactivated (LEN = 0x00). <<<0x0001XX0000 (where XX is ST Reserved)
RX95HF to Host	82	Error code	<<<0x8200
	00	Length of data	Invalid command length