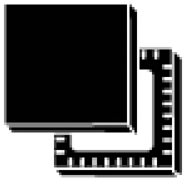
ST95HF

Near field communication transceiver

Datasheet - production data



VFQFPN32 5x5 mm

Features

- Belonging to ST25 family, that includes all NFC/RF ID tag and reader products from STMicroelectronics
- Operating modes supported:
  - Reader/Writer
  - Card Emulation (ISO/IEC 14443-3 Type A)
- Hardware features
  - Dedicated internal frame controller
  - Highly integrated Analog Front End (AFE) for RF communications
  - Transmission and reception modes
  - Optimized power management
  - Tag Detection mode
  - Field Detection mode
- RF communication @13.56 MHz
  - ISO/IEC 14443 Type A and B in Reader and Card Emulation modes
  - ISO/IEC 15693 in Reader mode
  - ISO/IEC 18092 in Reader and Card Emulation modes
  - MIFARE® Classic compatible
- Communication interfaces with a Host Controller
  - Serial peripheral interface (SPI) Slave interface up to 2 Mbps
  - Up to 528-byte command/reception buffer (FIFO) depending on communication protocol
- 32-lead, 5x5 mm, very thin fine pitch quad flat (VFQFPN) ECOPACK®2 package

Applications

Typical protocols supported:

- ISO/IEC 14443-3 Type A and B tags
- ISO/IEC 15693 tags
- ISO/IEC 18000-3M1 tags
- NFC Forum tags: Types 1, 2, 3 and 4
- ST Dual Interface EEPROM

Typical ST95HF applications include:

- Consumer electronics
- Gaming
- Healthcare
- Industrial

Typical ST95HF use cases include:

- NFC-enabled Wi-Fi pairing
- NFC-enabled Bluetooth pairing
- Data exchange

Communications with NFC/RFID tag (reader mode)

This is information on a product in full production.

Contents

List of tables

List of figures

1 Description

The ST95HF is an integrated transceiver IC for contactless applications.

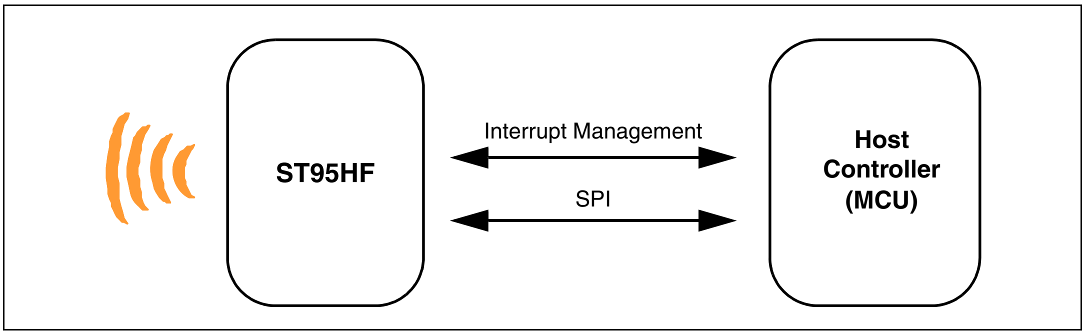
The ST95HF manages frame coding and decoding in Reader and Card Emulation modes for standard applications such as Near Field Communication (NFC), proximity and vicinity standards.

The ST95HF embeds an Analog Front End to provide the 13.56 MHz Air Interface.

The ST95HF supports ISO/IEC 14443 Type A and B communication in Reader and Card Emulation modes, ISO/IEC 15693 (single or double subcarrier in Reader mode only) and ISO/IEC 18092 communication protocols in Reader and Card Emulation modes.

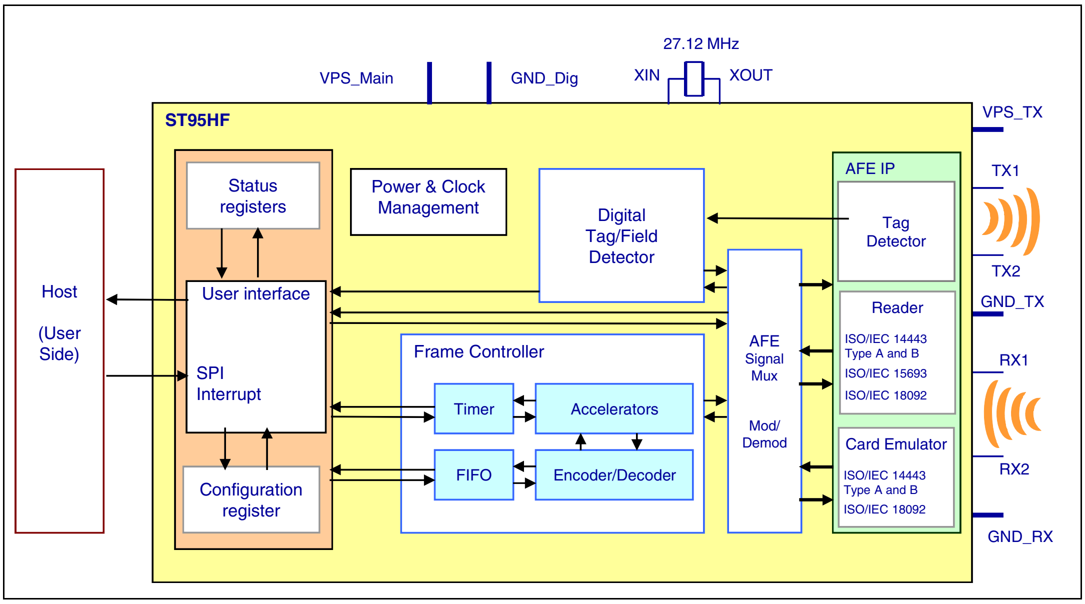
The ST95HF also supports the detection, reading and writing of NFC Forum Type 1, 2, 3 and 4 tags.

Figure 1. ST95HF application overview



1.1 Block diagram

Figure 2. ST95HF block diagram



1.2 List of terms

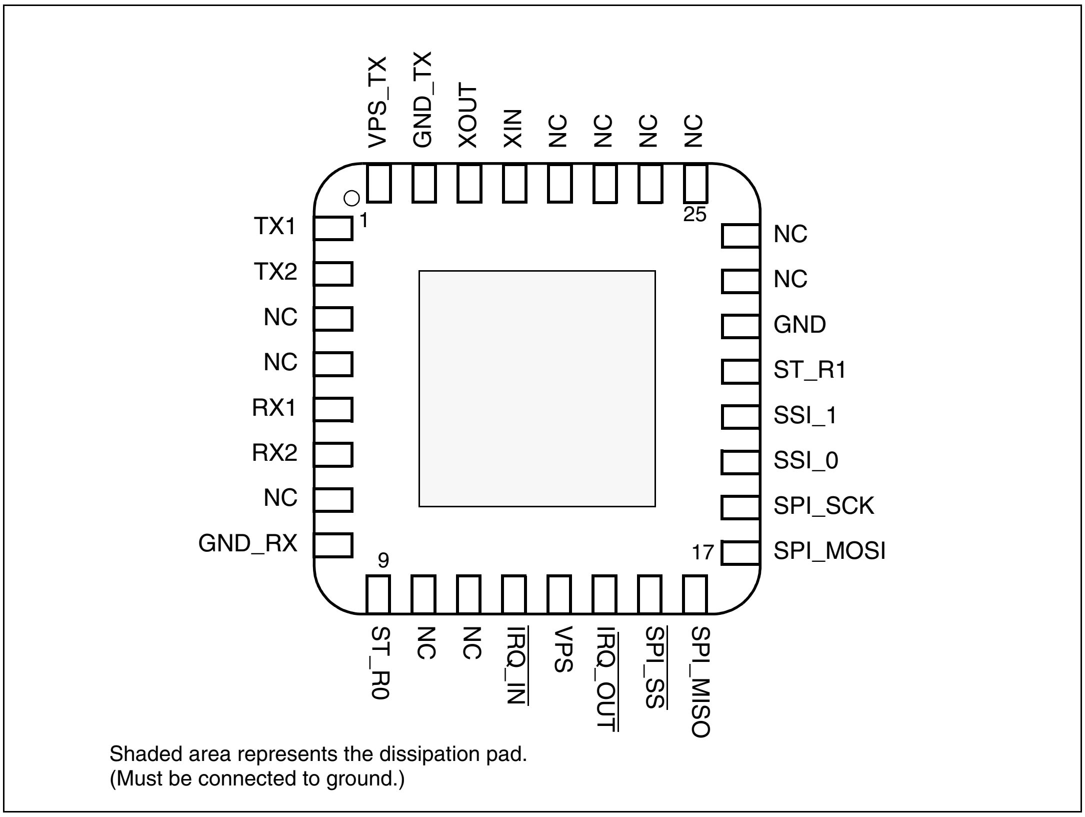
Table 1. List of terms

Term	Meaning
DAC	Digital analog converter
GND	Ground
HFO	High frequency oscillator
LFO	Low frequency oscillator
MCU	Microcontroller unit
MIFARE[1]	Communication protocol
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

1. MIFARE® and MIFARE® Classic are registered trademarks of NXP BV, and are used under license.

2 Pin and signal descriptions

Figure 3. ST95HF pinout description



Shaded area represents the dissipation pad.
(Must be connected to ground.)

Table 2. ST95HF pin descriptions

Pin	Pin name	Type[1]	Main function	Alternate function
1	TX1	O	Driver output 1	-
2	TX2	O	Driver output 2	-
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. ST95HF pin descriptions (continued)

Pin	Pin name	Type[1]	Main function	Alternate function
14	$\overline{\text{IRQ\_OUT}}$	O [4]	Interrupt output	-
15	$\overline{\text{SPI\_SS}}$	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	VPS_TX	P	Power supply (RF drivers)	-

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 ST95HF has 2 operating modes: Wait for Event (WFE) and Active. In Active mode, the ST95HF communicates actively with a tag or an external host (an MCU, for example). WFE mode includes four low consumption states: Power-up, Hibernate, Sleep/Field Detector and Tag Detector.

The ST95HF can switch from one mode to another.

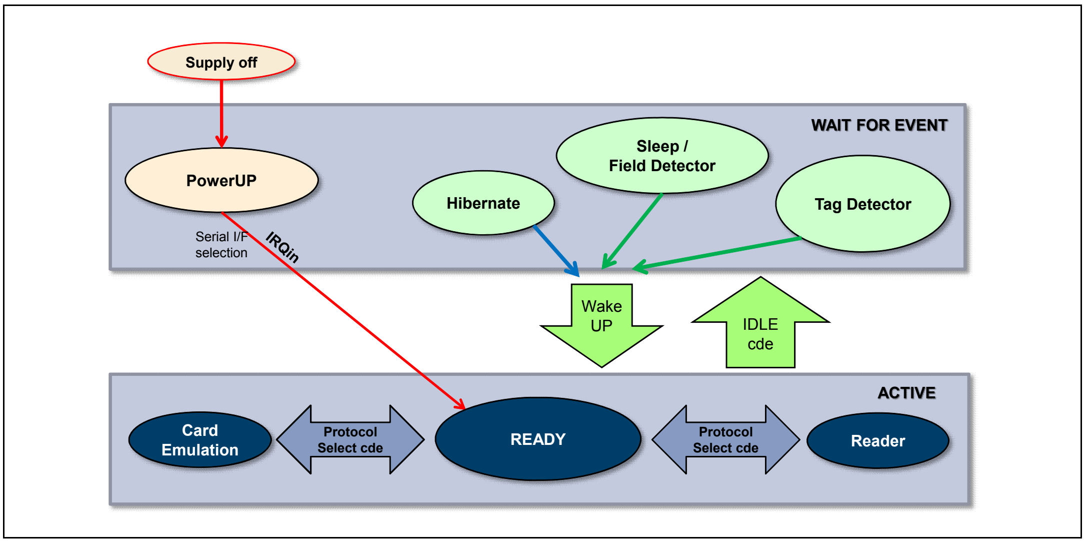
Table 3. ST95HF operating modes and states

Mode	State	Description
Wait For Event (WFE)	Power-up	This mode is accessible directly after POR. Low level on 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 ST95HF has to be woken-up in order to communicate. Low level on IRQ_IN pin (longer than 10 µs) is the only wakeup source.
	Sleep/Field Detector	Low power consumption state. Wakeup source is configurable: – Timer – IRQ_IN pin – SPI_SS pin – Field Detector LFO (low-frequency oscillator) is running in this state.
	Tag Detector	Low power consumption state with tag detection. Wakeup source is configurable: – Timer – IRQ_IN pin – SPI_SS pin – Tag detector LFO (low-frequency oscillator) is running in this state.
Active	Ready	In this mode, the RF is OFF and the ST95HF waits for a command (ProtocolSelect, ...) from the external host via the selected serial interface (SPI).
	Reader	The ST95HF can communicate with a tag using the selected protocol or with an external host using the SPI interface.
	Card Emulation	The ST95HF can communicate as a Card or Tag with an external reader. The Card or Tag application is located in the Host and communicates with the ST95HF via the SPI interface.

Hibernate, Tag Detector, 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 ST95HF can no longer communicate with the external host. It can only be woken up.

The behavior of the ST95HF in 'Tag Detector' state is defined by the Idle command.

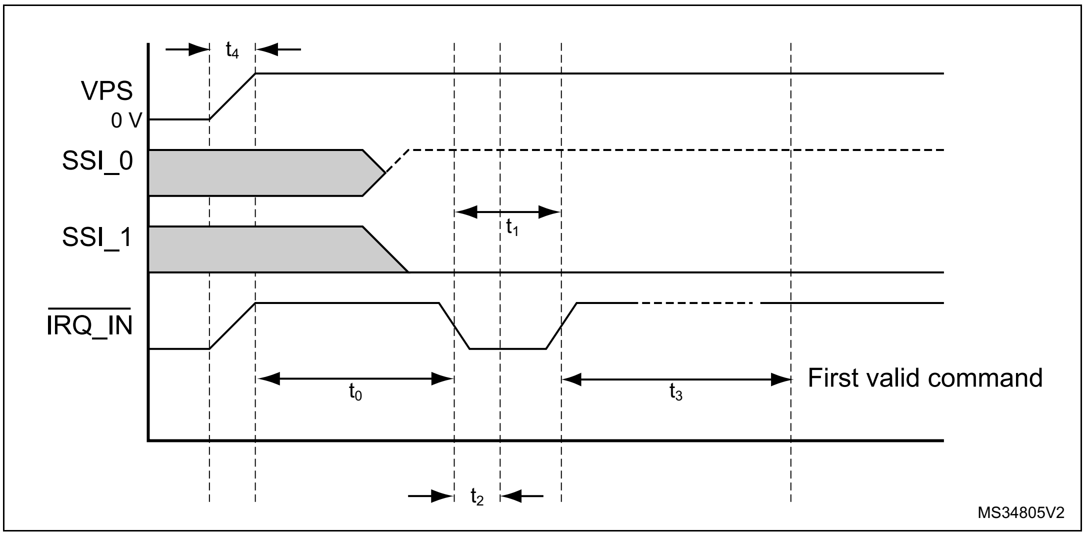
Figure 4. ST95HF initialization and operating state change



3.2 Startup sequence

After the power supply is established at power-on, the ST95HF waits for a low pulse on the pin 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. Note for pin SSI0: - - - SPI selected
2. Pin IRQ_IN low level < 0.2 VPS_Main.

Note: When ST95HF leaves WFE mode (from Power-up, Hibernate, Tag Detector, or Sleep/Field Detector) following an IRQ_IN low level pulse.

Figure 5 shows the power-up sequence for a ST95HF device; where,

- t_0 is the initial wake-up delay 100 µs (minimum)
- t_1 is the minimum interrupt width 10 µs (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} 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 $\overline{IRQ\_IN}$ 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
SSI_0	SPI: 1
SSI_1	SPI: 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 ST95HF.
2. Poll the ST95HF until it is ready to transmit the response.
3. Read the response.

The application software should never read data from the ST95HF without being sure that the ST95HF 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 ST95HF
- 0x03: Poll the ST95HF
- 0x02: Read data from the ST95HF
- 0x01: Reset the ST95HF

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

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

In Slave mode, the phase and polarization are defined with CPOL = 1 and CPHA = 1 or CPOL = 0 and CPHA = 0.

Figure 6. Sending command to ST95HF

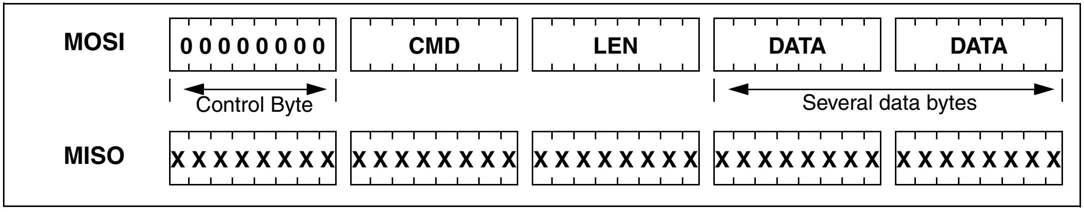


Figure 7. Polling the ST95HF until it is ready

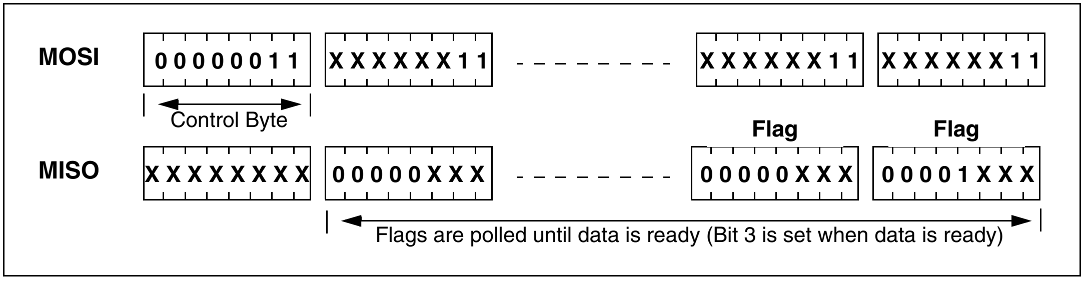
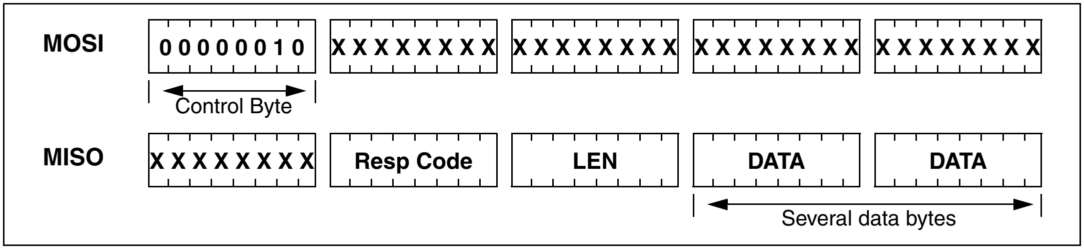


Table 5. Interpretation of flags

Bit	Meaning (application point of view)
[7:4]	Not significant
3	Data can be read from the ST95HF when set.
2	Data can be sent to the ST95HF when set.
[1:0]	Not significant

Figure 8. Reading data from ST95HF



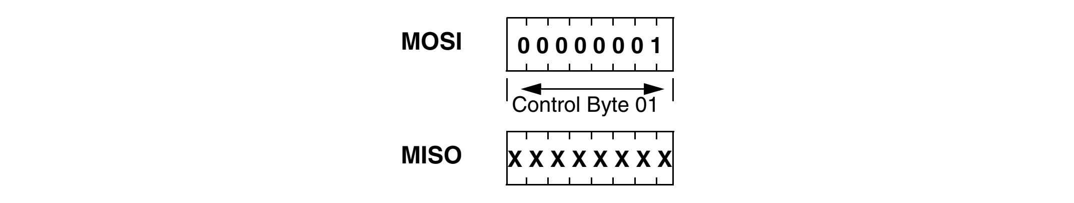
Data must be sampled at the rising edge of the SCK signal.

‘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 ST95HF, 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 ST95HF 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 ST95HF 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 ST95HF 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 ST95HF is ready.

Note that at the beginning of communication, the application does not need to check flags to start transmission. The ST95HF is assumed to be ready to receive a command from the application.

Figure 9. Reset the ST95HF



To reset the ST95HF 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 ST95HF into Power-up state. The ST95HF will wake up when pin IRQ_IN goes low. The ST95HF 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 ST95HF is configure to use the SPI serial interface, pin $\overline{\text{IRQ\_OUT}}$ is used to give additional information to user. When the ST95HF 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.**

4.2 Error codes

Table 6. Possible error codes and their meaning

Code	Name	Meaning
0x80	`EFrameRecvOK`	Frame correctly received (additionally see CRC/Parity information)
0x85	`EUserStop`	Stopped by user (used only in Card mode)
0x86	`ECommError`	Hardware communication error
0x87	`EFrameWaitTOut`	Frame wait time out (no valid reception)
0x88	`EInvalidSof`	Invalid SOF
0x89	`EBufOverflow`	Too many bytes received and data still arriving
0x8A	`EFramingError`	if start bit = 1 or stop bit = 0
0x8B	`EEgtError`	EGT time out
0x8C	`EInvalidLen`	Valid for ISO/IEC 18092, if Length <3
0x8D	`ECrcError`	CRC error, Valid only for ISO/IEC 18092
0x8E	`ERecvLost`	When reception is lost without EOF received (or subcarrier was lost)
0x8F	`ENoField`	When Listen command detects the absence of external field
0x90	`EUnintByte`	Residual bits in last byte. Useful for ACK/NAK reception of ISO/IEC 14443 Type A.

4.3 Support of long frames

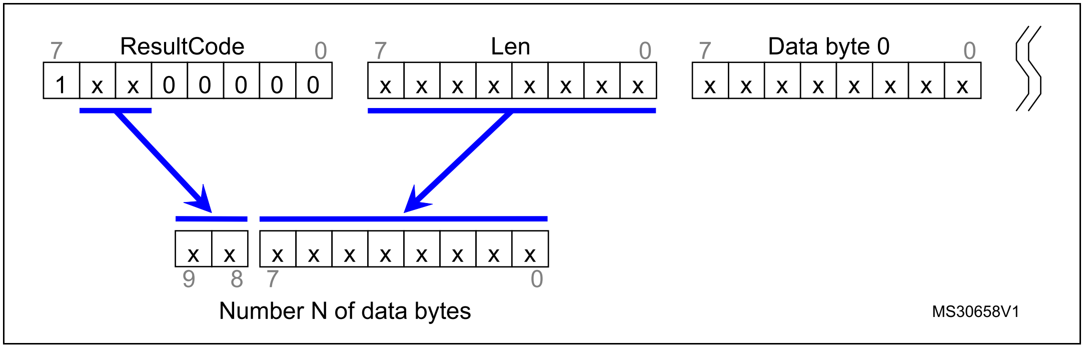
In Reader mode it is possible to receive up to 528 bytes of frame data from VICC and Type-B cards and up to 256 bytes of frame data from Type-A cards. In this case, the device sends a reply to the external MCU in the following format:

```
<ResultCode> + <Len> + <N bytes of data>
```

Table 7. Format of ResultCode

Bit	Meaning	
7	Always 1	
6	Bit 9 of Length	See examples and explanation below
5	Bit 8 of Length	
4	If set, there are residual bits in the last byte. Applicable only for Type-A protocol.	
3:0	Always 0	

Figure 10. Long frame format



The number of databytes is 10-bit long.

Table 8. Examples of ResultCode: Len pairs

ResultCode	Len	Length of data
0x80	0x00	0
0x80	0x01	1
0x80	0xFF	255
0xA0	0x00	256
0xA0	0x01	257
0xA0	0xFF	511
0xC0	0x00	512
0xC0	0x01	513

5 Commands

5.1 Command format

- The frame from the Host to the ST95HF has the following format:
  `<CMD><Len><Data>`
- The frame from the ST95HF 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 ST95HF
<<< Frame sent by the ST95HF to the Host

5.2 List of commands

Table 9 summarizes the available commands.

Table 9. List of ST95HF commands

Code	Command	Description
0x01	IDN	Requests short information about the ST95HF 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 (used in Card Emulation mode).
0x04	SendRecv	Sends data using the previously selected protocol and receives the tag response.
0x05	LISTEN	Listens for data using previously selected protocol (used in Card Emulation mode).
0x06	SEND	Sends data using previously selected protocol (used in Card Emulation mode).
0x07	IDLE	Switches the ST95HF into a low consumption Wait for Event (WFE) mode (Power-up, Hibernate, Sleep/Field or Tag Detection), 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 or ARC_B) register.

Table 9. List of ST95HF commands (continued)

Code	Command	Description
0x09	WrReg	Writes Analog Register Configuration (ACC_A or ARC_B)) register or writes index of ACC_A or ARC_B register address. Writes the Timer Window (TimerW) value dedicated to ISO/IEC 14443 Type A tags. Writes the AutoDetect Filter enable register dedicated to ISO/IEC 18092 tags.
0x0B	SubFreqRes	Returns the most recent subcarrier frequency detected for ISO/IEC 18092 communication.
0x0D	AcFilter	Enables or disables the anti-collision filter for ISO/IEC 14443 Type A protocol.
0x55	Echo	ST95HF 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 ST95HF and its revision.

Table 10. IDN command description

Direction	Data	Comments	Example
Host to ST95HF	0x01	Command code	>>>0x0100
	0x00	Length of data	
ST95HF to Host	0x00	Result code	<<<0x000F4E4643204653324A415354 34002ACE 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 ST95HF for communication with a reader or contactless tag.

Table 11. ProtocolSelect command description

Direction	Data	Comments	Example
Host to ST95HF	`0x02`	Command code	See *Table 12: List of <Parameters> values for the ProtocolSelect command for different protocols (Reader) on page 22*. See *Table 13: List of <Parameters> values for different protocols (Card Emulation) on page 25*.
	`<Len>`	Length of data	
	`<Protocol>`	Protocol codes (Reader): `00`: Field OFF `01`: ISO/IEC 15693 `02`: ISO/IEC 14443-A / NFC Forum Tag Type 1, Type 2, Type 4A `03`: ISO/IEC 14443-B / NFC Forum Tag Type 4B `04`: ISO/IEC 18092 / NFC Forum Tag Type 3 ------------------------------- Protocol codes (Card Emulation): `12`: ISO/IEC 14443-A `13`: ST Reserved (ISO/IEC 14443-B under qualification) `14`: ST Reserved (ISO/IEC 18092 under qualification)	
	`<Parameters>`	Each protocol has a different set of parameters. See *Table 12*.	
ST95HF to Host	`0x00`	Result code	`<<<0x0000` Protocol is successfully selected
	`0x00`	Length of data	
ST95HF to Host	`0x82`	Error code	`<<<0x8200` Invalid command length
	`0x00`	Length of data	
ST95HF to Host	`0x83`	Error code	`<<<0x8300` Invalid protocol
	`0x00`	Length of data	

Note that there is no ‘Field ON’ command. When the application selects an RF communication protocol, the field automatically switches ON if the Reader state is selected.

When the application selects a protocol, the ST95HF performs all necessary settings: it will choose the appropriate reception and transmission chains, switch ON or OFF the RF field and connect the antenna accordingly.

Different protocols have different sets of parameters. Values for the `<Parameters>` field are listed in *Table 12*.

Table 12. List of `<Parameters>` values for the ProtocolSelect command for different protocols (Reader)

Protocol	Code	Parameters			Examples of commands
		Byte	Bit	Function	
Field OFF	`0x00`	0	7:0	RFU	`>>>0x02020000`
ISO/IEC 15693	`0x01`	0	7:6	RFU	H 100 S: `>>>0x02 02 01 01` H 100 D: `>>>0x02 02 01 03` H 10 S: `>>>0x02 02 01 05` H 10 D: `>>>0x02 02 01 07` L 100 S: `>>>0x02 02 01 21` L 100 D: `>>>0x02 02 01 23` L 10 S: `>>>0x02 02 01 25` L 10 D: `>>>0x02 02 01 27` In these examples, the CRC is automatically appended.
			5:4	`00`: 26 Kbps (H) `01`: 52 Kbps `10`: 6 Kbps (L) `11`: RFU	
			3	`0`: Respect 312-µs delay `1`: Wait for SOF [1]	
			2	`0`: 100% modulation (100) `1`: 10% modulation (10)	
			1	`0`: Single subcarrier (S) `1`: Dual subcarrier (D)	
			0	Append CRC if set to '1'. [1]	
ISO/IEC 14443 Type A NFC Forum Tag Type 1 (Topaz) NFC Forum Tag Type 2 NFC Forum Tag Type 4A	`0x02`	0	7:6	Transmission data rate `00`: 106 Kbps `01`: 212 Kbps [2] `10`: 424 Kbps `11`: RFU	`>>>0x02020200`: ISO/IEC 14443 Type A tag, 106 Kbps transmission and reception rates, Time interval 86/90 Note that REQA, WUPA, Select20 and Select70 commands use a fixed interval of 86/90 µs between a request and its reply. Other commands use a variable interval with fixed granularity. Refer to the ISO/IEC 14443 standard for more details.
			5:4	Reception data rate `00`: 106 Kbps `01`: 212 Kbps [2] `10`: 424 Kbps `11`: RFU	
			3	RFU	
			2:0	RFU	
		1	7:0	PP	These 5 bytes are optional. The default PP:MM:DD value is 0 (corresponds to FDT 86/90µs) . For other values, FDT = (2^PP)*(MM+1)*(DD+128) *32/13.56 µs
		2	7:0	MM	
		3	7:0	DD (optional to PP:MM)	
		4	7:0	ST Reserved (Optional)	-
		5	7:0	ST Reserved (Optional)	-

Table 12. List of `<Parameters>` values for the ProtocolSelect command for different protocols (Reader) (continued)

Protocol	Code	Parameters			Examples of commands
		Byte	Bit	Function	
ISO/IEC 14443 Type B NFC Forum Tag Type 4B	0x03	0	7:6	Transmission data rate 00: 106 Kbps 01: 212 Kbps 10: 424 Kbps 11: 848 Kbps	>>>0x02020301: ISO/IEC 14443 Type B tag with CRC appended
			5:4	Reception data rate 00: 106 Kbps 01: 212 Kbps 10: 424 Kbps 11: 848 Kbps	
			3:1	RFU	
			0	Append CRC if set to ‘1’.[1]	
		1	7:0	PP	These 9 bytes are optional. Default value of PP:MM:DD is 0 and corresponds to FWT ~302µs. FWT = (2^PP)*(MM+1)*(DD+128)* 32/13.56 µs
		2	7:0	MM	
		3	7:0	DD (optional to PP:MM)	
		5:4	7:0	TTTT (Optional)	TR0 = TTTT/FC (LSB first), default 1023 = 0x3FF
		6	7:0	YY (Optional)	PCD Min TR1 (Min_TR1 = 8 * XX / f_S), default = 0
		7	7:0	ZZ (Optional)	PCD Max TR1 (Max_TR1 = 8 * ZZ / f_S), default = 26 = 0x1A
		8	7:0	ST Reserved (Optional)	-
		9	7:0	ST Reserved (Optional)	-

Table 12. List of <Parameters> values for the ProtocolSelect command for different protocols (Reader) (continued)

Protocol	Code	Parameters			Examples of commands
		Byte	Bit	Function	
ISO/IEC 18092 NFC Forum Tag Type 3	0x04	0	7:6	Transmission data rate 00: RFU 01: 212 Kbps 10: 424 Kbps 11: RFU	>>>0x02020451: ISO/IEC18092 tag, 212 Kbps transmission and reception rates with CRC appended. Parameter 'Slot counter' is not mandatory. If it is not present, it is assumed that SlotCounter = 0x00 (1 slot) For device detection commands, byte 1 bit 4 must be set to '0'. In this case, the FWT is 2.4 ms for the 1st slot and 1.2 ms more for each following slot, if slot counter is specified. If slot counter = 0x10, the ST95HF does not respect reply timings, but polls incoming data and searches a valid response during ~8.4 ms.
			5:4	Reception data rate 00: RFU 01: 212 Kbps 10: 424 Kbps 11: RFU	
			3:1	RFU	
			0	Append CRC if set to '1'.[1]	
		1	7:5	RFU	
			4	0: FWT = 2.4 ms 1: FWT is specified by PP:MM bits	
			3:0	Slot counter 0: 1 slot 1: 2 slots ... F: 16 slots	
		2	7:0	PP	These 3 bytes are optional. Default value PP:MM:DD: is 0 and corresponds to RWT ~302µs. RWT = (2^PP)*(MM+1)*(DD+128)*32/13.56µs
		3	7:0	MM	
		4	7:0	DD (optional to PP:MM)	

1. It is recommended to set this bit to '1'.
2. Not characterized.

Table 13. List of <Parameters> values for different protocols (Card Emulation)

Protocol (Card)	Code	Parameters			Examples of commands Comments
		Byte	Bit	Function	
ISO/IEC 14443 Type A [(1)]	0x12	0	7:6	Transmission data rate 00: 106 Kbps 01: 212 Kbps [(2)] 10: 424 Kbps [(2)] 11: RFU	>>>0x02021208 <<<0x0000 Card 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 [(2)] 10: 424 Kbps [(2)] 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. Topaz is not supported in Card Emulation mode.
2. Not qualified for this version.

5.5 Pollfield command (0x03) description

This command returns the current value of the FieldDet flag.

Table 14. POLLFIELD command description

Direction	Data	Comments	Example
Host to ST95HF	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 specidfied 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)	
ST95HF 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.	

The result of this command depends on the protocol selected. If we select a Reader mode protocol, the flag FieldDet is set to ‘1’ because the RF field is turned ON by the reader.

Table 15. Response for <POLLFIELD> command

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

5.6 Send Receive (SendRecv) command (0x04) description

This command sends data to a contactless tag and receives its reply.

Before sending this command, the Host must first send the PROTOCOLSELECT command to select an RF communication protocol.

If the tag response was received and decoded correctly, the <Data> field can contain additional information which is protocol-specific. This is explained in *Table 17*.

Table 16. SENDRECV command description

Direction	Data	Comments	Example
Host to ST95HF	0x04	Command code	See *Table 17* and *Table 26* for detailed examples.
	<Len>	Length of data	
	<Data>	Data to be sent	
ST95HF to Host	0x80	Result code	<<<0x800F5077FE01B3000000000071718EBA00 The tag response is decoded. This is an example of an ISO/IEC 14443 ATQB response (Answer to Request Type B)
	<Len>	Length of data	
	<Data>	Data received. Interpretation depends on protocol	
ST95HF to Host	0x90	Result code	<<<0x90040x240000 (exception for 4-bit frames where 'x' represents ACK or NAK value) 90: Result code for “non-integer number of bytes are received” 04: total length of data 0A or 00: Data 24: “2” means no CRC, “4” means 4 significant bits in Data byte. 00 00: No collision in response Example ACK <<< 0x90040A240000 Example NAK <<< 0x900400240000
	<Len>	Length of data	
	ACK or NAK	ISO 14443-A ACK or NAK detection	
	xx yy zz	3-byte response flag analysis	xx: Error type and number of significant bits in first data byte yy: First byte collision zz: First bit collision [(1)]
ST95HF to Host	X0 + <Len> + Data (see *Support of long frames on page 18*)		
ST95HF to Host	0x86	Error code	<<<0x8600 Communication error
	0x00	Length of data	
ST95HF to Host	0x87	Error code	<<<0x8700 Frame wait time out or no tag
	0x00	Length of data	

Table 16. SENDRECV command description (continued)

Direction	Data	Comments	Example
ST95HF to Host	0x88	Error code	<<<0x8800 Invalid SOF
	0x00	Length of data	
ST95HF to Host	0x89	Error code	<<<0x8900 Receive buffer overflow (too many bytes received)
	0x00	Length of data	
ST95HF to Host	0x8A	Error code	<<<0x8A00 Framing error (start bit = 0, stop bit = 1)
	0x00	Length of data	
ST95HF to Host	0x8B	Error code	<<<0x8B00 EGT time out (for ISO/IEC 14443-B)
	0x00	Length of data	
ST95HF to Host	0x8C	Error code	<<<0x8C00 Invalid length. Used in NFC Forum Tag Type 3, when field Length < 3
	0x00	Length of data	
ST95HF to Host	0x8D	Error code	<<<0x8D00 CRC error (Used in NFC Forum Tag Type 3 protocol)
	0x00	Length of data	
ST95HF to Host	0x8E	Error code	<<<0x8E00 Reception lost without EOF received
	0x00	Length of data	

1. See *Table 17* for details.

Table 17 gives examples of communication between the ST95HF and a contactless tag. The ST95HF receives a SendRecv command (>>> 0x04...) from the host and returns its response to the host (<<< 0x80...). *Table 17* provides more details on the ST95HF response format.

Table 17. List of <Data> Send values for the SENDRECV command for different protocols

Protocol	Explanation	Command example			Comments
ISO/IEC 15693	Send example	04	03	022000	Example of an Inventory command using different protocol configuration: Uplink: 100% ASK, 1/4 coding Downlink: High data rate, Single sub-carrier >>> 0x0403260100 (Inventory - 1 slot) <<< 0x800D0000CDE0406CD62902E0057900 If length of data is ‘0’, only the EOF will be sent. This can be used for an anti-collision procedure.
	Command code				
	Length of entire data field				
	Data				

Table 17. List of `<Data>` Send values for the SendRecv command for different protocols (continued)

Protocol	Explanation	Command example				Comments

ISO/IEC 14443 Type A
NFC Forum Tag Type 4A
NFC Forum Tag Type 1 (Topaz)
NFC Forum Tag Type 2
	Send example	04	07	9370800
				F8C8E
					28	Example of an NFC Forum Type 2 request sequence:
						`>>>0x04022607` (REQA)
						`<<<0x800544002800` (ATQA)
						`>>>0x0403932008` (Anti-collision CL1)
						`<<<0x80088804A8D5F1280000` (UID CL1)...
						
						Example of an NFC Forum Type 1 (Topaz) request sequence:
						`>>>0x04022607` (REQA)
						`<<<0x8005000C280000` (ATQ0 ATQ1)
						`>>>0x0408780000000000000A8` (RID)
						`<<<0x800B11486E567A003E45080000` (Header0 Header1 UID0 UID 1 UID2 UID3 CRC0 CRC1Signifcant bits indexColbyte IndexColbit)
						
						Application SW must specify how many bits to send in the last byte. If flag SplitFrame is set, ST95HF will expect 8 – <significant bit count> bits in the 1st byte during reception.
						In this case, the first byte received is padded with zeros in lsb to complete the byte, while the last byte received is padded with zeros in msb.
						
						Example of an anti-collision command /response in ISO/IEC 14443_A communication using a Split frame: (1)
						>>> `0x0403932008` (Anticol)
						<<< `0x800888047B75B7`**`B8`**`0204` (Collision Detected B8)
						>>> `0x0406934588040B`**`45`** (Anticol Split frame request 45)
						<<< `0x80064074B3`**`23`**`0000` (Spilt frame Answer 23)
						
						This command is useful for anti-collision.

	Command code					
	Length of entire data field					
	Data					
	Transmission flags: 7: Topaz send format. Use EOF instead of parity bit and use SOF at beginning of each byte. Pause between bytes and assume 1st byte is 7 bits. 6: SplitFrame 5: Append CRC 4: Parity Framing mode (2) [3:0]: Number of significant bits in last byte					

ISO/IEC 14443 Type B
NFC Forum Tag Type 4B
	Send example	04	03	050000		Example of an NFC Forum Type 4B request sequence:
						`>>>0x0403050000 (REQB)`
						`<<<0x800F5077FE01B3000000000071718EBA00 (ATQB)`

	Command code					
	Length of entire data field					
	Data					

Table 17. List of `<Data>` Send values for the SendRecv command for different protocols (continued)

Protocol	Explanation	Command example	Comments

ISO/IEC 18092
NFC Forum Tag Type 3
	Send example	`04 05 00FFFF0000`	Example of an ISO/IEC 18092 / NFC Forum Type 3 request sequence:
			`>>>0x040500FFFF0000 (REQC)`
			`<<<0x801201010102148E0DB413100B4B428485D0FF00 (ATQC)`

	Command code	`04`	
	Length of entire data field	`05`	
	Data	`00FFFF0000`	

1. For more information on using split frames, refer to *Appendix D on page 69*.

2. If Parity Framing mode is used (Bit 4 of transmission flag byte is set to ‘1’), then the parity bit must be coded inside the data for each byte to be sent using the send/receive command in Transmit mode, and is not decoded by the ST95HF in Receive mode. In Receive mode, each data byte is accompanied by an additional byte which encodes the parity: <data byte> <parity byte> <data byte > ...
Examples of data received by send / receive in Parity Framing mode:
80 05 32 80 34 00 00
meaning: if the ST95HF received 2 data bytes:
0x32 with parity = ‘1’ (0x80) and 0x34 with parity = ‘0’ (0x0) in Parity Framing mode. For more details, see *NFC Forum Tag Type 2 on page 75*.
The Parity Framing mode is compatible with MIFARE® Classic requirements. However, access to Authenticated state must be supported by the external secure host which embeds the MIFARE® Classic library.

Figure 11. Data transfer (in both command and response) when Parity Framing mode is enabled

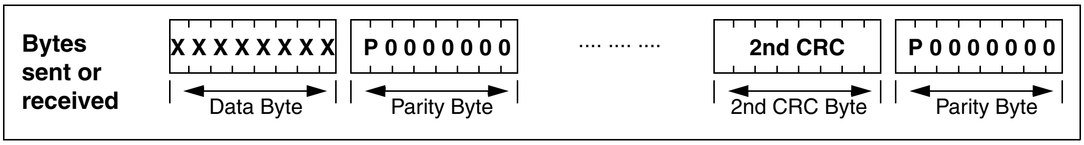


Table 18. List of `<Data>` Response values for the SendRecv command for different protocols

Protocol	Explanation	Response example	Comments
ISO/IEC 15693	Response example	`80 08 0000000000 77CF 00`	This is a response to Read Single Block command for ISO/IEC 15693 TAG. Actual TAG response is `<<<0x000000000077CF`, other fields are added by the ST95HF.
	Result code	`80`	
	Length of entire data field	`08`	
	Data received from tag	`0000000000`	
	Original (received) value of CRC	`77CF`	
	[7:2]: RFU 1: CRC error if set 0: Collision is detected if set	`00`	

Table 18. List of <Data> Response values for the SendRecv command for different protocols (continued)

Protocol	Explanation	Response example	Comments
ISO/IEC 14443 Type A NFC Forum Tag Type 4A NFC Forum Tag Type 1 (Topaz) NFC Forum Tag Type 2	Response example	80 or 90 [1] 09 80B30B8DB500 00 00 00	ISO/IEC 14443-A is bit oriented protocol, so we can receive non-integer amount of bytes. Number of significant bits in the 1st byte is the same as indicated in the command sent. To calculate a position of a collision, application has to take index of byte first. Index of bit indicates a position inside this byte. Note that both indexes start from 0 and bit index can be 8, meaning that collision affected parity. Note that collision information is only valid when bit 'Collision is detected' is set. [2]
	Result code	80 or 90 [1]	
	Length of entire data field	09	
	Data received from TAG	80B30B8DB500	
	7: Collision is detected 6: RFU 5: CRC error 4: Parity error [3:0]: Shows how many significant bits are there in the first byte	00	
	7:0: Index of the first byte where collision is detected	00	
	[7:4]: RFU [3:0]: Index of the first bit where collision is detected	00	
ISO/IEC 14443 Type B NFC Forum Tag Type 4B	Response example	80 0F 5092036A8D00000000007171 3411 00	-
	Result code	80	
	Length of entire data field	0F	
	Data received from tag	5092036A8D00000000007171	
	Original (received) value of CRC	3411	
	[7:2]: RFU 1: CRC error if set 0: RFU	00	
ISO/IEC 18092 NFC Forum Tag Type 3	Response example	80 12 01010105017B0...93FF 00	<<<0x801201010105017B06941004014B024F4993FF00
	Result code	80	
	Length of entire data field	12	
	Data received from tag	01010105017B0...93FF	
	[7:2]: RFU 1: CRC error if set 0: RFU	00	

1. Result code 90: Response is decoded but number of byte is not an integer.
2. For more information on using split frames, refer to *Appendix D on page 69*.

For more detailed examples of use with NFC Forum and ISO/IEC 15693 tags, refer to *Appendix D on page 69*.

If Parity Framing mode is used, the parity bit stays unchanged. On transmission, it is not encoded and on reception it is not decoded. The length of Data must be even. Each data byte is accompanied by an additional byte which encodes the parity:

```
<DataByte>, <Parity>, <DataByte>, <Parity> …
```

Table 19. Structure of Parity byte

Bit	Description
7	Parity bit
[6:0]	Reserved for future use

On reception, bits [6:0] of the parity byte are zeroes; on transmission, bits [6:0] are ignored.

5.7 Listen command (0x05) description

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

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

Table 20. LISTEN command description

Direction	Data	Comments	Example
Host to ST95HF	05	Command code	0x0500: Enters a Listening mode where the ST95HF waits for a command from an external reader.
	00	Length of data	
ST95HF to Host	00	Result code	0x0000: No error. Confirmation that ST95HF now is in Listening mode.
	00	Length of data	
ST95HF to Host	82	Error code	0x8200: Invalid command length
	00	Length of data	
ST95HF to Host	83	Error code	0x8300: Invalid protocol or protocol is not supported. For example, application selects protocol ISO/IEC 15693 using command select protocol and then executes the LISTEN command. ISO/IEC 15693 is not supported in Card Emulation mode.
	00	Length of data	
ST95HF 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 ST95HF 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 ST95HF buffer. The ST95HF 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 42*).

In all cases, the ST95HF 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 ST95HF sends 0x55 + 0x8500 (error code of the Listening state cancelled by the MCU).

Possible return codes are listed in *Table 21*.

Table 21. Response codes from the ST95HF in Listening mode

Direction	Data	Comments	Example
ST95HF to Host	`80`	Result code	`<<<0x800605000071FF00` The request from the Reader is decoded. This is an example of a response in ISO/IEC 14443 Type B protocol.
	`<Len>`	Length of data	
	`<Data>`	Data received. Interpretation depends on protocol	
ST95HF to Host	`85`	Error code	`<<<0x8500` Listening mode was cancelled by the application
	`00`	Length of data	
ST95HF to Host	`86`	Error code	`<<<0x8600` Communication error
	`00`	Length of data	
ST95HF to Host	`88`	Error code	`<<<0x8800` Invalid SOF
	`00`	Length of data	
ST95HF to Host	`89`	Error code	`<<<0x8900` Receive buffer overflow (too many bytes received)
	`00`	Length of data	
ST95HF to Host	`8A`	Error code	`<<<0x8A00` Framing error (start bit=0, stop bit=1)
	`00`	Length of data	
ST95HF to Host	`8B`	Error code	`<<<0x8B00` EGT time out (for ISO/IEC 14443-B)
	`00`	Length of data	
ST95HF to Host	`8D`	Error code	`<<<0x8D00` CRC error (Used in NFC Forum Tag Type 3 protocol)
	`00`	Length of data	
ST95HF - 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 ST95HF will send data back to the Host (Card Emulation application). This is explained in *Table 22*.

Table 22. 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							

5.8 Send command (0x06) description

This command immediately sends data to the reader without waiting for a reply.

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

Table 23. SEND command description

Direction	Data	Comments	Example
Host to ST95HF	06	Command code	Depends on protocol previously selected! >>>0x0606D07387080028: Emulation of TAG response in ISO/IEC 14443 Type B protocol
	<Len>	Length of data	
	<Data>	Data and additional parameter to be sent	
ST95HF to Host	00	Result code	<<<0x0000 Data was successfully sent
	00	Length of data	
ST95HF to Host	82	Error code	<<<0x8200 Invalid length (for example, Length=0 where it is not possible)
	00	Length of data	
ST95HF to Host	83	Error code	<<<0x8300 Invalid protocol previously selected by Select Protocol command
	00	Length of data	

Table 24. 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					
ISO/IEC-14443 Type B	Send example	06	04	01020304		-
	Command code					
	Length of entire data field					
	Data					
Reader ISO/IEC-18092 212/424	Send example	06	04	01020304		Note the difference in data in Reader and Card Emulation mode: in Reader mode there is no slot information.
	Command code					
	Length of entire data field					
	Data					

5.9 Idle command (0x07) description

This command switches the ST95HF 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 25. IDLE command description

Direction	Data	Comments	Example

Host to ST95HF	`07`	Command code	Example of switch from Active mode to Hibernate state:
			`>>>0x07 0E 08 04 00 04 00 18 00 00 00 00 00 00 00 00`
			Example of switch from Active to WFE mode (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`
			Example of switch from Active to WFE mode (wake-up by low pulse on SPI_SS pin):
			`>>>0x07 0E 10 01 00 38 00 18 00 00 60 00 00 00 00 00`
			Example of switch from Active mode to WFE mode (Sleep / Field Detector with wake-up by Field Detection):
			`>>> 0x07 0E 04 01 42 38 00 18 00 00 60 00 00 00 00`
			Example of wake-up by Timeout (7 seconds):
			Duration before Timeout = 256 * t_L * (WU period + 2) * (MaxSleep + 1)
			`>>>0x07 0E 01 21 00 38 00 18 00 60 60 00 00 00 00 08`
			Example of switch from Active to Tag Detector mode (wake-up by tag detection or low pulse on IRQ_IN pin) (32 kHz, inactivity duration = 272 ms, DAC oscillator = 3 ms, Swing = 63 pulses of 13.56 MHz):
			`>>>0x07 0E 0A 21 00 79 01 18 00 20 60 60 64 74 3F 08`
			Example of a basic Idle command used during the Tag Detection Calibration process:
			`>>>0x07 0E 03 A1 00 F8 01 18 00 20 60 60 00 xx 3F 01`
			where `xx` is the DacDataH value.

	`0E`	Length of data	
	`<WU Source>`	Specifies authorized wake-up sources and the LFO frequency	
	`EnterCtrlL`	Settings to enter WFE mode	
	`EnterCtrlH`		
	`WUCtrlL`	Settings to wake-up from WFE mode	
	`WUCtrlH`		
	`LeaveCtrlL`	Settings to leave WFE mode (Default value = 0x1800)	
	`LeaveCtrlH`		
	`<WUPeriod>`	Period of time between two tag detection bursts. Also used to specify the duration before Timeout.	
	`<OscStart>`	Defines the Wait time for HFO to stabilize: `<OscStart> * tL` (Default value = 0x60)	
	`<DacStart>`	Defines the Wait time for DAC to stabilize: `<DacStart> * tL` (Default value = 0x60)	
	`<DacDataL>`	Lower compare value for tag detection (1). This value must be set to `0x00` during tag detection calibration.	
	`<DacDataH>`	Higher compare value for tag detection (1). This is a variable used during tag detection calibration.	
	`<SwingsCnt>`	Number of swings HF during tag detection (Default value = 0x3F)	
	`<MaxSleep>`	Max. number of tag detection trials before Timeout (1). This value must be set to `0x01` during tag detection calibration. Also used to specify duration before Timeout. MaxSleep must be: 0x00 < MaxSleep < 0x1F	

Table 25. IDLE command description (continued)

Direction	Data	Comments	Example
ST95HF to Host	`0x00`	Result code	This response is sent only when ST95HF exits WFE mode. `<<<0x000101` Wake-up by Timeout `<<<0x000102` Wake-up by tag detect `<<<0x000108` Wake-up by low pulse on IRQ_IN pin
	`0x01`	Length of data	
	`<Data>`	Data (Wake-up source): 0x01: Timeout 0x02: Tag detect 0x08: Low pulse on IRQ_IN pin 0x10: Low pulse on SPI_SS pin	
ST95HF to Host	`0x82`	Error code	`<<<0x8200` Invalid command length
	`0x00`	Length of data	

1. An initial calibration is necessary to determine DacDataL and DacDataH values required for leaving Tag Detector state. For more information, contact your ST sales office for the corresponding application note.

5.9.1 Idle command parameters

The Idle command (Host to ST95HF) has the following structure (all values are hexadecimal):

Table 26. Idle command structure

`07`	`0E`	`xx`	`yy zz`	`yy zz`	`yy zz`	`aa`	`bb`	`cc`	`dd ee`	`ff`	`gg`
Command code	Data length	WU source	Enter Control	WU Control	Leave Control	WU Period	Osc Start	DAC Start	DAC Data	Swing Count	Max Sleep

Table 27. 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: `0x01`: Time out `0x02`: Tag Detection `0x04`: Field Detector `0x06`: Not defined `0x08`: Low pulse on IRQ_IN `0x10`: Low pulse on SPI_SS
Enter Control	These two bytes (EnterCtrlL and EnterCtrlH) define the resources when entering WFE mode. `0x0400`: Hibernate `0x0100`: Sleep (or 0x`2100` if Timer source is enabled) `0x0142`: Sleep (if Field Detector source is enabled) `0xA200`: Tag Detector Calibration `0x2100`: Tag Detection

Table 27. Summary of Idle command parameters (continued)

Parameter	Description
WU Control	These two bytes (WuCtrlL and WuCtrlH) define the wake-up resources. 0x0400: Hibernate 0x3800: Sleep/Field Detector 0xF801: Tag Detector Calibration 0x7901: Tag Detection
Leave Control	These two bytes (LeaveCtrlL and LeaveCtrlH) define the resources when returning to Ready state. 0x1800: Hibernate 0x1800: Sleep/Field Detector 0x1800: Tag Detector Calibration 0x1800: Tag Detection
WU Period	This byte is the coefficient used to adjust the time allowed between two tag detections. Also used to specify the duration before Timeout. (Typical value: 0x20) Duration before Timeout = 256 * t_L * (WU period + 2) * (MaxSleep + 1)
Osc Start	This byte defines the delay for HFO stabilization. (Recommended value: 0x60) Defines the Wait time for HFO to stabilize: `<OscStart> * tL`
DAC Start	This byte defines the delay for DAC stabilization. (Recommended value: 0x60) Defines the Wait time for DAC to stabilize: `<DacStart> * tL`
DAC Data	These two bytes (DacDataL and DacDataH) define the lower and higher comparator values, respectively. These values are determined by a calibration process. When using the demo board, these values should be set to approximately 0x64 and 0x74, respectively.
Swing Count	This byte defines the number of HF swings allowed during Tag Detection. (Recommended value: 0x3F)
Max Sleep	This byte defines the maximum number of tag detection trials or the coefficient to adjust the maximum inactivity duration before Timeout. MaxSleep must be: 0x00 < MaxSleep < 0x1F This value must be set to 0x01 during tag detection calibration. Also used to specify duration before Timeout. Duration before Timeout = 256 * t_L * (WU period + 2) * (MaxSleep + 1) (Typical value: 0x28)

5.9.2 Using LFO frequency setting to reduce power consumption

In WFE mode, the high frequency oscillator (HFO) is stopped and most processes being executed are clocked by the low frequency oscillator (LFO). To minimize ST95HF power consumption in WFE mode, the slower the LFO frequency, the lower the power consumption.

Example 1: Setting a lower LFO frequency

The following equation defines a basic timing reference:

t_{REF} = 256*t_L ms (where t_L = 1/f_{LFO})
t_{REF} = 8 ms (when bits [7:6] are set to “00”, or 32 kHz)
t_{REF} = 64 ms (when bits [7:6] are set to “11”, or 4 kHz)

5.9.3 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 a tag is detected (bit 1 set to ‘1’) or on a low pulse on pin IRQ_IN (bit 3 set to ‘1’), set the register to 0x0A.

5.9.4 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 ST95HF to Ready state. Some methods are nearly automatic, such as waiting for a timer overflow or a tag 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 ST95HF enters Power-up state.

To wake up the ST95HF and set it to Ready state, the user must send a low pulse on the IRQ_IN pin. The ST95HF 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 ST95HF can wake-up from Hibernate state is by an external event (low pulse on pin IRQ_IN).

A basic Idle command is:

```
>>>0x07 0E 08 04 00 04 00 18 00 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 IRQ_IN or SPI_SS pin)

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 IRQ_IN:

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

A similar command can be implemented using pin SPI_SS as a wake-up source:

```
>>>0x07 0E 10 01 00 38 00 18 00 00 60 00 00 00 00 00
```

Wake-up by timeout

The LFO is required to use the timer. However, this increases the typical power consumption by 80 µA. Several parameters can be modified to reduce power consumption as much as possible.

The Duration before Timeout is defined by parameters WU period and MaxSleep, respectively 0x60 and 0x08 in the following example.

Duration before Timeout = 256 * t_L * (WU period + 2) * (MaxSleep + 1)

Note: Note that: 0x00 < MaxSleep < 0x1F.

An Idle command example when wake-up source is timer (`0x01`) when f_{LFO} = 32 kHz (mean power consumption is 25 µA)

```
>>>0x07 0E 01 21 00 38 00 18 00 60 60 00 00 00 00 08
```

An Idle command example when wake-up source is timer (`0xC1`) when f_{LFO} = 4 kHz (mean power consumption is 20 µA):

```
>>>0x07 0E C1 21 00 38 00 18 00 60 60 00 00 00 00 08
```

The same command can be used mixing a timer and the $\overline{\text{IRQ\_IN}}$ pin (`0xC9`) as a wake-up source:

```
>>>0x07 0E C9 21 00 38 00 18 00 60 60 00 00 00 00 08
```

Wake-up by Tag Detection

In this mode, the typical consumption can greatly vary in function of parameter settings (WU period without RF activity and Swing Count defining the RF burst duration). Using default settings, consumption in the range of 100 µA can be achieved.

Tag Detector is a state where ST95HF is able to detect an RF event, a wake-up will occur when a tag sufficiently modifies the antenna load and is detected by the ST95HF.

An Idle command example when wake-up source is Tag Detection (`0x02`):

```
>>>0x07 0E 02 21 00 79 01 18 00 20 60 60 64 74 3F 08
```

The same command can be used mixing Tag Detection and the $\overline{\text{IRQ\_IN}}$ pin (`0x0A`) as a wake-up source:

```
>>>0x07 0E 0A 21 00 79 01 18 00 20 60 60 64 74 3F 08
```

The tag detection sequence is defined by dedicated parameters:

- WU source (Byte 3) (*Wake-up source register on page 64*)
  - The Timeout bit (bit 0) must be set to ‘1’ in order to manage a certain number of emitted bursts. Otherwise, bursts will be sent indefinitely until a stop event occurs (for example, tag detection or a low pulse on pin $\overline{\text{IRQ\_IN}}$).
  - The Tag Detect bit (bit 1) must be set to ‘1’ to enable RF burst emissions.
  - It is recommended to also set Bits 3 or 4 to ‘1’ to ensure that it is possible to leave Tag Detect mode via an external event (for example, a low pulse on pin $\overline{\text{IRQ\_IN}}$).
- WU period (Byte 10): Defines the period of inactivity ($t_{INACTIVE}$) between two RF bursts:

  $t_{INACTIVE}$ = (WuPeriod + 2) * t_{REF}
- OscStart, DacStart (Bytes 11 and 12): Define the set-up time of the HFO and Digital Analog Converter, respectively. In general, 3 ms is used both set-up times.

  HFO | DAC set-up time = (OscStart | DacStart) * t_L
- DacDataL, DacDataH (Bytes 13 and 14): Reference level for Tag Detection (calculated during the tag detection calibration process).
- SwingsCnt (Byte 15): Represents the number of 13.56-MHz swing allowed during a Tag Detection burst. We recommend using `0x3F`.
- Maxsleep (Byte 16): The ST95HF emits (MaxSleep +1) bursts before leaving Tag Detection mode if bit 0 (Timer Out) of the WU source register is set to ‘1’. Otherwise, when this bit is set to ‘0’, a burst is emitted indefinitely.

Note: Bytes 4 to 9 should be used as shown in the examples in Section 5.9: Idle command (0x07) description.

Note that the MaxSleep value is coded on the 5 least significant bits, thus:
0x00 < MaxSleep < 0x1F.

All the previously described command parameters must be chosen accordingly for the initial tag detection calibration when setting up the ST95HF.

Their value will impact tag detection efficiency, and ST95HF power consumption during Tag Detection periods.

5.9.5 Tag detection calibration procedure

The Idle command allows the use of a tag detection as a wake-up event. Certain parameters of the Idle command are dedicated to setting the conditions of a tag detection sequence.

During the tag detection sequence, the ST95HF regularly emits RF bursts and measures the current in the antenna driver I_{DRIVE} using the internal 6-bit DAC.

When a tag enters the ST95HF antenna RF operating volume, it modifies the antenna loading characteristics and induces a change in I_{DRIVE}, and consequently, the DAC data register reports a new value.

This value is then compared to the reference value established during the tag detection calibration process. This enables the ST95HF to decide if a tag has entered or not its operating volume.

The reference value (DacDataRef) is established during a tag detection calibration process using the ST95HF application setting with no tag in its environment.

The calibration process consists in executing a tag detection sequence using a well-known configuration, with no tag within the antenna RF operating volume, to determine a specific reference value (DacDataRef) that will be reused by the host to define the tag detection parameters (DacDataL and DacDataH).

During the calibration process, DacDataL is forced to 0x00 and the software successively varies the DacDataH value from its maximum value (0xFE) to it minimum value (0x00). At the end of the calibration process, DacDataRef will correspond to the value of DacDataH for which the wake-up event switches from Timeout (no tag in the RF field) to tag detected.

To avoid too much sensitivity of the tag detection process, we recommend using a guard band. This value corresponds to 2 DAC steps (0x08).

Recommended guard band value:

DacDataL = *DacDataRef* – *Guard* and *DacDataH* = *DacDataRef* + *Guard*

The parameters used to define the tag detection calibration sequence (clocking, set-up time, burst duration, etc.) must be the same as those used for the future tag detection sequences.

When executing a tag detection sequence, the ST95HF compares the DAC data register value to the DAC Data parameter values (DacDataL and DacDataH) included in the Idle command. The ST95HF will exit WFE mode through a Tag Detection event if the DAC data register value is greater than the DAC Data parameter high value (DacDataH) or less than the DAC Data parameter low value (DacDataL). Otherwise, it will return to Ready state after a Timeout.

An efficient 8-step calibration algorithm is described in *Example of tag detection calibration process on page 65*.

An example of a basic Idle command used during the Tag Detection Calibration process:

```
>>>0x07 0E 03 A1 00 F8 01 18 00 20 60 60 00 xx 3F 01
```

where *xx* is the DacDataH value.

An example of a tag detection sequence is provided in *Example of tag detection command using results of tag detection calibration on page 68*.

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

This command is used to read the Wakeup register.

Table 28. RdReg command description

Direction	Data	Comments	Example
Host to ST95HF	`0x08`	Command code	Ex 1. >>>`0x0803690100` Reads the ACC_A or ARC_B register. [(1)]
	`0x03`	Length of data	
	`0x62` or `0x69`	Register address	
	`0x01`	Register size	
	`0x00`	ST Reserved	
ST95HF to Host	`0x00`	Result code	`<<<0x000101` Wake-up by Timeout (Ex. 1) `<<<0x000102` Wake-up by Tag Detect (Ex. 1) `<<<0x000113` Depth = 1, Gain = 3 (Ex. 2) `<<< 0x000127` (ACC_A register) Default value for Demodulator Sensitivity and Load Modulation. See Write Register description for more information on received data. `<<< 0x000113` (ARC_B register) Depth = 1, Gain = 3 (Ex. 2). See Write Register description for more information on received data.
	`<Len>`	Length of data (= RegCount)	
	`<RegData>`	Register data	
ST95HF 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`) or by setting the ARC_B register index (`0x0903680001`) .

Note: *The Management of the Analog Register Configuration register* (ACC_A or ARC_B*) is described in Section 5.11: Write Register (WrReg) command (0x09) description.*

5.11 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 (ARC_B) 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.
- set the Timer Window (TimerW) value used to improve ST95HF demodulation when communicating with ISO/IEC 14443 Type A tags
- set the AutoDetect Filter used to help synchronization of ST95HF with ISO/IEC 18092 tags
- configure the HF2RF bit[a] to manage I_{CC} RF (V_{PS_TX}) consumption in Ready state

5.11.1 Improving RF performance

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

Modulation Index and Receiver Gain parameters for Reader mode

The default values of these parameters (*Table 35*) are set by the PROTOCOLSELECT command, but they can be overwritten using the Write Register (WRREG) command (`0x09`). *Table 31* and *Table 32* list possible values for the Modulation Index and Receiver Gain parameters, respectively.

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

[a] When the HF2RF bit is ‘0’, Reader mode is possible (default mode). When set to ‘1’, V_{PS_TX} power consumption is reduced (Ready mode).

Table 29. WrReg command description (Modulation Index and Receiver Gain)

Direction	Data	Comments	Example

Host to ST95HF	`0x09`	Command code	`>>>0x090468010113`
			Update ARC_B value to 0x13
			`>>>0x0903680001`
			Set Analog Register Index to 0x01 (ARC_B) [1]

	`0x03` or `0x04`	Length of data	
	`0x68`	Analog Register Configuration address index	
	`0x00` or `0x01`	Flag Increment address or not after Write command	
	`0x01`	Index pointing to the Modulation Index and Receiver Gain values in the ARC_B register (0x01) (see *Section 5.11.1*)	
	`0xXX`	New value for Modulation Index and Receiver Gain nibbles (see *Section 5.11.1*)	
ST95HF to Host	`0x00`	Result code	<<<0x0000 Register written
	`0x00`	Length of data (= `RegCount`)	

[1] This command must be executed before reading the ARC_B register (`0x0803690100`).

Load Modulation Index and Demodulator Sensitivity parameters for Card Emulation mode

The default value of these parameters (*Table 35*) is set by the ProtocolSelect command, but they can be overwritten using the Write Register (WrReg) command (`0x09`). *Table 33* and *Table 34* 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 30. WrReg command description (Load Modulation Index and Demodulator Sensitivity)

Direction	Data	Comments	Example

Host to ST95HF	`0x09`	Command code	`>>>0x0903680004`
			Set Analog Register Index to 0x04 (ACC_A) [1]
			`>>>0x090468010425`
			Update ACC_A to 0x25

	`0x03` or `0x04`	Length of data	
	`0x68`	Analog Register Configuration address index	
	`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 5.11.1*)	
	`0xXX`	New value for Demodulator Sensitivity and Load Modulation (see *Section 5.11.1*)	

Table 30. WrReg command description (Load Modulation Index and Demodulator Sensitivity) (continued)

Direction	Data	Comments	Example
ST95HF to Host	0x00	Result code	<<<0x0000 Register written
	0x00	Length of data (= RegCount)	

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

How to modify Analog Register Configuration register (ARC_B) values

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

   For example, to select the ISO/IEC 18092 protocol:

   Send ProtocolSelect command: >>>0x02020451
   ST95HF reply: <<<0x0000

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

   a) Write the ARC_B register index at 0x01: >>>0x0903680001
   ST95HF reply: <<<0x0000

   b) Read the ARC_B register value: >>>0x0803690100
   ST95HF reply: <<<0x015F

   In this example, the ARC_B register value is 0x5F, where “5” is the Modulation Index and “F” is the Receiver Gain.

3. Modify the Modulation Index and Receiver Gain values with 0x23.

   Write the ARC_B register index: >>>0x090468010123
   ST95HF reply: <<<0x0000

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

   a) Write the ARC_B register index at 0x01: >>>0x0903680001
   ST95HF reply: <<<0x0000

   b) Read the ARC_B register value: >>>0x0803690100
   ST95HF reply: <<<0x0123

Modulation Index and Receiver Gain values

Table 31. Possible Modulation Index values

Code	1	2	3	4	5	6	D
Modulation Index [1]	10%	17%	25%	30%	33%	36%	95%

1. Characterized only using ISO/IEC 10373 test set-up.

Table 32. Possible Receiver Gain values

Code	0	1	3	7	F
Receiver Gain [1]	34 dB	32 dB	27 dB	20 dB	8 dB

1. Characterized by design simulation.

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`
   ST95HF reply: `<<<0x0000`

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

   a) Write the ACC_A register index at 0x04: `>>>0x0903680004`
   ST95HF reply: `<<<0x0000`

   b) Read the ACC_A register value: `>>>0x0803690100`
   ST95HF 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`
   ST95HF reply: `<<<0x0000`

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

   a) Write the ACC_A register index at 0x04: `>>>0x0903680004`
   ST95HF reply: `<<<0x0000`

   b) Read the ACC_A register value: `>>>0x0803690100`
   ST95HF reply: `<<<0x0125`

Load Modulation Index and Demodulator Sensitivity values

Table 33. 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 34. 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’.

Default code per protocol

Table 35. ARC_B default code for available Reader protocols

Communication protocol	Default value	Recommended values for ST95HF demo board	Possible Modulation Index values (MS nibble)	Possible Receiver Gain values (LS nibble)
ISO/IEC 14443 Type A reader	0xDF	0xD7	0xD	0x0, 0x1, 0x3, 0x7 or 0xF
ISO/IEC 14443 Type B reader	0x2F	0x37	0x1, 0x2, 0x3 or 0x4	0x0, 0x1, 0x3, 0x7 or 0xF
ISO/IEC 18092 reader	0x5F	0x23	0x1, 0x2, 0x3 or 0x4	0x0, 0x1, 0x3, 0x7 or 0xF
ISO/IEC 15693 reader 30%	0x53	0x53	0x4, 0x5 or 0x6	0x0, 0x1, 0x3, 0x7 or 0xF
ISO/IEC 15693 reader 100%	0xD3	0xD3	0xD	0x0, 0x1, 0x3, 0x7 or 0xF

Table 36. ACC_A default code for available Card Emulation protocols

Communication protocol	Default value	Recommended values for ST95HF demo board	Possible Demodulator Sensitivity values (MS nibble)	Possible Load Modulation Index values (LS nibble)
ISO/IEC 14443 Type A	0x27	0x27	0x2	From 0x1 to 0xF

5.11.2 Improving frame reception for ISO/IEC 14443 Type A tags

To improve ST95HF demodulation when communicating with ISO/IEC 14443 Type A tags, it is possible to adjust the synchronization between digital and analog inputs by fine-tuning the Timer Window (TimerW) value. This can be done using the Write Register (WrReg) command to set a new TimerW value (min. 0x50, max. 0x60). The recommended value is 0x56 or 0x58 when using the ST95HF demo board.

The default value of this parameter (0x52) is set by the ProtocolSelect command, but it can be overwritten using the WrReg command (`0x09`).

Table 37. WRREG command description (Timer Window)

Direction	Data	Comments	Example
Host to ST95HF	0x09	Command code	>>>0x09043A005804 Set recommended TimerW value.
	0x03 or 0x04	Length of data	
	0x3A	Timer Window (TimerW) value	
	0x00 or 0x01	Flag Increment address or not after Write command	
	0xXX	Set TimerW value (recommended value is 0x56 or 0x58)	
	0x04	TimerW value confirmation	
ST95HF to Host	0x00	Result code	<<<0x0000 Register written
	0x00	Length of data (= RegCount)	

5.11.3 Improving RF reception for ISO/IEC 18092 tags

To improve ST95HF reception when communicating with ISO/IEC 18092 tags, it is possible to enable an AutoDetect filter to synchronize ISO/IEC 18092 tags with the ST95HF. This can be done using the Write Register (WRREG) command to enable the AutoDetect filter.

By default, this filter is disabled after the execution of the PROTOCOLSELECT command, but it can be enabled using the WRREG command (0x09).

Table 38. WRREG command description (AutoDetect Filter)

Direction	Data	Comments	Example
Host to ST95HF	0x09	Command code	>>>0x09040A0102A1 Enable the AutoDetect filter.
	0x03 or 0x04	Length of data	
	0x0A	AutoDetect filter control value	
	0x00 or 0x01	Flag Increment address or not after Write command	
	0x02	AutoDetect filter enable	
	0xA1	AutoDetect filter confirmation	
ST95HF to Host	0x00	Result code	<<<0x0000 Register written
	0x00	Length of data (= RegCount)	

5.12 Subcarrier frequency response (0x0B) description

The SubFreqRes command (0x0B) returns the most recent subcarrier frequency detected for ISO/IEC 18092 communication.

Table 39. SubFreqRes command description

Direction	Data	Comments	Example
Host to ST95HF	`0x0B`	Command code	`>>>0x0B00`
	`0x00`	Length of data	
ST95HF to Host	`0x00`	Result code	`<<<0x00010F` In this example, `0x0F`: is a frequency divider. Use this value to configure the ST95HF. Subcarrier frequency is $f_S = \frac{f_C}{2(N+1)}$
	`0x01`	Length of data	
	`<FreqDiv>`	*N*, frequency divider	

5.13 AcFilter command (0x0D) description

This command activates/deactivates the anti-collision filter in Type A Card Emulation mode.

Table 40. AC filter command description (Command <Len> > 0x02)

Direction	Data	Comments	Example

Host to ST95HF	`0D`	Command code	General case (LEN ≠ 0x00, 0x01 or 0x02)
			`>>>0D0B440320880251744AEF2280`
			Activate filter for 2-cascade anti-collision
			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
			All other values greater than 2 will result in an 'Invalid command length' error.
			*** 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*)

	`<Len>`	Length of data	
	`<ATQA>` (2 bytes)	Unused and proprietary bits of SAK (protocol bits will be handled by firmware)	
	`<SAK>`	Unused and proprietary bits of SAK (protocol bits will be handled by firmware)	
	`<UID part 1>`	UID for cascade level 1 (Mandatory)	
	`<UID part 2>`	UID for cascade level 2 (Optional)	
	`<UID part 3>`	UID for cascade level 3 (Optional)	

Table 40. AC filter command description (Command <Len> > 0x02) (continued)

Direction	Data	Comments	Example
ST95HF to Host	00	Result code	AC Filter is successfully activated. <<<0x0000 AC Filter is successfully deactivated (LEN = 0x00). <<<0x0001XX0000 (where XX is ST Reserved)
	00	Length of data	
ST95HF to Host	82	Error code	<<<0x8200 Invalid command length
	00	Length of data	

Table 41. ST95HF state when behaving as ISO/IEC 14443-A tag

Value	AC state
0x00	Idle
0x01	ReadyA
0x02	ReadyB
0x03	ReadyC
0x04	Active
0x80	Halt
0x81	ReadyAX
0x82	ReadyBX
0x83	ReadyCX
0x84	ActiveX

The AcFilter command activates or deactivates an anti-collision filter for ISO/IEC 14443A card protocol. This command also forces or indicates the Card Emulator state.

If the AC filter is not activated, all received commands are stored in the ST95HF buffer before being read by the external microcontroller.

If the AC filter is activated, internal firmware will try to interpret the commands sent by the reader itself and perform an anti-collision sequence. Non-interpreted commands are stored in the ST95HF buffer before being read by the external microcontroller to perform Level 4 of the ISO/IEC 14443 Type A protocol.

The ST95HF is able to interpret and respond to the commands listed in the table below. The device remains in Listening mode after executing these commands.

Table 42. Commands to which the device is able to respond

Command	Code	Definition
SENS_REQ	26 (7-bit)	Sense request
ALL_REQ	52 (7-bit)	WakeUp All request

Table 42. Commands to which the device is able to respond (continued)

Command	Code	Definition
SDD_REQ	93, 95, 97	Single device detection request
SEL_REQ	9370, 9570, 9770	Select request

5.14 Echo command (0x55) description

The ECHO command verifies the possibility of communication between a Host and the ST95HF. This command also can be used to exit Listening mode without an error when the ST95HF has received a Listen command.

Table 43. ECHO command description

Direction	Data	Comments	Example
Host to ST95HF	0x55	Command code	-
ST95HF to Host	0x55	Code response	>>> 0x55: Sends an ECHO command <<< 0x55: Response to an ECHO command

6 Electrical characteristics

6.1 Absolute maximum ratings

Table 44. Absolute maximum ratings

Symbol	Parameter	Value	Unit
VPS_Main	Supply voltage [1]	–0.3 to 7.0	V
VPS_TX	Supply voltage (RF drivers)	–0.3 to 7.0	V
V_{IO}	Input or output voltage relative to ground	–0.3 to VPS_Main +0.3	V
$V_{MaxCarrier}$	Maximum input voltage (pins RX1 and RX2)	±14.0	V
T_A	Ambient operating temperature	–25 to +85	°C
	Ambient operating temperature (RF mode)	–25 to +85	
T_{STG}	Storage temperature (Please also refer to package specification).	–65 to +150	°C
T_{LEAD}	Lead temperature during soldering	See note[2]	°C
V_{ESD}	Electrostatic discharge voltage according to JESD22-A114, Human Body Model	2000	V
P_{TOT} [3]	Total power dissipation per package	1	W

1. To properly reset the device, VPS_Main must be tied to 0V before executing the start-up sequence.
2. Compliant with JEDEC standard J-STD-020D (for small-body, Sn-Pb or Pb assembly), the ST ECOPACK® 7191395 specification, and the European directive on Restrictions on Hazardous Substances (RoHS directive 2011/65/EU of July 2011).
3. Depending on the thermal resistance of package.

Note: Stresses listed above may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of the specification is not implied.
Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

6.2 DC characteristics

Table 45. DC characteristics

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VPS_Main	Supply voltage	-	2.7	3.0	5.5	V
VPS_TX	Supply voltage (RF drivers)	-	2.7	3.0	5.5	V
V_{IL}	Input low voltage (I/Os)	-	0	-	0.2 x VPS_Main	V
V_{IH}	Input high voltage (I/Os)	-	0.7 x VPS_Main	-	VPS_Main	V
V_{OH}	Output high voltage (I/Os)	I_{OH} = - 8 µA	0.7 x VPS_Main	-	VPS_Main	V
V_{OL}	Output low voltage (I/Os)	I_{OLMAX} = 500 µA	0	-	0.15 x VPS_Main	V
POR	Power-on reset voltage	-	-	1.8	-	V

6.3 Power consumption characteristics

T_A = –25 °C to 85 °C, unless otherwise specified.

Table 46. Power consumption characteristics (VPS_Main from 2.7 to 3.3 V)

Symbol	Parameter	Condition	Typ.	Max.	Unit
I_{CC} (V_{PS}) Power-up	Supply current in power-up state	T_A = 25°C	200	600	µA
I_{CC} (V_{PS}) Card Emulation	Supply current in Card Emulation mode	T_A = 25 °C	2.5	5.0	mA
I_{CC} (V_{PS}) Hibernate	Supply current in Hibernate state	T_A = 25 °C	1	5	µA
I_{CC} (V_{PS}) Sleep/Field Detector	Supply current in Sleep/Field Detector state	T_A = 25 °C	20/25	80	µA
I_{CC} (V_{PS}) Ready	Supply current in Ready state	T_A = 25 °C	2.5	5.0	mA
I_{CC} (V_{PS}) Tag Detect	Average supply current in Tag Detector state	T_A = 25 °C, 4 RF bursts per second	50	100	µA

The ST95HF supports two VPS_TX supply ranges for RF drivers: 2.7V to 3.3V or 4.5V to 5.5V. Antenna matching circuit must be defined accordingly.

Table 47. Power consumption characteristics (VPS_TX from 2.7 to 3.3 V)

Symbol	Parameter	Condition	Typ.	Max.	Unit
I_{CC} RF (V_{PS_TX}) RF Field ON	Supply current in RF Field (Reader mode) [1]	T_A = 25 °C	70	100	mA
I_{CC} RF (V_{PS_TX}) RF Field OFF	Supply current in RF Field (Ready mode) [2]	T_A = 25 °C	-	200	µA
I_{CC} RF (V_{PS_TX}) Tag Detect	Peak[3] current during Burst detection	T_A = 25 °C	70	100	mA

1. Parameter measured using recommended output matching network (Z load is 27 Ω and 0°).

2. This consumption can be reduced to approximately 2 µA (typ.) by setting a control bit (bit HF2RF) to ‘1’ using command `090468010710`. In this case, Reader mode is not available.
   To re-enable Reader mode, reset the HF2RF bit to ‘0’ using the command `090468010700` or execute a new PROTOCOLSELECT command.

3. The maximum differential input voltage between pins RX1 and RX2 (VRx1-Rx2) has a peak-peak of 18 V.

Table 48. Power consumption characteristics (VPS_TX from 4.5 to 5.5 V)

Symbol	Parameter	Condition	Typ.	Max.	Unit
I_{CC} RF (V_{PS_TX}) RF Field ON	Supply current in RF Field (Reader mode) [1]	T_A = 25 °C	120	200	mA
I_{CC} RF (V_{PS_TX}) RF Field OFF	Supply current in RF Field (Ready mode) [2]	T_A = 25 °C	-	300	µA
I_{CC} RF (V_{PS_TX}) Tag Detect	Peak[3] current during Burst detection	T_A = 25 °C	120	200	mA

1. Parameter measured using recommended output matching network. (Z load is 16 Ω and 0°).

2. This consumption can be reduced to approximately 2 µA (typ.) by setting a control bit (bit HF2RF) to ‘1’ using command `090468010710`. In this case, Reader mode is not available.
   To re-enable Reader mode, reset the HF2RF bit to ‘0’ using the command `090468010700` or execute a new PROTOCOLSELECT command.

3. The maximum differential input voltage between pins RX1 and RX2 (VRx1-Rx2) has a peak-peak of 18 V. This voltage can be limited by adding a damping resistor in parallel of the antenna or between ST_R0 and Ground.

6.4 SPI characteristics

The ST95HF supports (CPOL = 0, CPHA = 0) and (CPOL = 1, CPHA = 1) modes.

Table 49. SPI interface characteristics

Symbol	Parameter	Condition	Min.	Max.	Unit
f_{SCK} $1/\ t_{c(SCK)}$	SPI clock frequency	-	-	2.0	MHz
V_{IL}	Input low voltage	-	-	0.3 x V_{PS}	V
V_{IH}	Input high voltage	-	0.7 x V_{PS}	-	
V_{OL}	Output low voltage	-		0.4 x V_{PS}	
V_{OH}	Output high voltage	-	0.7 x V_{PS}	-	
$t_{SU(NSS)}$[1]	NSS setup time	-	70	-	ns
$t_{h(NSS)}$[1]	NSS hold time	-	0	-	
$t_{CH(SCKL)}$[1]	Clock low time	-	200	-	ns
$t_{CH(SCKH)}$[1]	Clock high time	-	200	-	
$t_{SU(SI)}$[1]	Data slave Input setup time	-	20	-	ns
$t_{h(SI)}$[1]	Data slave Input hold time	-	80	-	
$t_{v(SO)}$[1]	Data slave output valid time	-		80	ns
$t_{h(SO)}$[1]	Data slave output hold time	After enable edge	0	-	
$C_{b\_SPI\_IN}$	Capacitive load for input pins NSS, CLK, MOSI	-	-	3	pF
$C_{b\_SPI\_OUT}$	Capacitive load for input pins MOSI	-	-	20	pF

1. Values based on design simulation and/or characterization results, and not on tested in production.

Figure 12. SPI timing diagram (Slave mode and CPOL = 0, CPHA = 0)

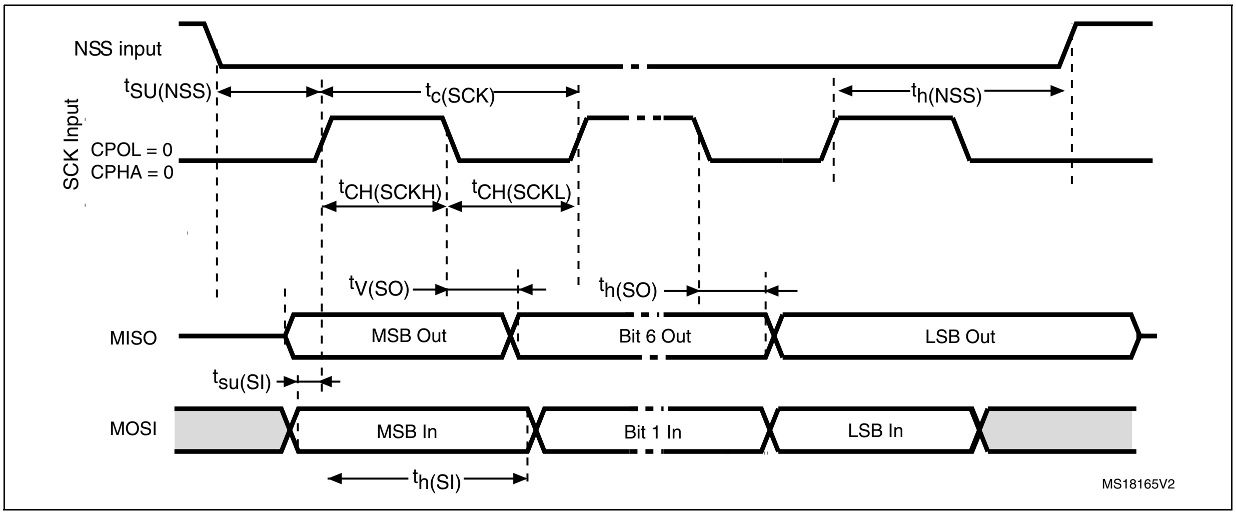
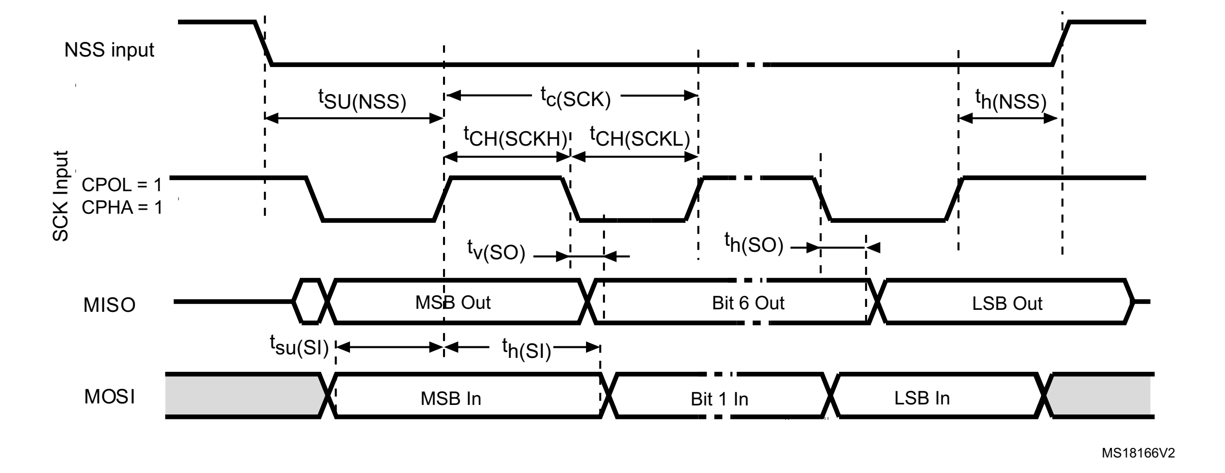


Figure 13. SPI timing diagram (Slave mode and CPOL = 1, CPHA = 1)



6.5 RF characteristics

Test conditions are T_A = 0°C to 50°C, unless otherwise specified.
VPS_TX = 3V ± 10% and VPS_MAIN = 3V ± 10%.

Table 50. Tag/Card Emulation characteristics

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
f_C	Frequency of operating field (carrier frequency)	-	13.553	13.56	13.567	MHz
MI Carrier	Carrier Modulation Index from reader ISO/IEC 14443 -Type A	-	-	-	100	%
HField [(1)]	Operating field strength in ISO/IEC 14443	-	1.5	-	7.5	A/m
$V_{MaxCarrier}$	Input voltage between RX1 and RX2	-	-	-	18	V
f_S [(2)]	Frequency of Subcarrier modulation (ISO/IEC 14443, ISO/IEC 18092 and ISO/IEC 15693)	-	-	-	f_C / 16	
ALoad	Load Modulation Amplitude $f_C + f_S$ $f_C - f_S$	ISO 10373-6 test methods for ISO/IEC 14443-A @ 1.5 A/m	18 18	-	-	mV
	$f_C + f_S$ $f_C - f_S$	ISO 10373-6 test methods for ISO/IEC 14443-A @ 7.5A/m	8 8	-	-	mV
ALoad	Load Modulation Amplitude $f_C + f_S$ $f_C - f_S$	ECMA 356 test methods for ECMA 340 @ 1.5 A/m	18 18	-	-	mV
	$f_C + f_S$ $f_C - f_S$	ECMA 356 test methods for ECMA 340 @ 7.5 A/m	8 8	-	-	mV
DataR	ISO/IEC 14443 Type A		106	-	212	Kbps

1. Maximum values based on design simulation and/or characterization results, and not tested in production.
2. Parameter measured on samples using recommended output matching network.

Table 51. Field detection characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit
-	Level of detection [(1)]	0.5	-	8	A/m

1. Parameter measured using recommended output matching network for ISO/IEC 14443 communication.

Table 52. Reader characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit
f_C	Frequency of operating field (carrier frequency)	13.553	13.56	13. 567	MHz
MI Carrier	Carrier modulation index[1] ISO/IEC 14443-A ISO/IEC 14443-B ISO/IEC 18092 ISO/IEC 15693 (10% modulation)[2] ISO/IEC 15693 (100% modulation)	- 8 8 10 80	-	100 14 14 30 100	%
Transmitter specifications (VPS_TX = 2.7 to 3.3 V)					
-	Z_{OUT} differential impedance between TX1 and TX2[1]	-	27	-	Ω
-	Output power for 3V operation on pin VPS_TX [1][2]	-	55	-	mW
Transmitter specifications (VPS_TX = 4.5 to 5.5 V)					
-	Z_{OUT} differential impedance between TX1 and TX2[1]	-	16	-	Ω
-	Output power for 5V operation on pin VPS_TX [1] [2]	-	230	-	mW
Receiver specifications					
-	Small signal differential input resistance (Rx1/Rx2)[1]	-	100	-	kΩ
VRx1-Rx2	Differential input voltage between pins RX1 and RX2[3]	-	-	18	V
-	Small signal differential input capacitance (Cx1/Cx2)[1]	-	22	-	pF
-	Sensitivity (106 Kbps data rate)[4]	-	8	-	mV

1. Maximum values based on design simulation and/or characterization results, and not tested in production.
2. Parameter measured on samples using recommended output matching network. (Z load is 27 Ω and 0°.)
3. This voltage can be limited by adding a damping resistor in parallel of the antenna or between ST_R0 and Ground.
4. Based on ISO/IEC 10373-6 protocol measurement. The reader sensitivity corresponds to the load modulation value of the REQ reply sent by an ISO reference card when decoded by the ST95HF.

6.6 Oscillator characteristics

The external crystal used for this product is a 27.12 MHz crystal with an accuracy of ± 14 kHz.

Table 53. HFO 27.12 MHz oscillator characteristics[1] [2]

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
f_{XTAL}	Oscillator frequency	-	-	27.12	-	MHz
R_F	Feedback resistor	-	-	2	-	MΩ
C	Recommended load capacitance versus equivalent serial resistance of the crystal (R_S)[3]	R_S = 30 Ω	-	6	-	pF
$t_{SU(HFO)}$[4]	Startup time	V_{PS} is stabilized	-	6	10	ms

[1] Resonator characteristics given by the crystal/ceramic resonator manufacturer.

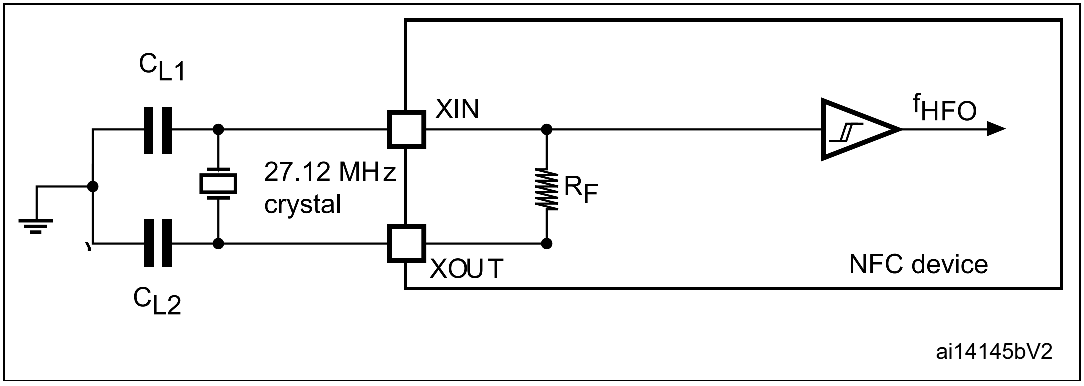
[2] Based on characterization, not tested in production.

[3] The relatively low value of the R_F resistor offers a good protection against issues resulting from use in a humid environment, due to the induced leakage and the bias condition change. However, it is recommended to take this point into account if the Host is used in tough humidity conditions.

[4] $t_{SU(HFO)}$ is the startup time measured from the moment it is enabled (by software) to a stabilized 27.12 MHz oscillation is reached. This value is measured for a standard crystal resonator and it can vary significantly with the crystal manufacturer.

For C_{L1} and C_{L2}, it is recommended to use high-quality external ceramic capacitors in the 10 pF to 20 pF range (typ.), designed for high-frequency applications, and selected to match the requirements of the crystal or resonator (see *Figure 14*). C_{L1} and C_{L2} are usually the same size. The crystal manufacturer typically specifies a load capacitance which is the series combination of C_{L1} and C_{L2}.

Figure 14. Typical application with a 27.12 MHz crystal



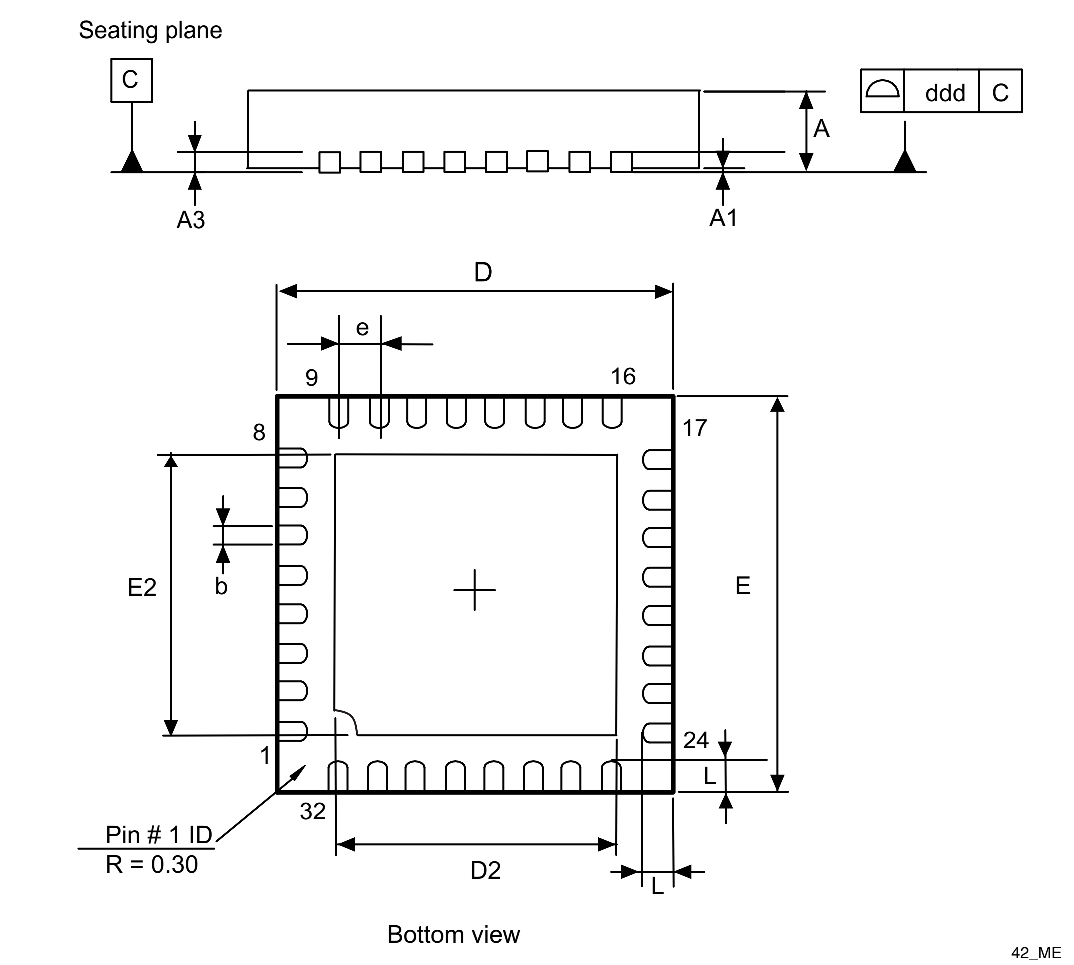
Note: For C_{L1} and C_{L2} it is recommended to use high-quality ceramic capacitors in the 10 to 20 pF range, selected to match the requirements of the crystal or resonator. C_{L1} and $C_{L2,}$ have usually the same size. The crystal manufacturer typically specifies a load capacitance which is the series combination of C_{L1} and C_{L2}.
Load capacitance C_L has the following formula: $C_L = C_{L1} \times C_{L2} / (C_{L1} + C_{L2}) + C_{stray}$, where C_{stray} (typically between 2 and 7 pF) is the pin capacitance and board or trace PCB-related capacitance.

7 Package information

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at *www.st.com*. ECOPACK® is an ST trademark.

7.1 VFQFPN32 package information

Figure 15. VFQFPN32 - 32-pin, 5x5 mm, 0.5 mm pitch very thin profile fine pitch quad flat package outline



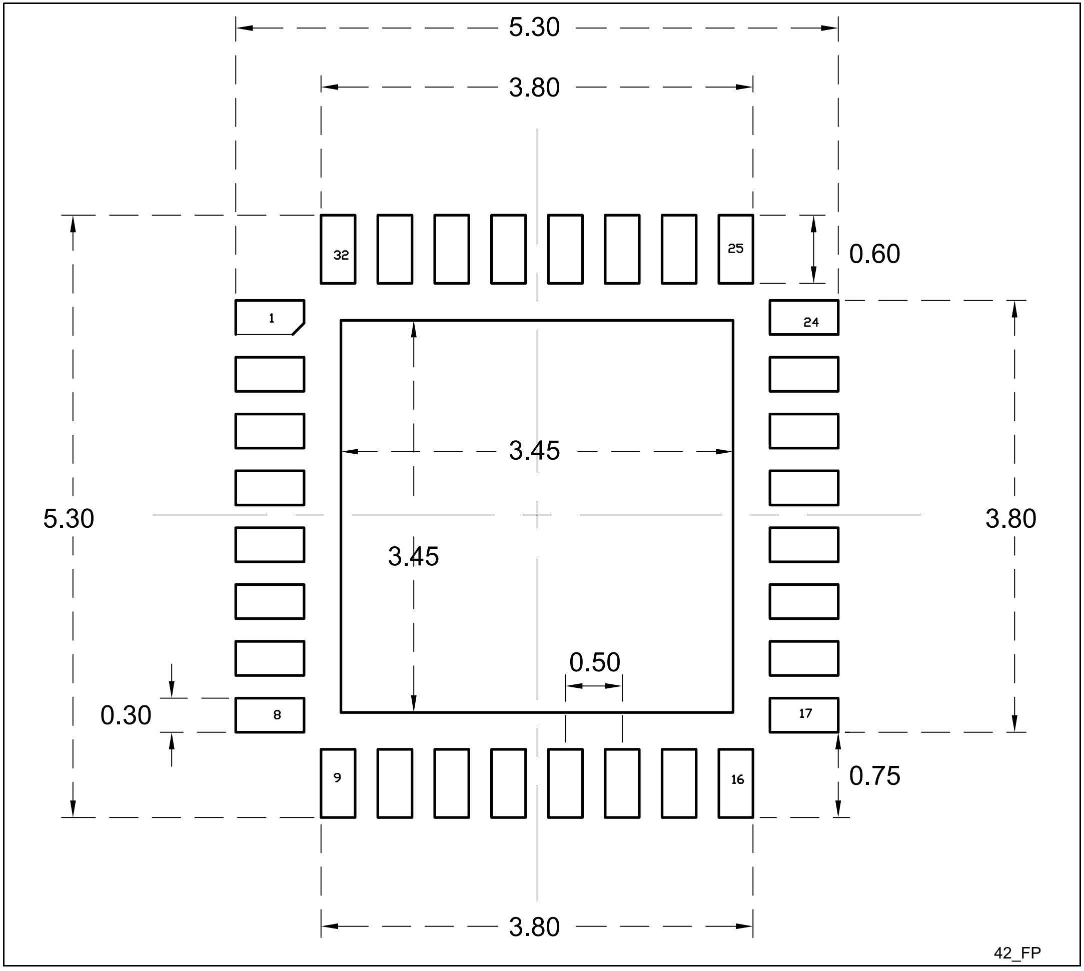
1. Drawing is not to scale.

Table 54. VFQFPN32 - 32-pin, 5x5 mm, 0.5 mm pitch very thin profile fine pitch quad flat package mechanical data

Symbol	millimeters			inches[1]		
	Min	Typ	Max	Min	Typ	Max
A	0.800	0.900	1.000	0.0315	0.0354	0.0394
A1	0.000	0.020	0.050	0.0000	0.0008	0.0020
A3	-	0.200	-	-	0.0079	-
b	0.180	0.250	0.300	0.0071	0.0098	0.0118
D	4.850	5.000	5.150	0.1909	0.1969	0.2028
D2	3.400	3.450	3.500	0.1339	0.1358	0.1378
E	4.850	5.000	5.150	0.1909	0.1969	0.2028
E2	3.400	3.450	3.500	0.1339	0.1358	0.1378
e	0.450	0.500	0.550	0.0177	0.0197	0.0217
L	0.300	0.400	0.500	0.0118	0.0157	0.0197
ddd	-	-	0.080	-	-	0.0031

1. Values in inches are converted from mm and rounded to 4 decimal digits.

Figure 16. VFQFPN32 - 32-pin, 5x5 mm, 0.5 mm pitch very thin profile fine pitch quad flat package recommended footprint



1. Dimensions are expressed in millimeters.

8 Part numbering

Table 55. Ordering information scheme

Example:	ST	95	HF	– V	MD	5	T

Device type

ST = NFC transceiver

Wired access

95 = SPI

Frequency band

HF = High frequency (13.56 MHz)

Operating voltage

V = 2.7 to 5.5 V

Package

MD = 32-pin VFQFPN (5 x 5 mm)

Operating temperature

5 = –25° to +85° C

Packaging

T = Tape and Reel

Not all combinations are necessarily available. For a list of available options (speed, package, etc.) or for further information on any aspect of this device, please contact your nearest STMicroelectronics Sales Office.

Appendix A Additional Idle command description

This section provides examples of use for the IDLE command.

The wake-up source is the third of the 16 bytes in the IDLE command. This byte specifies authorized Wake-up events. This revision now also provides the capability to set the LFO frequency in WFE mode.

The LFO frequency and the authorized wake-up source settings are stored in the Wake-up source register as the parameters of the IDLE command.

The Wake-up event is updated by the ST95HF when it exits WFE mode.

The contents of the Wake-up event register can be read using the Read Register command or in the ST95HF reply to the Idle command.

Table 56. Wake-up source register

Bits [7:6]	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LFO frequency	RFU[1]	IRQ on pin $\overline{\text{SPI\_SS}}$	IRQ on pin $\overline{\text{IRQ\_IN}}$	Field Detect	Tag Detect	Timeout

[1] Must be set to ‘0’.

Table 57. Wake-up event register

Bits [7:6]	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LFO frequency	RFU	IRQ on pin $\overline{\text{SPI\_SS}}$	IRQ on pin $\overline{\text{IRQ\_IN}}$	Field Detect	Tag Detect	Timeout

Bits [7:6] define the LFO frequency (f_{LFO}):

00: 32 kHz 01: 16 kHz
10: 8 kHz 11: 4 kHz

Bit 4: When set, the ST95HF will wake up when an external interrupt (low level on pin $\overline{\text{SPI\_SS}}$) is detected.

Bit 3: When set, the ST95HF will wake up when an external interrupt (low level on pin $\overline{\text{IRQ\_IN}}$) is detected. This is useful for SPI communication. It is recommended to set this bit to ‘1’ in order to recover in the event of a system crash.

Bit 2: When set, the ST95HF will wake up when an RF field is detected.

Bit 1: When set, the ST95HF will wake up when a tag is detected in the RF field. This bit must also be set during Tag Detection calibration or during a Tag Detection sequence.

Bit 0: When set, the ST95HF will wake up and return to Ready state at the end of a predefined cycle. The Timeout (TO) value is defined by the MaxSleep and Wake-up period:

$TO = (MaxSleep *(WuPeriod+1)*t_{REF}$

$t_{REF} = 256*t_L = 8$ ms ($f_{LFO} = 32$ kHz), mean power consumption in Sleep mode is 25 µA

$t_{REF} = 256*t_L = 64$ ms ($f_{LFO} = 4$ kHz), mean power consumption in Sleep mode is 20 µA

Note: Note that: 0x00 < MaxSleep < 0x1F.

This bit must be set when using the timer as a possible wake-up source. It must be set during Tag Detection Calibration to force a wake-up after the first Tag Detection trial.

Appendix B Example of tag detection calibration process

The following script works on the DEMO_CR95HF evaluation board and with the ST95HF development software available on *www.st.com*.

This is a dichotomous approach to quickly converge to the DacDataRef value for which a wake-up event switches from tag detection to Timeout. In this process, only the DacDataH parameter is changed in successive Idle commands. And we look at the wake-up event reply to decide the next step.

```
00 01 02 corresponds to a Tag Detect,
00 01 01 corresponds to a Timeout.
```

```
REM, Tag Detection Calibration Test

REM,   Sequence: Power-up Tag Detect Wake-up by Tag Detect (1 try
measurement greater or equal to DacDataH) or Timeout

REM,   CMD 07 0E 03 A100 D801 1800 01 60 60 00 XX 3F 00

REM,   03   WU source = Tagdet or Timeout

REM,   A100  Initial Dac Compare

REM,   F801  Initial Dac Compare

REM,   1800  HFO

REM,   20 Wup Period 32 Inactivity period = 256ms (LFO @ 32kHz)

REM,   60 Osc  3ms   (LFO @ 32kHz)

REM,   60 Dac  3ms   (LFO @ 32kHz)

REM,   00  DacDataL  = minimum level (floor)

REM,   xx  DacDataH 00 = minimum level (ceiling)

REM,   3F  Swing 13.56  4.6 us

REM,   01 Maximum number of Sleep before Wakeup 2


REM, Tag Detection Calibration Test

REM, During tag detection calibration process DacDataL = 0x00

REM, We execute several tag detection commands with different
DacDataH values to determine DacDataRef level corresponding to
ST95HF application set-up

REM, DacDataReg value corresponds to DacDataH value for which Wake-
up event switches from Timeout (0x01) to Tag Detect (0x02)

REM, Wake-up event  = Timeout  when  DacDataRef is between DacDataL
and DacDataH

REM, Search DacDataref value corresponding to value of DacDataH for
which Wake-up event switches from Tag Detect (02) to Timeout(01)
```

```
REM, Step 0: force wake-up event to Tag Detect (set DacDataH = 0x00)

REM, With these conditions Wake-Up event must be Tag Detect

>>> CR95HFDLL_STCMD, 01070E03A100F801180020606000003F01

<<< 000102

REM, Read Wake-up event  = Tag Detect  (0x02); if not, error .


REM, Step 1: force Wake-up event to Timeout (set DacDataH = 0xFC

REM, With these conditions, Wake-Up event must be Timeout

>>> CR95HFDLL_STCMD, 01070E03A100F801180020606000FC3F01

<<< 000101

REM, Read Wake-up event = Timeout (0x01); if not, error .


REM, Step 2: new DacDataH value = previous DacDataH +/-  0x80

REM, If previous Wake-up event was Timeout (0x01) we must decrease
DacDataH (-0x80)

>>> CR95HFDLL_STCMD, 01070E03A100F8011800206060007C3F01

<<< 000101

REM, Read Wake-up event = Timeout (0x01) or Wake-up event = Tag
Detect (0x02)


REM, Step 3: new DacDataH value = previous DacDataH +/- 0x40

REM, If previous Wake-up event was Timeout (0x01), we must decrease
DacDataH (-0x40); else, we increase DacDataH (+ 0x40)

>>> CR95HFDLL_STCMD, 01070E03A100F8011800206060003C3F01

<<< 000102

REM,  Read Wake-up event = Timeout (0x01)  or  Wake-up event  = Tag
Detect (0x02)


REM, Step 4: new DacDataH value = previous DacDataH +/- 0x20

REM, If previous Wake-up event was Timeout (0x01), we must decrease
DacDataH (-0x20); else, we increase DacDataH (+ 0x20)

>>> CR95HFDLL_STCMD, 01070E03A100F8011800206060005C3F01

<<< 000102

REM, Read Wake-up event = Timeout (0x01)  or  Wake-up event = Tag
Detect (0x02)
```

```
REM, Step 5: new DacDataH value = previous DacDataH +/- 0x10

REM, If previous Wake-up event was Timeout (0x01), we must decrease
DacdataH (-0x10); else, we increase DacDataH (+ 0x10)

>>> CR95HFDLL_STCMD, 01070E03A100F8011800206060006C3F01

<<< 000102

REM, Read Wake-up event = Timeout (0x01) or Wake-up event = Tag
Detect (0x02)


REM, Step 6: new DacDataH value = previous DacDataH +/- 0x08

REM, If previous Wake-up event was Timeout (0x01), we must decrease
DacDataH (-0x08); else, we increase DacDataH (+ 0x08)

>>> CR95HFDLL_STCMD, 01070E03A100F80118002060600743F01

<<< 000101

REM, Read Wake-up event = Timeout (0x01) or  Wake-up event = Tag
Detect (0x02)


REM, Step 7: new DacDataH value = previous DacDataH +/- 0x04

REM, If previous Wake-up event was Timeout (0x01), we must decrease
DacDataH (-0x04); else, we increase DacDataH (+ 0x04)

>>> CR95HFDLL_STCMD, 01070E03A100F80118002060600703F01

<<< 000101

REM, Read Wake-up event = Timeout (0x01) or  Wake-up event = Tag
Detect (0x02)


REM, If  last Wake-up event = Tag Detect (0x02), search DacDataRef =
last DacDataH value

REM, If  last Wake-up event = Timeout (0x01), search DacDataRef =
last DacDataH value -4


REM, For tag detection usage, we recommend setting DacDataL =
DacDataRef -8 and DacDataH = DacDataRef +8


>>> CR95HFDLL_STCMD, 01070E0B21007801180020606064743F01

<<< 000101
```

Appendix C Example of tag detection command using results of tag detection calibration

The following script works on the DEMO_CR95HF evaluation board and with the ST95HF development software available on *www.st.com*.

This is an example of a Tag Detection command when a tag is not present in the RF operating volume using the ST95HF:

```
>>> CR95HFDLL_STCMD, 01 070E0B21007801180020606064743F01

<<< 000101 Wake-up event = Timeout (0x01)

>>> CR95HFDLL_STCMD, 01 0803620100

<<< 000101
```

This is an example of a Tag Detection command when a tag is present in the RF operating volume using the ST95HF:

```
>>> CR95HFDLL_STCMD, 01 070E0B21007801180020606064743F01

<<< 000102 Wake-up event = Tag Detect (0x02)

>>> CR95HFDLL_STCMD, 01 0803620100

<<< 000102
```

Appendix D Examples of ST95HF command code to activate NFC Forum and ISO/IEC 15693 tags

The following script works on the DEMO_CR95HF evaluation board and with the ST95HF development software available on www.st.com.

This section provides examples of ST95HF command code used to activate NFC Forum and ISO/IEC 15693 tags using ST95HF development software.

`CR95HFDLL_STCMD`: Is the standard ST95HF frame exchange command. In this command, the first byte `01` is not sent, it is only requested by the ST95HF development software in order to recognize if it is a user or service command.

`CR95HFDLL_SENDRECV`: Is the encapsulated ST95HF SendReceive command for which command codes, number of bytes, and CRC are automatically appended to the parameter.

In this section,

- The ST95HF command overhead (command code, length of data and transmission flag) is in black.
- The Tag instruction is in blue.
- The ST95HF response overhead (result code, length of data and status) is in green.
- The Tag response is in red.

When the CRC append option is set in the Protocol Select command, the CRC is automatically appended by the ST95HF, but the CRC is not visible in the instruction log file.

When the CRC is present in the command or response, CRC reply is in *italics*.

The following symbols correspond to:

>>> Frame sent by Host to ST95HF

<<< Frame received by Host from ST95HF

D.1 ISO/IEC 14443 Type A

D.1.1 NFC Forum Tag Type 1 (Topaz)

```
REM,  ST95HF code example to support NFC Forum Tag Type 1 14443_A

REM,  TEST TOPAZ 14443A (UID 6E567A00)

REM,  first byte 01 in CR95HFDLL_STCMD is only requested by ST95HF
Development SW

REM,  RFOFF

>>> CR95HFDLL_STCMD, 01 02020000

<<< 0000

REM,  TEST  TOPAZ   14443A (UID 6E567A00)

REM,  Sel Prot 14443A  option TOPAZ

>>> CR95HFDLL_STCMD, 01 020402000300

<<< 0000
```

```
REM, Optimization of synchronization between digital and analog
inputs by adjusting TimerW value (default 0x52, min. 0x50, max.
0x60). Recommended value is 0x56 or 0x58 for NFC Forum Tag Type 1
(Topaz).

>>> CR95HFDLL_STCMD, 01 09043A005804

<<< 0000

REM,  Recommended modulation and gain is 0xD1 or 0xD3 for NFC Forum
Tag Type 1 (Topaz).

>>> CR95HFDLL_STCMD, 01 0904680101D1

<<< 0000

REM, last Byte x7 or x8 in CR95HFDLL_SENDRECV  command  number of
bits in the 14443 _Type A  frame

REM,  REQA reply ATQA 000C

>>> CR95HFDLL_STCMD, 01 04 02 26 07

<<< 80 05 000C 280000

REM,  RID reply HR0 HR1 UID0 UID 1 UID2 UID3

>>> CR95HFDLL_STCMD, 01 04 08 78000000000000 A8

<<< 80 0B 11 48 6E567A00 3E45 080000

REM,  RAll 0408 0000 UID0 UID 1 UID2 UID3 Reply HR0 HR1 UID0 UID 1
UID2 UID3 datas

>>> CR95HFDLL_STCMD, 01 04 08 000000 6E567A00 A8

<<< 80 40 11 48 6E567A00
0002250000100E000313D1010F5402656E5573652043523935524620210000000000
0000000000000000000000000000000000000000000CCCCCC

REM,  Read ad08 00 UID0 UID 1 UID2 UID3

>>> CR95HFDLL_STCMD, 01 04 08 01 0800 6E567A00 A8

<<< 80 07 08 00 87C1 080000

REM,  Write_E ad08   data 12 UID0 UID 1 UID2 UID3

>>> CR95HFDLL_STCMD, 01 04 08 53 0812 6E567A00 A8

<<< 80 07 08 12 14F2 080000

REM,  Read ad08 00 UID0 UID 1 UID2 UID3

>>> CR95HFDLL_STCMD, 01 04 08 01 0800 6E567A00 A8

<<< 80 07 08 12 14F2 080000

REM,  Write_NE ad08   data A5 UID0 UID 1 UID2 UID3

>>> CR95HFDLL_STCMD, 01 04 08 1A 08A5 6E567A00 A8

<<< 80 07 08 B7 B300 080000

REM,  Read ad08 00 UID0 UID 1 UID2 UID3

>>> CR95HFDLL_STCMD, 01 04 08 01 0800 6E567A00 A8
```

```
<<< 80 07 08 B7 B300 080000
REM,  Write_E ad08   data 00 UID0 UID 1 UID2 UID3
>>> CR95HFDLL_STCMD, 01 04 08 53 0800 6E567A00 A8
<<< 80 07 08 00 87C1 080000
REM,  Read ad08 00 UID0 UID 1 UID2 UID3
>>> CR95HFDLL_STCMD, 01 04 08 01 0800 6E567A00 A8
<<< 80 07 08 00 87C1 080000
```

D.1.2 NFC Forum Tag Type 2

```
REM, ST95HF code example to support NFC Forum Tag Type 2 14443_A
REM,  TEST INVENTORY then Read & Write in Memory


REM, Protocol select 14443A
>>> CR95HFDLL_STCMD, 01 02020200
<<< 0000
REM, Optimization of synchronization between digital and analog
inputs by adjusting TimerW value (default 0x52, min. 0x50, max.
0x60). Recommended value is 0x56 or 0x58 for NFC Forum Tag Type 2.
>>> CR95HFDLL_STCMD, 01 09043A005804
<<< 0000
REM,  Recommended modulation and gain is 0xD1 or 0xD3 for NFC Forum
Tag Type 2.
>>> CR95HFDLL_STCMD, 01 0904680101D1
<<< 0000
>>> CR95HFDLL_ANTICOLSELECT123


------ ISO14443-A STARTING ANTICOLLISION ALGORITHM ------
ISO14443-A REQAreply ATQA
>>> CR95HFDLL_SENDRECV, 26 07
<<< 80 05 4400 280000
ISO14443-A ANTICOL 1
>>> CR95HFDLL_SENDRECV, 93 20 08
<<< 80 08 8804179F04 280000
ISO14443-A SELECT 1
>>> CR95HFDLL_SENDRECV, 93 70 8804179F04 28
<<< 80 06 04 DA17 080000
```

```
ISO14443-A ANTICOL 2
>>> CR95HFDLL_SENDRECV, 9520 08
<<< 80 08 7910000069 280000
ISO14443-A SELECT 2
>>> CR95HFDLL_SENDRECV, 9570 7910000069 28
<<< 80 06 00 FE51 080000

--> UID = 04179F10000069
--> TAG selected
------ ISO14443-A END OF ANTICOLLISION ALGORITHM ------

REM,  READ @A5
>>> CR95HFDLL_SENDRECV, 300C 28
<<< 80 15  00000000FFFFFFFFFFFFFFFFFFFFFFFF F4CD  080000
REM,  WRITE @0C data A5
>>> CR95HFDLL_SENDRECV, A20CA5A5A5A5 28
<<< 87 00 : Frame wait time out OR no tag
REM,  READ @A5
>>> CR95HFDLL_SENDRECV, 300C 28
<<< 80 15  A5A5A5A5FFFFFFFFFFFFFFFFFFFFFFFF 84D8  080000
```

D.1.3 NFC Forum Tag Type 2 or 4: using split frames to resolve collisions

```
REM,  TEST  ANTICOLISION 2 tags 14443_A
REM,  CR95HF CONFIGURATION : ISO14443-A protocol
>>> CR95HFDLL_SELECT, 02000280
<<< 0000
REM,  ISO14443-A : CONFIG
>>> CR95HFDLL_STCMD, 01 09043A005A04
<<< 0000
REM, ISO14443-A : INCREASE DEMOD GAIN
>>> CR95HFDLL_STCMD, 01 0904680101DF
<<< 0000

REM, Anticollision  2 tags
REM, Tag 1 --> UID = 044B744AEF2280
```

```
REM, Tag 2 --> UID = 043B114AEF2280

REM, Response When 2 tags are present NVB = nb Byte OK + nb bit OK

REM, Collision  B8

REM, First Byte Coll 02 (3 eme Byte) (8804 ok)

REM, Index bit Coll 04 (5eme bit) => SN finish by 0 or 1

REM,   set NVB = 45


REM,  REQA  Poll  field with Two tags In operating volume

>>> CR95HFDLL_STCMD, 01 04022607

<<< 80 05 4403 280000


REM,   Ant CL1  Collision Detection (NVB 20) none data on UID

REM, Response Flag F1=B8  F2=02 F3=04

REM,  F1=B8 collision detected 8 bits significatifs in first byte of response

REM,  F2=02  collision detected in 3rd response byte, index start at 0

REM,  F3=04 collision detected on 5th bit, index start at 0

>>> CR95HFDLL_STCMD, 01 0403932008

<<< 80 08 88047B75B7 B80204


REM,  Collision Management Usage of split frame with NVB = 45 4bytes OK  + 5 lsb bit OK in next byte

REM,  last bit (collision one)  arbitrary fixed to select only one branch of UID tree

REM,  padding of last byte with zeros as msb (tranmission lsb first) 0B

REM,  command flag 45 usage of Split Frame (4) with 5 bits significatif in last byte

REM,  uncomplete response byte is padded with 0 as lsb bits

REM,  Response Flag F1=23  F2=00 F3=00

REM,  F1=23, No collision

REM,   3 bit significatif (msb last received bits) in first response byte 40

REM,  F2=00, not significant

REM,  F3=00, not significant

>>> CR95HFDLL_STCMD, 01 0406934588040B45
```

```
<<< 80 06 4074B3 230000
REM,  Activation tag Highest branch
>>> CR95HFDLL_SENDRECV, 937088044B74B328
<<< 80 06 24D836 080000
REM,   Ant CL2
>>> CR95HFDLL_STCMD, 01 0403952008
<<< 80 08 4AEF228007 280000
REM,  Sel CL2 tag Highest branch
>>> CR95HFDLL_SENDRECV, 95704AEF22800728
<<< 80 06 20FC70 080000

REM,  ISO14443-A HLTA tag1 Highest branch
>>> CR95HFDLL_SENDRECV, 500028
<<< 8700 : Frame wait time out OR no tag

REM,  WUPA  Second tags In field
>>> CR95HFDLL_STCMD, 01 04025207
<<< 80 05 4403 280000
REM,   Ant CL1  Detection Collision second tag (No collision)
>>> CR95HFDLL_STCMD, 01 0403932008
<<< 80 08 88047B75B7 B80204
REM,  Activation tag lowest branch   043B114AEF2280
>>> CR95HFDLL_SENDRECV, 937088043B11A6789808
<<< 80 06 24D836 080000
REM,   Ant CL2
>>> CR95HFDLL_STCMD, 01 0403952008
<<< 80 08 4AEF228007 280000
REM,  Sel CL2 tag Highest branch
>>> CR95HFDLL_SENDRECV, 95704AEF22800728
<<< 80 06 20FC70 080000

REM,  ISO14443-A HLTA tag2 Lowest branch
>>> CR95HFDLL_SENDRECV, 500028
<<< 8700 : Frame wait time out OR no tag
```

```
REM,  REQA no other tag In operating volume
>>> CR95HFDLL_STCMD, 01 04022607
<<< 8700
```

D.1.4 NFC Forum Tag Type 2

Communication using Parity Framing mode which is compliant with MIFARE® framing requirements.

```
REM, TEST Extract NDEF Message of NFC Tag Type 2 using Parity
Framing mode option


REM, ISO14443-A protocol select
>>> 02000280
<<< 0000
REM, ISO14443-A configuration
>>> 09043A005A04
<<< 0000
REM,  ISO14443-A: Increase modulation and gain
>>> 0904680101DF
<<< 0000


REM, REQA
>>> 04 02 26 07
<<< 80 05 4400 280000


REM,   Ant CL1
>>> 04 03 93 20 08
<<< 80 08 8804CB8CCB 280000
REM,  Sel CL1
>>> 04 08 9370 8804CB8CCB 28
<<< 80 06 04 DA17 080000
REM,   Ant CL2
>>> 04 03 9520 08
<<< 80 08 1A432880F1 280000
REM,  Sel CL2
>>> 04 08 9570 1A432880F1 28
<<< 80 06 00 FE51 080000
```

```
REM, All commands below are sent using Parity Framing mode option
which defines the parity bit value coming with data byte.

REM, All commands byte including CRC are sent or received in format
Data Byte Parity Byte.

REM, Special case occur when receiving single nibble for ACK (9004
0A 2400) or NAK (9004 0y 2400) where ‘y’ depends on error code.

REM, Row0 SN0 SN1 SN2 BCC0

REM, Read ROW0 option (cmd CRC1  addr CRC1 CRC2) Parity
automatically included

REM, response Result code Length Data Status

>>> 04 05 30 00 02 A8 08

<<< 80 15 04 CB 8C CB 1A 43 28 80 F1 48 00 00 E1 10 12 00 CF2F 080000


REM, Read ROW0 option Parity Framing (cmd CRC1  addr CRC1 CRC2)
Parity specify after each byte

REM, response Result code Length Data (data byte+parity byte) Status

>>> 04 09 3080 0080 0200 A800 18

<<< 80 27 0400 CB00 8C00 CB00 1A00 4300 2880 8000 F100 4880 0080 0080
E180 1000 1280 0080 CF80 2F00 080000


REM, Read ROW0 option Parity Framing (cmd CRC1  addr CRC1 CRC2)
with error in Parity

REM, Response Result code Length Data (data byte+parity byte) Status

REM, NACK

REM, CR95HFDLL_STCmd

REM, CR95HFDLL_STCmd

REM,  Read ROW1_9 option Parity Framing (cmd CRC1  addr CRC1 CRC2)
Parity specify after each byte   REM

REM,   Row1 SN3 SN4 SN5 SN6

>>> 04 09 3080 0100 8B80 B900 18

<<< 80 27 1A00 4300 2880 8000 F100 4880 0080 0080 E180 1000 1280 0080
0100 0380 A080 1000 1880 7A00 080000


REM,   Row2 BCC1 internal lock byte0 lock byte1

>>> 04 09 3080 0200 1000 8B80 18

<<< 80 27 F100 4880 0080 0080 E180 1000 1280 0080 0100 0380 A080 1000
4480 0380 0B00 D180 A580 4C00 080000
```

```
REM,   Row3 CCFile Magic E1 Ver 10 MMY Size 12 Access 00

>>> 04 09 3080 0380 9980 9A80 18

<<< 80 27 E180 1000 1280 0080 0100 0380 A080 1000 4480 0380 0B00 D180
0100 0700 5580 0100 8880 1300 080000


REM,   Row4 First TL (T01 Prop L 3 V)

>>> 04 09 3080 0400 2600 EE80 18

<<< 80 27 0100 0380 A080 1000 4480 0380 0B00 D180 0100 0700 5580 0100
7300 7480 2E80 6380 5A80 2A00 080000


REM,   Row5 Second TLV NDEF MSG (T03 L 0B V www.st.com)

>>> 04 09 3080 0580 AF80 FF80 18

<<< 80 27 4480 0380 0B00 D180 0100 0700 5580 0100 7300 7480 2E80 6380
6F80 6D00 FE00 0080 3F80 8500 080000


REM,   Row6  Second TLV NDEF MSG  ( T V  www.st.com)

>>> 04 09 3080 0680 3400 CD00 18

<<< 80 27 0100 0700 5580 0100 7300 7480 2E80 6380 6F80 6D00 FE00 0080
0080 FF80 0080 FF80 3F80 F680 080000


REM,   Row7 Secobd TLV NDEF MSG (T V www.st.com)

>>> 04 09 3080 0700 BD80 DC00 18

<<< 80 27 7300 7480 2E80 6380 6F80 6D00 FE00 0080 0080 FF80 0080 FF80
4500 7300 7300 6100 4880 CB00 080000


REM,   Row8 Third TLV Terminator (T V FE 00)

>>> 04 09 3080 0800 4A00 2480 18

<<< 80 27 6F80 6D00 FE00 0080 0080 FF80 0080 FF80 4500 7300 7300 6100
0080 FF80 0080 FF80 F080 4B80 080000


REM,   Row9 Read Scratch pad

>>> 04 09 3080 0980 C380 3580 18

<<< 80 27 0080 FF80 0080 FF80 4500 7300 7300 6100 0080 FF80 0080 FF80
4500 4600 2000 5400 9A80 4880 080000


REM, Write ROW9 option Parity Framing (cmd CRC1  addr CRC1 CRC2)
Parity specify after each byte
```

```
REM, Response Result code Length Data (data byte+parity byte) Status
REM,   Row9 Write Scratch pad (ACK)
>>> 04 11 A200 0980 AA80 5580 AA80 5580 2900 7D80 18
<<< 90 04 0A 240000

REM,   Row9 Read Scratch pad
>>> 04 09 3080 0980 C380 3580 18
<<< 80 27 AA80 5580 AA80 5580 4500 7300 7300 6100 0080 FF80 0080 FF80
4500 4600 2000 5400 1780 B480 080000

REM,   Row9 Write Scratch pad (ACK)
>>> 04 11 A200 0980 0080 FF80 0080 FF80 C800 2780 18
<<< 90 04 0A 240000

REM,   Row9  Read Scratch pad
>>> 04 09 3080 0980 C380 3580 18
<<< 80 27 0080 FF80 0080 FF80 4500 7300 7300 6100 0080 FF80 0080 FF80
4500 4600 2000 5400 9A80 4880 080000

REM,   Select Sector 0 (NACK)
>>> 04 09 C200 FF80 C200 E880 18
<<< 90 04 00 240000

BREAK
>>> CR95HFDLL_RESET_SPI
<<< 8000
>>> CR95HFDLL_ECHO
<<< 5500
>>> CR95HFDLL_IDN
<<< 00 0F 4E46 4320 4653 324A 4153 5434 002ACE
```

D.1.5 NFC Forum Tag Type 4A

```
**** ST95HF code example to support NFC Forum Tag Type 4A (14443-A)
&  NDEF message
REM, 14443B (ST95HF Protocol Selection 14443_A)
REM, first Byte 01 in CR95HFDLL_STCMD is only requested by ST95HF
Development SW
```

```
********** ST95HF setting to support extended Frame Waiting Time
**********

>>> CR95HFDLL_STCMD, 01 020402000180

<<< 0000

REM, Optimization of synchronization between digital and analog
inputs by adjusting TimerW value (default 0x52, min. 0x50, max.
0x60). Recommended value is 0x56 or 0x58 for NFC Forum Tag Type 1
(Topaz).

>>> CR95HFDLL_STCMD, 01 09043A005804

<<< 0000

REM,  Recommended modulation and gain is 0xD1 or 0xD3 for NFC Forum
Tag Type 1 (Topaz).

>>> CR95HFDLL_STCMD, 01 0904680101D1

<<< 0000

REM, last Byte x7 or x8 in CR95HFDLL_SENDRECV  command  number of
bit in the 14443 _Type A  frame

>>> CR95HFDLL_ANTICOLSELECT123


------ ISO14443-A STARTING ANTICOLLISION ALGORITHM ------

ISO14443-A REQA

>>> CR95HFDLL_SENDRECV, 26 07

<<< 80 05  0400  280000

ISO14443-A ANTICOL 1

>>> CR95HFDLL_SENDRECV, 9320 08

<<< 80 08  08192D A29E  280000

ISO14443-A SELECT 1

>>> CR95HFDLL_SENDRECV, 937008192DA29E 28

<<< 80 06  20 FC70  080000

--> UID = 192DA29E   ,  TAG selected

------ ISO14443-A END OF ANTICOLLISION ALGORITHM ------


***   ISO14443A_4 RATS/ATS  (bit rate capability/FDT/CID usage)

>>> CR95HFDLL_SENDRECV, E050 28

<<< 80 0A  057833B003 A0F8  080000


******      ISO14443A_4 PPS   (Protocol parameter data rate)

>>> CR95HFDLL_SENDRECV, D01100 28
```

```
<<< 80 06  D0 7387  080000


**  ISO14443_4 APDU  (command & reply are using Iblock format,
Prolog Information (APDU) Epilog)

*** 7816_ APDU format (Class Instruction, Param ,  Length cmd data
Length expeted)

*** last byte 28 is a control byte to request ST95HF to
automatically happen CRC as Epilog

*** In response first 2 Byte 80 xx and last three bytes 08 0000 are
ST95HF's control bytes

*** Detect & Access NDEF Message

*** Select Application by name

>>> CR95HFDLL_SENDRECV, 02 00 A4040007D2760000850100 28

<<< 80 08  02 9000 F109  080000


*******************      Select CC File by name

>>> CR95HFDLL_SENDRECV, 03 00 A4000002E103 28

<<< 80 08  03 9000 2D53  080000


*******************      ReadBinary  CC (offset Le)

>>> CR95HFDLL_SENDRECV, 02 00 B000000F 28

<<< 80 17  02 000F1000FF00FF0406000100FF0000 9000 B755  080000


*******************      Select NDEF MSG  by Identifier 0001

>>> CR95HFDLL_SENDRECV, 03 00 A40000020001 28

<<< 80 08  03 9000 2D53  080000


*******************     ReadBinary NDEF MSG (MSG Length offset 00 2
bytes)

>>> CR95HFDLL_SENDRECV, 02 00 B0000002 28

<<< 80 0A  02 0015 9000 ABB3  080000


*******************      Select NDEF File by name

>>> CR95HFDLL_SENDRECV, 03 00 A40000020001 28

<<< 80 08  03 9000 2D53  080000


*******************      ReadBinary NDEF (MSG offset 02 , 20 Bytes)
```

```
>>> CR95HFDLL_SENDRECV, 02 00 B0000215 28

<<< 80 1D  02D101115402656E4D32344C5231362074797065 2034 9000 25C5  080000


***      Header D1 type 01 Payload 11 type 54 status 02 english 656E , MSG : M24LR16 type
```

D.2 ISO/IEC 14443 Type B

D.2.1 NFC Forum Tag Type 4B

```
**** ST95HF code example to support NFC Forum Tag Type 4B (14443-B) &  NDEF message

REM, Check ST95HF setting & Protocol selection

REM,  FIELD OFF

REM, first Byte 01 in CR95HFDLL_STCMD is only requested by ST95HF Development SW

>>> CR95HFDLL_STCMD, 01 02020000

<<< 0000

REM, 14443B (ST95HF PROTOCOL Selection 14443_B

>>> CR95HFDLL_STCMD, 01 020403010180

<<< 0000

REM, 14443B Optimization ST95HF Analog Configuration for 144443 (0x30)

>>> CR95HFDLL_STCMD, 01 090468010130

<<< 0000


REM, Access to NFC FORUM TAG Type 4B

REM, REQB 0x 050000 + CRC_B (APf AFI Param (slot0))

REM, Reply ATQB 0x50 4Bytes 4 Bytes 3 Bytes + CRC_B (PUPI AppliData Protocol Info)

REM, Reply from ST95HF 80 0F 50AABBCCDD30ABAB010081E1AE00 00

REM, 80 response OK, 0F nb byte response including tag reply and the ultimate ST95HF status byte 00 (reply OK)

REM, Tag reply 50AABBCCDD30ABAB010081E1AE00

REM, Response code 50

REM, Pupi AABBCCDD

REM, AFI 30 access control
```

```
REM,  CRC_B(AID) ABAB

REM,  Nb Appli (1) 01

REM,  Prot Info byte1   00  (106 Kbps both direction)

REM,  Prot Info byte 2  81( frame max 256 Bytes ISO compliant)
0081E1AE0000

REM,  Prot Info byte 3  E1 (Max frame wait time 4.9 ms Appli
proprietary CID supported)

REM, CRC_B AE00


REM,  14443_3

REM,  REQB ....

>>> CR95HFDLL_STCMD, 01 04 03 050000

<<< 80 0F 50AABBCCDD30ABAB010081E1 AE00 00


REM,  ATTRIB 0x1D PUPI 1byte 1byte 1byte 1 byte  + CRC_B (1D
Identifier Param1 Param2 Param3 Param4)

REM,  Param1   00    use default TR0 TR1 use EOF

REM,  Param2  07     max frame size 106 Kbps Up & Dwn link

REM,  Param3  01     ISO14443 compliant

REM,  Param4 08      CID (8)   card Identifier

REM,  reply ST95HF 80 04 18EBC3 00

REM,  80 response OK  04 nb byte response  including ultimate byte
00 ST95HF reply OK

REM,  Reply 10F9E0  coefBufferLength 1  CID 1  + CRC_B


REM,  ATTRIB ....CID0

>>> CR95HFDLL_STCMD, 01 04 09 1D AABBCCDD00070100

<<< 80 04 10 F9E0 00


REM,  14443_4  ,  CID not used

REM,  APDU  for NDEF management

REM,  command format (INF)  CLA INS P1 P2 Lc(optional)
Data(optional)

REM,  Response (optional ):  body (optional) Sw1 sW2

REM,  Block Format Prolog INFO Epilog ( 02 [CID] [NAD]  [INF] CRC_B
)
```

```
REM,  Sequence lecture NDEF  ( for all  following commands CRC_B is
automatically appends by ST95HF)


REM,  Select application suivant la version du tag (100)
>>> CR95HFDLL_SENDRECV,  02  00 A4 040007D2760000850100
<<< 80 06 029000296A 00
REM,  response 90 00 ok
REM,  response  6A 82 application not found
REM,  Select CC
>>> CR95HFDLL_SENDRECV, 03 00 A4 0000 02 E103
<<< 80 06 03 9000 F530 00
REM,  Read CC
>>> CR95HFDLL_SENDRECV, 02 00 B0 0000 0F
<<< 80 15 02 000F1000FF00FF0406000110020000 9000 E7FA 00
REM, Select Ndef  0001
>>> CR95HFDLL_SENDRECV, 03  00 A4 0000 02 0001
<<< 80 06 03 9000 F530 00
REM, Read Msg Length
>>> CR95HFDLL_SENDRECV, 02 00 B0 0000 02
<<< 80 08 02 0013 9000 53AA 00
REM, Select Ndef  0001
>>> CR95HFDLL_SENDRECV, 03 00 A4 0000 02 0001
<<< 80 06 03 9000 F530 00
REM, Read Message
>>> CR95HFDLL_SENDRECV, 02 00 B0 0002 13
<<< 80 19 02 D1010F5402656E557365204352393548462021 9000 8571 00
```

D.3 ISO/IEC 18092

D.3.1 NFC Forum Tag Type 3

```
REM, ST95HF code example to support NFC Forum Tag Type 3
REM,  TEST INVENTORY ISO/IEC 18092
REM,  RFOFF
>>> CR95HFDLL_STCMD, 01 02020000
<<< 0000
```

```
REM,  Select Protocol 14443C

>>> CR95HFDLL_STCMD, 01 02020451

<<< 0000

REM, ISO/IEC 18092 New Modulation and Gain 0x50

>>> CR95HFDLL_STCMD, 01 090468010150

<<< 0000

REM, ISO/IEC 18092 Enable AutoDetect Filter to synchronize NFC Forum Tag Type 3 with
ST95HF device

>>> CR95HFDLL_STCMD, 01 09040A0102A1

<<< 0000

REM,  REQC 00 FFFF 00 00  (command code System code No request slot 0)

REM, ATQC  80 12  01  010102148E0DB413 (Manuf ID) 100B4B428485D0FF  (Manuf
Parameter)

>>> CR95HFDLL_STCMD, 01 04  05 00FFFF0000

<<< 80 12 01 010102148E0DB413 100B4B428485D0FF  00
```

D.4 ISO/IEC 15693

D.4.1 ISO/IEC 15693 tag

```
REM,  Test Tag ISO/IEC 15693  (LR family)

REM,  Protocol Selection  Up link  Ask 30%  coding 1/4

REM,  Down link Single Sub carrier High data rate

REM,  Inventory One Slot

REM, Command Protocol Select 02 02 01 05


REM,  Protocol Selection

>>> CR95HFDLL_STCMD, 01 02020105

<<< 0000


REM,  Modification of IndexMod & Gain in Analog Value register
@69_index1 0x50

>>> CR95HFDLL_STCMD, 01 090468010150

<<< 0000


REM, Inventory 1 Slot

>>> CR95HFDLL_STCMD, 01 0403 260100
```

```
<<< 80 0D 0000B7100128B42102E0 66CC 00


REM,  GetSystem Info

REM, Flags, UID E00221B4280110B7 DSFID  00 AFI 00 MemorySize 3F
BlockSize 03 IC Reference 21


>>> CR95HFDLL_SENDRECV, 022B

<<< 80 12 00 0F B7100128B42102E000003F03 21 DFB0 00
```

REM, Test Tag ISO/IEC 15693 (Dual family)

```
REM,  Protocol Selection  Up link  Ask 30%  coding 1/4

REM,  Down link Single Sub carrier High data rate

REM,  Inventory 1 Slot

REM, Command Protocol Select 02 02 01 05


REM,  Protocol Selection

>>> CR95HFDLL_STCMD, 01 02020105

<<< 0000


REM,  Modification of IndexMod & Gain in Analog Value register
@69_index1 0x50

>>> CR95HFDLL_STCMD, 01 0904680101 50

<<< 0000


REM, Inventory 1 Slot

>>> CR95HFDLL_STCMD, 01 0403 260100

<<< 80 0D 00FF07062092132C02E0 3D22 00


REM, GetSystem Info

REM, Flags ,UID E0022C1392200607 DSFID  FF AFI 00 MemorySize 07FF
BlockSize 03 IC Reference 2C


>>> CR95HFDLL_SENDRECV, 0A2B

<<< 80 13 00 0F 07062092132C02E0 FF 00 FF07 03 2C 984D  00
```

Appendix E Card emulation communication

The following information is a log of basic exchanges between the CR95HF transceiver and ST95HF card emulator.

`CR95HFDLL_STCMD`: Is the standard ST95HF frame exchange command. In this command, the first byte `01` is not sent, it is only requested by the ST95HF development software in order to recognize if it is a user or service command.

`CR95HFDLL_SENDRECV`: Is the encapsulated CR95HF SendReceive command for which command codes, number of bytes, and CRC are automatically appended to the parameter.

In this section,

- The CR95HF data exchanges are in blue.
- The ST95HF card emulation data exchanges are in green.
- The CR95HF and ST95HF command overhead (command code, length of data and transmission flag) is in **bold**.
- The CRC value is in red.

When the CRC append option is set in the Protocol Select command, the CRC is automatically appended by the ST95HF, but the CRC is not visible in the instruction log file.

When the CRC is present in the command or response, CRC reply is in *italics*.

The following symbols correspond to:

>>> Frame sent by Host to ST95HF
<<< Frame received by Host from ST95HF

Table 58. Example logs

Comment	CR95HF Reader	Direction	ST95HF Card Emulation	Comment
Exchange: Host -> Reader		CR95 ST95	Exchange: Host -> Card Emulator	

Set Reader Config.
ISO/IEC 14443 Type A at 106 Kbps
	-	-	REM, DEMO ST95HF for Card Emulation Exchange
			REM, 0A
			REM, Check serial com with host
			>>> ST95HFDLL_ECHO
			<<< 5500
			>>> ST95HFDLL_IDN
			<<< 000F*4E4643204653324A41535434*002ACE
				Check ST95HF connection to Host

	-	-	REM, POLLFIELD >>> ST95HFDLL_STCMD, 01 0300 <<< 000100	No Field detected
	REM, Reader for Card Emulation Test Max waiting time for manual exchange REM, 0B >>> CR95HFDLL_STCMD, 01 020402000A05 <<< 0000	-	REM, Select Emulation Tag Mode 14443A >>> ST95HFDLL_STCMD, 01 02021208 <<< 0000	Set ST95HF config. Card Emulator ISO/IEC 14443 Type A at 106 Kbps

Table 58. Example logs (continued)

Comment	CR95HF Reader	Direction	ST95HF Card Emulation	Comment

Set Reader Config.
ISO/IEC 14443 Type A at 106 Kbps (cont.)
	REM, 02000180
	REM, CR95HFDll_STCmd
	REM, CR95HFDll_STCmd 01 09043A00xx04
	REM, Default xx 52 Max 5A
	>>> CR95HFDLL_STCMD, 01 09043A005804
	<<< 0000
	REM, Increase demodulator gain D3
	>>> CR95HFDLL_STCMD, 01 0904680101D3
	<<< 0000
	REM, increase Dec tolerance
	>>> CR95HFDLL_STCMD, 01 09110B00241204880F1F3F3F01000505005500
	<<< 0000
		-	REM, Read Default value Card Emulator Configuration register ACCA
			REM, ACCA (7:6 card clamp 00 default
			REM, ACCA (5:4 Card Demod 01 noise immunity low sensitivity 10%
			REM, ACCA (3:0 mod depth default 7 mon 0 max F)
			REM, READ Default value Reg ACCA Set Reg Index @4
			>>> ST95HFDLL_STCMD, 01 0903680004
			<<< 0000
			REM, READ Reg ACCA card config 1 Reg @69 index 4
			>>> ST95HFDLL_STCMD, 01 0803690100
			<<< 000127
				Check Configuration

	BREAK	-	REM, CONFIGURE CARD EMULATOR (Set UID) REM, Set Anti-collision Filter Type A UID 0251744AEF2280 REM, ATQA 4403 SAK 20 UID1 88025174 UID2 4AEF2280 >>> ST95HFDLL_STCMD, 01 0D0B4403208880251744AEF2280 <<< 0000 BREAK	Set ST95HF Identity

	-	-	REM, POLLFIELD
			>>> ST95HFDLL_STCMD, 01 0300
			<<< 000101
			REM, LISTEN
			>>> ST95HFDLL_STCMD, 01 0500
			<<< 0000
			BREAK
				Check RF Field On
				ST95HF Set in Listening mode

Table 58. Example logs (continued)

Comment	CR95HF Reader	Direction	ST95HF Card Emulation	Comment
Execute Anti-collision Process ISO/IEC 14443 Type A	>>> CR95HFDLL_ANTICOLSELECT123		-	Anti-collision Process on RX side (no return to Host) Response automatically sent to reader ISO/IEC 14443-3
	------ ISO14443-A Start of Anti-Collision Algorithm ------			
	ISO14443-A REQA			
	>>> CR95HFDLL_SENDRECV, 2607	>		
	<<< 80054403280000	<		
	ISO14443-A ANTICOL 1			
	>>> CR95HFDLL_SENDRECV, 932008	>		
	<<< 800888025174A9280000	<		
	ISO14443-A SELECT 1			
	>>> CR95HFDLL_SENDRECV, 937088025174A928	>		
	<<< 800624D836080000	<		
	ISO14443-A ANTICOL 2			
	>>> CR95HFDLL_SENDRECV, 952008	>		
	<<< 80084AEF228007280000	<		
	ISO14443-A SELECT 2			
	>>> CR95HFDLL_SENDRECV, 95704AEF22800728	>		
	<<< 800620FC70080000	<		
	--> UID = 0251744AEF2280			
	--> TAG selected			
	------ ISO14443-A End of Anti-Collision Algorithm ------			
	BREAK			

Table 58. Example logs (continued)

Comment	CR95HF Reader	Direction	ST95HF Card Emulation	Comment
Applicative Communication ISO/IEC 14443-4: Data Exchange between Reader (CR95HF) & Card Emulator (ST95HF)				
Application comm protocol setup ISO/IEC 14443-4	-	-	-	ST95HF is in Listening mode
	REM, RATS 1A >>> CR95HFDLL_SENDRECV, *E050***28**	>	>>> ST95HFDLL_POLLING_READING <<< **8005***E050***BCA508** REM, 1B SEND ATS 060706757781028028	After capture, Data is transferred to Host which defines the response
	<<< **800F***06757781028***002F***00800* BREAK	<	>>> ST95HFDLL_STCMD, 01 **060B***06757781028002F00800***28** <<< 0000	
	-	-	REM, LISTEN >>> ST95HFDLL_STCMD, 01 0500 <<< 0000 BREAK	ST95HF returns to Listening mode
	REM, PPS 2A >>> CR95HFDLL_SENDRECV, *D01100***28**	>	>>> ST95HFDLL_POLLING_READING <<< **8006***D0110052A6***08** REM, 2B PPS	After capture, Data is transferred to Host which defines the response
	<<< **800A***D0***7387***0800*	<	>>> ST95HFDLL_STCMD, 01 **0606***D07387*0800**28** <<< 0000	
	BREAK	-	REM, LISTEN >>> ST95HFDLL_STCMD, 01 0500 <<< 0000 BREAK	ST95HF returns to Listening mode

Table 58. Example logs (continued)

Comment	CR95HF Reader	Direction	ST95HF Card Emulation	Comment

Application comm protocol setup
ISO/IEC 14443-4 (cont.)
	REM, IBLOCK Exchange 3A
	>>> CR95HFDLL_SENDRECV, *0200A40400007D2760000860100***28**
	<<< **800C***029000***F109***0800*
		>
		<
			>>> ST95HFDLL_POLLING_READING
			<<< **8010***0200A4040007D27600008601004***4BF708**
			REM, 3B IBLOCK
			>>> ST95HFDLL_STCMD, 01 **0608***029000F1090800***28**
			<<< 0000
				After capture, Data is transferred to Host which defines the response

	-	-	REM, LISTEN >>> ST95HFDLL_STCMD, 01 0500 <<< 0000 BREAK	ST95HF returns to Listening mode
	REM, Set RF OFF >>> CR95HFDLL_STCMD, 01 02020000 <<< 0000	-	>>> ST95HFDLL_POLLING_READING <<< **8F00**	No Field error
Close SPI link	BREAK >>> CR95HFDLL_RESET_SPI <<< 0000	-	BREAK REM, 4B >>> ST95HFDLL_RESET_SPI <<< 0000	Close SPI link

Revision history

Table 59. Document revision history

Date	Revision	Changes
14-Apr-2014	1	Initial release.
10-Jun-2014	2	Updated *Section 3.2: Startup sequence on page 13* and *Table 44: Absolute maximum ratings on page 52*.
20-Aug-2014	3	Updated *Applications on page 1*, *Table 16: SendRecv command description on page 27* and *Table 18: List of <Data> Response values for the SendRecv command for different protocols on page 30*.
08-Feb-2016	4	Updated *Features* and *Applications* on cover page. Updated *Section 1: Description* and *Section 7: Package information*. Updated *Table 1: List of terms* and *Figure 5: Power-up sequence*. Removed former *Table 52: DC characteristics (VPS_Main = 3V±10% or 5V±10% and VPS_TX = 5V±10%)*.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.