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

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

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

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

MFRC630

MFRC630 and MFRC630 *plus*: High-performance frontend for MIFARE and NTAG products

Rev. 4.6 — 15 May 2018
227546

Product data sheet
COMPANY PUBLIC

1 General description

MFRC630 and MFRC630 *plus*, the high-performance frontend for MIFARE and NTAG products.

The MFRC630 multi-protocol NFC frontend IC supports the following operating modes:

- Read/write mode supporting ISO/IEC 14443A and MIFARE Classic
- Read/write mode supporting NTAG

The MFRC630's internal transmitter is able to drive a reader/writer antenna designed to communicate with ISO/IEC 14443A and MIFARE Cassic IC-based cards and transponders without additional active circuitry. The digital module manages the complete ISO/IEC 14443A framing and error detection functionality (parity and CRC).

The MFRC630 supports MIFARE Classic with 1 kB memory, MIFARE Classic with 4 kB memory, MIFARE Ultralight, MIFARE Ultralight C, MIFARE Plus and MIFARE DESFire products. The MFRC630 supports higher transfer speeds of the MIFARE product family of up to 848 kbit/s in both directions.

The following host interfaces are supported:

- Serial Peripheral Interface (SPI)
- Serial UART (similar to RS232 with voltage levels dependent on pin voltage supply)
- I^2C-bus interface (two versions are implemented: I2C and I2CL)

The MFRC630 supports the connection of a secure access module (SAM). A dedicated separate I^2C interface is implemented for a connection of the SAM. The SAM can be used for high secure key storage and acts as a very performant crypto coprocessor. A dedicated SAM is available for connection to the MFRC630.

In this document the term „MIFARE Classic card“ refers to a MIFARE Classic IC-based contactless card.

2 Features and benefits

- High-performance multi-protocol NFC frontend for transfer speed up to 848 kbit/s
- Supports ISO/IEC 14443 A, MIFARE Classic and NTAG
- Supports MIFARE Classic product encryption by hardware in read/write mode
  Allows reading cards based on MIFARE Ultralight, MIFARE Classic with 1 kB memory , MIFARE Classic with 4 kB memory, MIFARE DESFire EV1, MIFARE DESFire EV2 and MIFARE Plus ICs
- Low-power card detection
- Antenna connection with minimum number of external components
- Supported host interfaces:
  - SPI up to 10 Mbit/s
  - I^2C-bus interfaces up to 400 kBd in Fast mode, up to 1000 kBd in Fast mode plus
  - RS232 Serial UART up to 1228.8 kBd, with voltage levels dependent on pin voltage supply
- Separate I^2C-bus interface for connection of a secure access module (SAM)
- FIFO buffer with size of 512 byte for highest transaction performance
- Flexible and efficient power saving modes including hard power down, standby and low-power card detection
- Cost saving by integrated PLL to derive system clock from 27.12 MHz RF quartz crystal
- 3 V to 5.5 V power supply (MFRC63002)
  2.5 V to 5.5 V power supply (MFRC63003)
- Up to 8 free programmable input/output pins
- Typical operating distance in read/write mode for communication to a ISO/IEC 14443A and MIFARE Classic card up to 12 cm, depending on the antenna size and tuning. The version MFRC63003 offers a more flexible configuration for Low-Power Card detection compared to the MFRC63002 with the new register LPCD_OPTIONS. In addition, the MFRC63003 offers new additional settings for the Load Protocol which fit very well to smaller antennas. The MFRC63003 is therefore the recommended version for new designs.

3 Quick reference data

Table 1. Quick reference data MFRC63002HN

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{DD}	supply voltage			3.0	5.0	5.5	V
$V_{DD(PVDD)}$	PVDD supply voltage		[1]	3.0	5.0	V_{DD}	V
$V_{DD(TVDD)}$	TVDD supply voltage			3.0	5.0	5.5	V
I_{pd}	power-down current	PDOWN pin pulled HIGH	[2]	-	8	40	nA
I_{DD}	supply current			-	17	20	mA
$I_{DD(TVDD)}$	TVDD supply current			-	100	250	mA
T_{amb}	operating ambient temperature			-25	+25	+85	°C
T_{stg}	storage temperature	no supply voltage applied		-55	+25	+125	°C

[1] VDD(PVDD) must always be the same or lower voltage than VDD.
[2] I_{pd} is the sum of all supply currents

Table 2. Quick reference data MFRC63003HN

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{DD}	supply voltage			2.5	5.0	5.5	V
$V_{DD(PVDD)}$	PVDD supply voltage		[1]	2.5	5.0	V_{DD}	V
$V_{DD(TVDD)}$	TVDD supply voltage			2.5	5.0	5.5	V
I_{pd}	power-down current	PDOWN pin pulled HIGH	[2]	-	8	40	nA
I_{DD}	supply current			-	17	20	mA
$I_{DD(TVDD)}$	TVDD supply current			-	180	350	mA
		absolute limiting value		-	-	500	mA
T_{amb}	operating ambient temperature	device mounted on PCB which allows sufficient heat dissipation		-40	+25	+105	°C
T_{stg}	storage temperature	no supply voltage applied		-55	+25	+125	°C

[1] VDD(PVDD) must always be the same or lower voltage than VDD.
[2] I_{pd} is the sum of all supply currents

4 Ordering information

Table 3. Ordering information

Type number	Package		
	Name	Description	Version
MFRC63002HN/TRAYB[1]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; MSL1, 32 terminals + 1 central ground; body 5 × 5 × 0.85 mm	SOT617-1
MFRC63002HN/TRAYBM[2]			
MFRC63002HN/T/R[3]			
MFRC63003HN/TRAYB[4]		plastic thermal enhanced very thin quad flat package; no leads; MSL2, 32 terminals + 1 central ground; body 5 × 5 × 0.85 mm, wettable flanks	
MFRC63003HN/T/R[5]			

[1] Delivered in one tray
[2] Delivered in five trays
[3] Delivered on reel with 6000 pieces
[4] Delivered in one tray, MOQ (Minimum order quantity) : 490 pcs
[5] Delivered on reel with 6000 pieces; MOQ (Minimum order quantity) : 6000 pcs

5 Block diagram

The analog interface handles the modulation and demodulation of the antenna signals for the contactless interface.

The contactless UART manages the protocol dependency of the contactless interface settings managed by the host.

The FIFO buffer ensures fast and convenient data transfer between host and the contactless UART.

The register bank contains the settings for the analog and digital functionality.

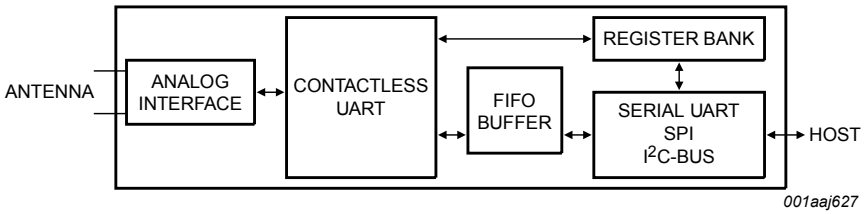
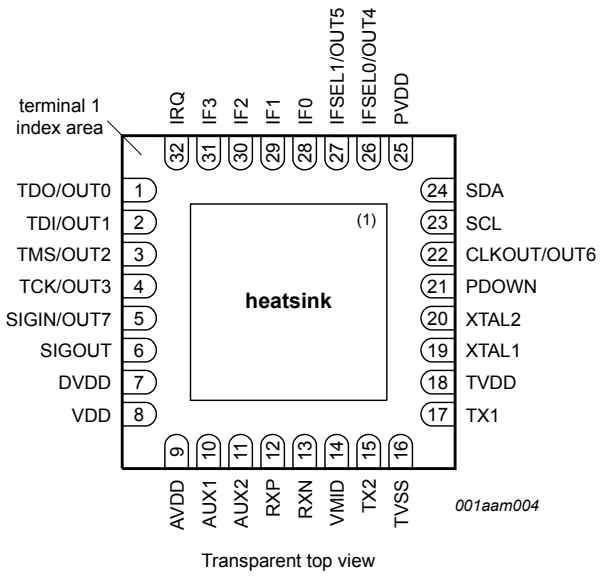


Figure 1. Simplified block diagram of the MFRC630

6 Pinning information



Transparent top view

1. VSS - heatsink connection

Figure 2. Pinning configuration HVQFN32 (SOT617-1)

6.1 Pin description

Table 4. Pin description

Pin	Symbol	Type	Description
1	TDO / OUT0	O	test data output for boundary scan interface / general purpose output 0
2	TDI / OUT1	I	test data input boundary scan interface / general purpose output 1
3	TMS / OUT2	I	test mode select boundary scan interface / general purpose output 2
4	TCK / OUT3	I	test clock boundary scan interface / general purpose output 3
5	SIGIN /OUT7	I/O	Contactless communication interface output. / general purpose output 7
6	SIGOUT	O	Contactless communication interface input.
7	DVDD	PWR	digital power supply buffer [1]
8	VDD	PWR	power supply
9	AVDD	PWR	analog power supply buffer [1]
10	AUX1	O	auxiliary outputs: Pin is used for analog test signal
11	AUX2	O	auxiliary outputs: Pin is used for analog test signal
12	RXP	I	receiver input pin for the received RF signal.
13	RXN	I	receiver input pin for the received RF signal.
14	VMID	PWR	internal receiver reference voltage [1]
15	TX2	O	transmitter 2: delivers the modulated 13.56 MHz carrier
16	TVSS	PWR	transmitter ground, supplies the output stage of TX1, TX2
17	TX1	O	transmitter 1: delivers the modulated 13.56 MHz carrier

Pin	Symbol	Type	Description
18	TVDD	PWR	transmitter voltage supply
19	XTAL1	I	crystal oscillator input: Input to the inverting amplifier of the oscillator. This pin is also the input for an externally generated clock (fosc = 27.12 MHz)
20	XTAL2	O	crystal oscillator output: output of the inverting amplifier of the oscillator
21	PDOWN	I	Power Down (RESET)
22	CLKOUT / OUT6	O	clock output / general purpose output 6
23	SCL	O	Serial Clock line
24	SDA	I/O	Serial Data Line
25	PVDD	PWR	pad power supply
26	IFSEL0 / OUT4	I	host interface selection 0 / general purpose output 4
27	IFSEL1 / OUT5	I	host interface selection 1 / general purpose output 5
28	IF0	I/O	interface pin, multifunction pin: Can be assigned to host interface RS232, SPI, I^2C, I^2C-L
29	IF1	I/O	interface pin, multifunction pin: Can be assigned to host interface SPI, I^2C, I^2C-L
30	IF2	I/O	interface pin, multifunction pin: Can be assigned to host interface RS232, SPI, I^2C, I^2C-L
31	IF3	I/O	interface pin, multifunction pin: Can be assigned to host interface RS232, SPI, I^2C, I^2C-L
32	IRQ	O	interrupt request: output to signal an interrupt event
33	VSS	PWR	ground and heat sink connection

[1] This pin is used for connection of a buffer capacitor. Connection of a supply voltage might damage the device.

7 Functional description

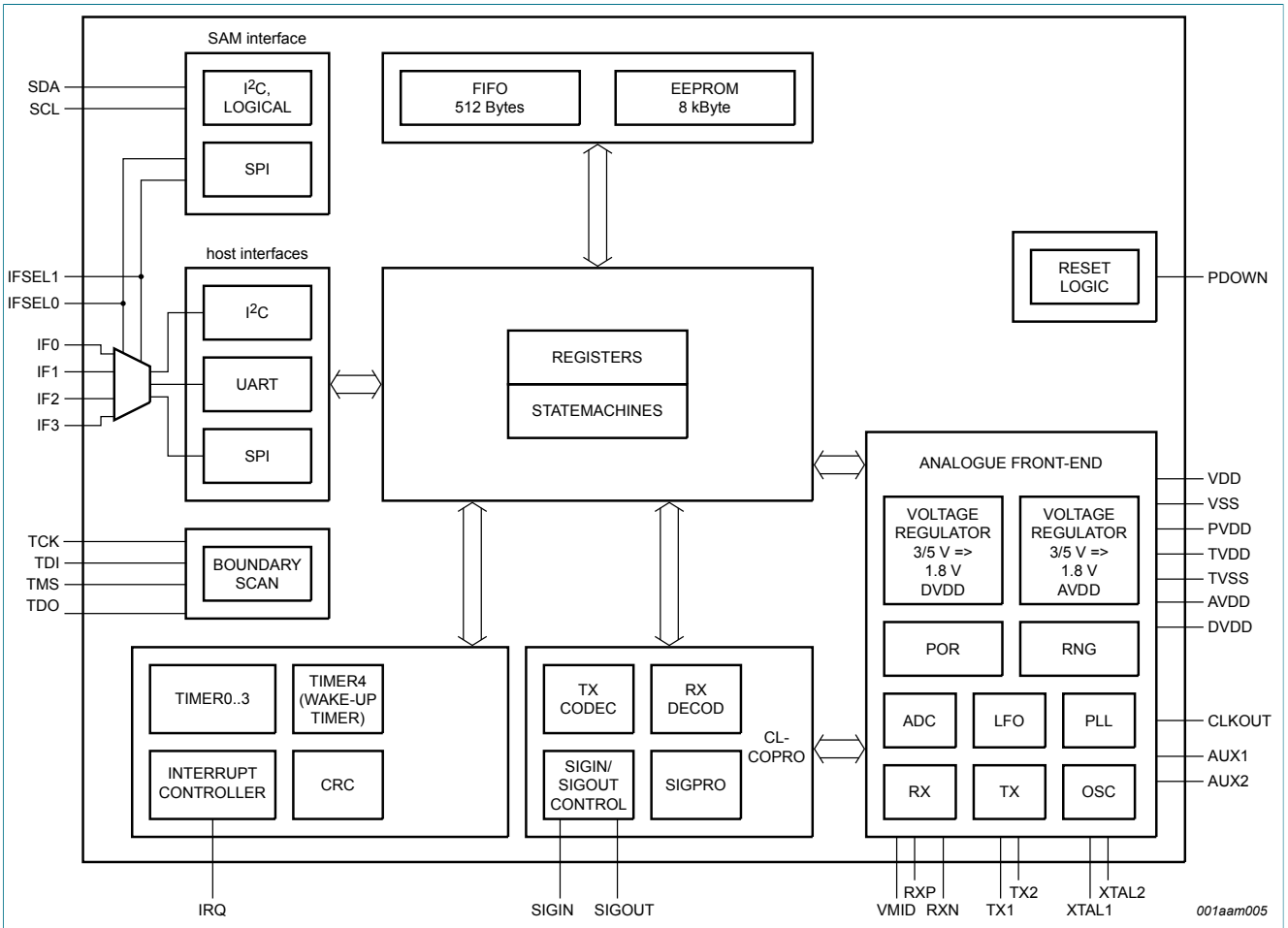


Figure 3. Detailed block diagram of the MFRC630

7.1 Interrupt controller

The interrupt controller handles the enabling/disabling of interrupt requests. All of the interrupts can be configured by firmware. Additionally, the firmware has possibilities to trigger interrupts or clear pending interrupt requests. Two 8-bit interrupt registers IRQ0 and IRQ1 are implemented, accompanied by two 8-bit interrupt enable registers IRQ0En and IRQ1En. A dedicated functionality of bit 7 to set and clear bits 0 to 6 in this interrupt controller registers is implemented.

The MFRC630 indicates certain events by setting bit IRQ in the register Status1Reg and additionally, if activated, by pin IRQ. The signal on pin IRQ may be used to interrupt the host using its interrupt handling capabilities. This allows the implementation of efficient host software.

Table 4. shows the available interrupt bits, the corresponding source and the condition for its activation. The interrupt bits Timer0IRQ, Timer1IRQ, Timer2IRQ, Timer3OIRQ, in register IRQ1 indicate an interrupt set by the timer unit. The setting is done if the timer underflows.

The TxIRQ bit in register IRQ0 indicates that the transmission is finished. If the state changes from sending data to transmitting the end of the frame pattern, the transmitter unit sets the interrupt bit automatically.

The bit RxIRQ in register IRQ0 indicates an interrupt when the end of the received data is detected.

The bit IdleIRQ in register IRQ0 is set if a command finishes and the content of the command register changes to idle.

The register WaterLevel defines both - minimum and maximum warning levels - counting from top and from bottom of the FIFO by a single value.

The bit HiAlertIRQ in register IRQ0 is set to logic 1 if the HiAlert bit is set to logic 1, that means the FIFO data number has reached the top level as configured by the register WaterLevel and bit WaterLevelExtBit.

The bit LoAlertIRQ in register IRQ0 is set to logic 1 if the LoAlert bit is set to logic 1, that means the FIFO data number has reached the bottom level as configured by the register WaterLevel.

The bit ErrIRQ in register IRQ0 indicates an error detected by the contactless UART during receive. This is indicated by any bit set to logic 1 in register Error.

The bit LPCDIRQ in register IRQ0 indicates a card detected.

The bit RxSOFIRQ in register IRQ0 indicates a detection of a SOF or a subcarrier by the contactless UART during receiving.

The bit GlobalIRQ in register IRQ1 indicates an interrupt occurring at any other interrupt source when enabled.

Table 5. Interrupt sources

Interrupt bit	Interrupt source	Is set automatically, when
Timer0IRQ	Timer Unit	the timer register T0 CounterVal underflows
Timer1IRQ	Timer Unit	the timer register T1 CounterVal underflows
Timer2IRQ	Timer Unit	the timer register T2 CounterVal underflows
Timer3IRQ	Timer Unit	the timer register T3 CounterVal underflows
TxIRQ	Transmitter	a transmitted data stream ends
RxIRQ	Receiver	a received data stream ends
IdleIRQ	Command Register	a command execution finishes
HiAlertIRQ	FIFO-buffer pointer	the FIFO data number has reached the top level as configured by the register WaterLevel
LoAlertIRQ	FIFO-buffer pointer	the FIFO data number has reached the bottom level as configured by the register WaterLevel
ErrIRQ	contactless UART	a communication error had been detected
LPCDIRQ	LPCD	a card was detected when in low-power card detection mode
RxSOFIRQ	Receiver	detection of a SOF or a subcarrier
GlobalIRQ	all interrupt sources	will be set if another interrupt request source is set

7.2 Timer module

Timer module overview

The MFRC630 implements five timers. Four timers -Timer0 to Timer3 - have an input clock that can be configured by register T(x)Control to be 13.56 MHz, 212 kHz, (derived from the 27.12 MHz quartz) or to be the underflow event of the fifth Timer (Timer4). Each timer implements a counter register which is 16 bit wide. A reload value for the counter is defined in a range of 0000h to FFFFh in the registers TxReloadHi and TxReloadLo. The fifth timer Timer4 is intended to be used as a wakeup timer and is connected to the internal LFO (Low Frequency Oscillator) as input clock source.

The TControl register allows the global start and stop of each of the four timers Timer0 to Timer3. Additionally, this register indicates if one of the timers is running or stopped. Each of the five timers implements an individual configuration register set defining timer reload value (e.g. T0ReloadHi,T0ReloadLo), the timer value (e.g. T0CounterValHi, T0CounterValLo) and the conditions which define start, stop and clockfrequency (e.g. T0Control).

The external host may use these timers to manage timing relevant tasks. The timer unit may be used in one of the following configurations:

- Time-out counter
- Watch-dog counter
- Stop watch
- Programmable one-shot timer
- Periodical trigger

The timer unit can be used to measure the time interval between two events or to indicate that a specific event has occurred after an elapsed time. The timer register content is modified by the timer unit, which can be used to generate an interrupt to allow an host to react on this event.

The counter value of the timer is available in the registers T(x)CounterValHi, T(x)CounterValLo. The content of these registers is decremented at each timer clock.

If the counter value has reached a value of 0000h and the interrupts are enabled for this specific timer, an interrupt will be generated as soon as the next clock is received.

If enabled, the timer event can be indicated on the pin IRQ (interrupt request). The bit Timer(x)IRQ can be set and reset by the host controller. Depending on the configuration, the timer will stop counting at 0000h or restart with the value loaded from registers T(x)ReloadHi, T(x)ReloadLo.

The counting of the timer is indicated by bit TControl.T(x)Running.

The timer can be started by setting bits TControl.T(x)Running and TControl.T(x)StartStopNow or stopped by setting the bits TControl.T(x)StartStopNow and clearing TControl.T(x)Running.

Another possibility to start the timer is to set the bit T(x)Mode.T(x)Start, this can be useful if dedicated protocol requirements need to be fulfilled.

7.2.1 Timer modes

7.2.1.1 Time-Out- and Watch-Dog-Counter

Having configured the timer by setting *register T(x)ReloadValue* and starting the counting of Timer(x) by setting bit TControl.T(x)StartStop and TControl.T(x)Running, the timer unit decrements the *T(x)CounterValue* Register beginning with the configured start event. If the configured stop event occurs before the Timer(x) underflows (e.g. a bit is received from the card), the timer unit stops (no interrupt is generated).

If no stop event occurs, the timer unit continues to decrement the counter registers until the content is zero and generates a timer interrupt request at the next clock cycle. This allows to indicate to a host that the event did not occur during the configured time interval.

7.2.1.2 Wake-up timer

The wake-up Timer4 allows to wakeup the system from standby after a predefined time. The system can be configured in such a way that it is entering the standby mode again in case no card had been detected.

This functionality can be used to implement a low-power card detection (LPCD). For the low-power card detection it is recommended to set T4Control.T4AutoWakeUp and T4Control.T4AutoRestart, to activate the Timer4 and automatically set the system in standby. The internal low frequency oscillator (LFO) is then used as input clock for this Timer4. If a card is detected the host-communication can be started. If bit T4Control.T4AutoWakeUp is not set, the MFRC630 will not enter the standby mode again in case no card is detected but stays fully powered.

7.2.1.3 Stop watch

The elapsed time between a configured start- and stop event may be measured by the MFRC630 timer unit. By setting the registers *T(x)ReloadValueHi, T(x)reloadValueLo* the timer starts to decrement as soon as activated. If the configured stop event occurs, the timers stops decrementing. The elapsed time between start and stop event can then be calculated by the host dependent on the timer interval TTimer:

$$\Delta T = \left(Treload_{value} - Timer_{value} \right) {}^{*} T_{Timer} \quad (1)$$

If an underflow occurred which can be identified by evaluating the corresponding IRQ bit, the performed time measurement according to the formula above is not correct.

7.2.1.4 Programmable one-shot timer

The host configures the interrupt and the timer, starts the timer and waits for the interrupt event on pin IRQ. After the configured time the interrupt request will be raised.

7.2.1.5 Periodical trigger

If the bit *T(x)Control.T(x)AutoRestart is set and the interrupt is activated*, an interrupt request will be indicated periodically after every elapsed timer period.

7.3 Contactless interface unit

The contactless interface unit of the MFRC630 supports the following read/write operating modes:

- ISO/IEC14443 type A and MIFARE Classic

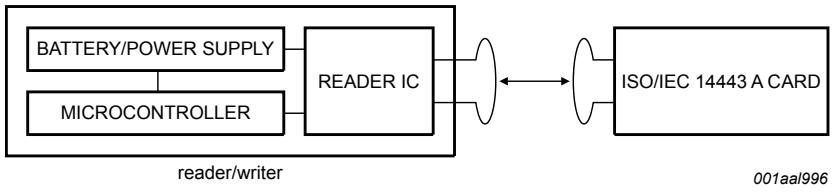
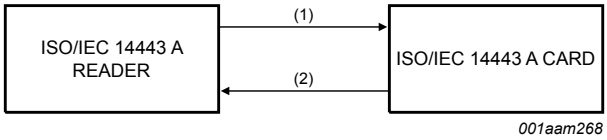


Figure 4. Read/write mode

A typical system using the MFRC630 is using a microcontroller to implement the higher levels of the contactless communication protocol and a power supply (battery or external supply).

7.3.1 Communication mode for ISO/IEC 14443 type A and for MIFARE Classic

The physical level of the communication is shown in Figure 5.



1. Reader to Card 100 % ASK, Miller Coded, Transfer speed 106 kbit/s to 848 kbit/s
2. Card to Reader, Subcarrier Load Modulation Manchester Coded or BPSK, transfer speed 106 kbit/s to 848 kbit/s

Figure 5. Read/write mode for ISO/IEC 14443 type A and read/write mode for MIFARE Classic

The physical parameters are described in Table 5.

Table 6. Communication overview for ISO/IEC 14443 type A and read/write mode for MIFARE Classic

Communication direction	Signal type	Transfer speed			
		106 kbit/s	212 kbit/s	424 kbit/s	848 kbit/s
Reader to card (send data from the MFRC630 to a card) fc = 13.56 MHz	reader side modulation	100 % ASK	100% ASK	100% ASK	100% ASK
	bit encoding	modified Miller encoding	modified Miller encoding	modified Miller encoding	modified Miller encoding
	bit rate [kbit/s]	fc / 128	fc / 64	fc / 32	fc / 16
Card to reader (MFRC630 receives data from a card)	card side modulation	subcarrier load modulation	subcarrier load modulation	subcarrier load modulation	subcarrier load modulation
	subcarrier frequency	fc / 16	fc / 16	fc / 16	fc / 16
	bit encoding	Manchester encoding	BPSK	BPSK	BPSK

The MFRC630 connection to a host is required to manage the complete ISO/IEC 14443 type A and MIFARE Classic communication protocol. Figure 6 shows the data coding and framing according to ISO/IEC 14443 type A and MIFARE Classic.

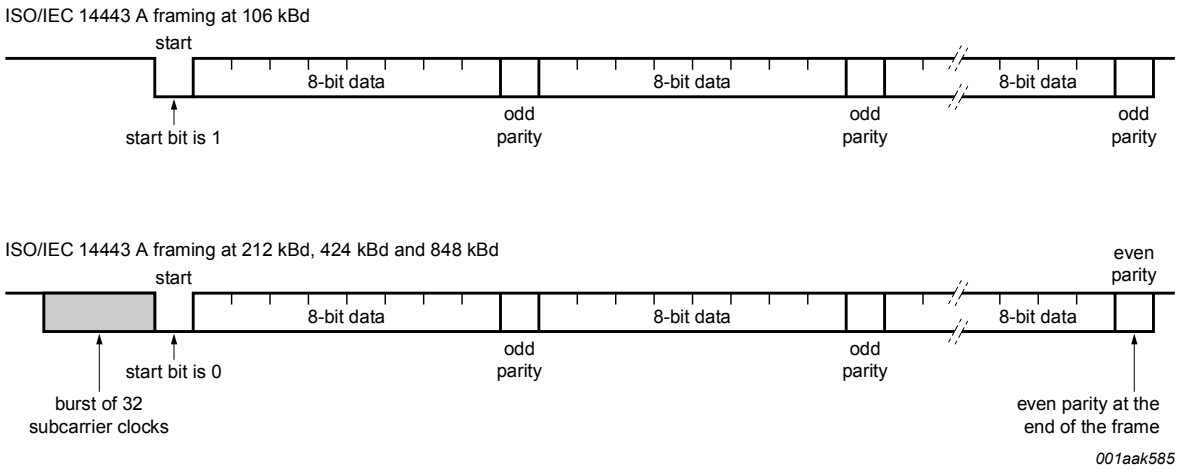


Figure 6. Data coding and framing according to ISO/IEC 14443 A

The internal CRC coprocessor calculates the CRC value based on ISO/IEC 14443 A part 3 and handles parity generation internally according to the transfer speed.

7.4 Host interfaces

7.4.1 Host interface configuration

The MFRC630 supports direct interfacing of various hosts as the SPI, I^2C, I^2CL and serial UART interface type. The MFRC630 resets its interface and checks the current host interface type automatically having performed a power-up or resuming from power down. The MFRC630 identifies the host interface by the means of the logic levels on the control pins after the Cold Reset Phase. This is done by a combination of fixed pin connections.The following table shows the possible configurations defined by IFSEL1,IFSEL0:

Table 7. Connection scheme for detecting the different interface types

Pin	Pin Symbol	UART	SPI	I^2C	I^2C-L
28	IF0	RX	MOSI	ADR1	ADR1
29	IF1	n.c.	SCK	SCL	SCL
30	IF2	TX	MISO	ADR2	SDA
31	IF3	PAD_VDD	NSS	SDA	ADR2
26	IFSEL0	VSS	VSS	PAD_VDD	PAD_VDD
27	IFSEL1	VSS	PAD_VDD	VSS	PAD_VDD

7.4.2 SPI interface

7.4.2.1 General

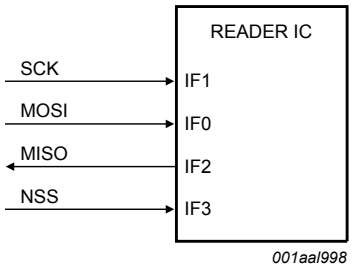


Figure 7. Connection to host with SPI

The MFRC630 acts as a slave during the SPI communication. The SPI clock SCK has to be generated by the master. Data communication from the master to the slave uses the Line MOSI. Line MISO is used to send data back from the MFRC630 to the master.

A serial peripheral interface (SPI compatible) is supported to enable high speed communication to a host. The implemented SPI compatible interface is according to a standard SPI interface. The SPI compatible interface can handle data speed of up to 10 Mbit/s. In the communication with a host MFRC630 acts as a slave receiving data from the external host for register settings and to send and receive data relevant for the communication on the RF interface.

NSS (Not Slave Select) enables or disables the SPI interface. When NSS is logical high, the interface is disabled and reset. Between every SPI command the NSS must go to logical high to be able to start the next command read or write.

On both data lines (MOSI, MISO) each data byte is sent by MSB first. Data on MOSI line shall be stable on rising edge of the clock line (SCK) and is allowed to change on falling edge. The same is valid for the MISO line. Data is provided by the MFRC630 on the falling edge and is stable on the rising edge.The polarity of the clock is low at SPI idle.

7.4.2.2 Read data

To read out data from the MFRC630 by using the SPI compatible interface the following byte order has to be used.

The first byte that is sent defines the mode (LSB bit) and the address.

Table 8. Byte Order for MOSI and MISO

	byte 0	byte 1	byte 2	byte 3 to n-1	byte n	byte n+1
MOSI	address 0	address 1	address 2	……..	address n	00h
MISO	X	data 0	data 1	……..	data n - 1	data n

Remark: The Most Significant Bit (MSB) has to be sent first.

7.4.2.3 Write data

To write data to the MFRC630 using the SPI interface the following byte order has to be used. It is possible to write more than one byte by sending a single address byte (see.8.5.2.4).

The first send byte defines both, the mode itself and the address byte.

Table 9. Byte Order for MOSI and MISO

	byte 0	byte 1	byte 2	3 to n-1	byte n	byte n + 1
MOSI	address 0	data 0	data 1	……..	data n - 1	data n
MISO	X	X	X	……..	X	X

Remark: The Most Significant Bit (MSB) has to be sent first.

7.4.2.4 Address byte

The address byte has to fulfil the following format:

The LSB bit of the first byte defines the used mode. To read data from the MFRC630 the LSB bit is set to logic 1. To write data to the MFRC630 the LSB bit has to be cleared. The bits 6 to 0 define the address byte.

NOTE: When writing the sequence [address byte][data0][data1][data2]..., [data0] is written to address [address byte], [data1] is written to address [address byte + 1] and [data2] is written to [address byte + 2].

Exception: This auto increment of the address byte is not performed if data is written to the FIFO address

Table 10. Address byte 0 register; address MOSI

7	6	5	4	3	2	1	0
address 6	address 5	address 4	address 3	address 2	address 1	address 0	1 (read) 0 (write)
MSB							LSB

7.4.2.5 Timing Specification SPI

The timing condition for SPI interface is as follows:

Table 11. Timing conditions SPI

Symbol	Parameter	Min	Typ	Max	Unit
t_{SCKL}	SCK LOW time	50	-	-	ns
t_{SCKH}	SCK HIGH time	50	-	-	ns
$t_{h(SCKH-D)}$	SCK HIGH to data input hold time	25	-	-	ns
$t_{su(D-SCKH)}$	data input to SCK HIGH set-up time	25	-	-	ns
$t_{h(SCKL-Q)}$	SCK LOW to data output hold time	-	-	25	ns
$t_{(SCKL-NSSH)}$	SCK LOW to NSS HIGH time	0	-	-	ns
t_{NSSH}	NSS HIGH time	50	-	-	ns

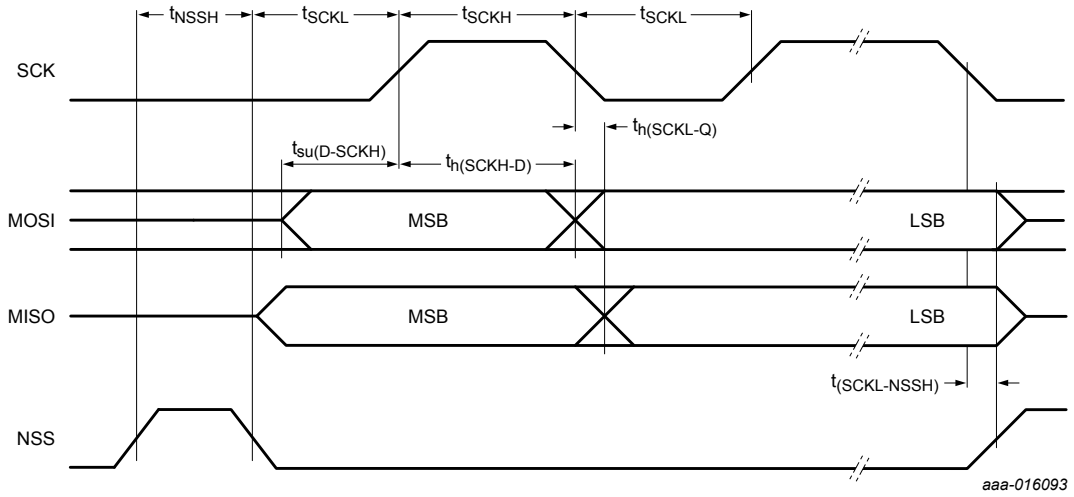


Figure 8. Connection to host with SPI

Remark: To send more bytes in one data stream the NSS signal must be LOW during the send process. To send more than one data stream the NSS signal must be HIGH between each data stream.

7.4.3 RS232 interface

7.4.3.1 Selection of the transfer speeds

The internal UART interface is compatible to a RS232 serial interface. The levels supplied to the pins are between VSS and PVDD. To achieve full compatibility of the voltage levels to the RS232 specification, a RS232 level shifter is required.

Table 12 "Selectable transfer speeds" describes examples for different transfer speeds and relevant register settings. The resulting transfer speed error is less than 1.5 % for all described transfer speeds. The default transfer speed is 115.2 kbit/s.

To change the transfer speed, the host controller has to write a value for the new transfer speed to the register SerialSpeedReg. The bits BR_T0 and BR_T1 define factors to set the transfer speed in the SerialSpeedReg.

Table 11 "Settings of BR_T0 and BR_T1" describes the settings of BR_T0 and BR_T1.

Table 12. Settings of BR_T0 and BR_T1

BR_T0	0	1	2	3	4	5	6	7
factor BR_T0	1	1	2	4	8	16	32	64
range BR_T1	1 to 32	33 to 64	33 to 64	33 to 64	33 to 64	33 to 64	33 to 64	33 to 64

Table 13. Selectable transfer speeds

Transfer speed (kbit/s)	Serial SpeedReg (Hex.)	Transfer speed accuracy (%)
7.2	FA	-0.25
9.6	EB	0.32

Transfer speed (kbit/s)	Serial SpeedReg (Hex.)	Transfer speed accuracy (%)
14.4	DA	-0.25
19.2	CB	0.32
38.4	AB	0.32
57.6	9A	-0.25
115.2	7A	-0.25
128	74	-0.06
230.4	5A	-0.25
460.8	3A	-0.25
921.6	1C	1.45
1228.8	15	0.32

The selectable transfer speeds as shown are calculated according to the following formulas:

if BR_T0 = 0: transfer speed = 27.12 MHz / (BR_T1 + 1)
if BR_T0 > 0: transfer speed = 27.12 MHz / (BR_T1 + 33)/$2^{(BR\_T0 - 1)}$

Remark: Transfer speeds above 1228.8 kBits/s are not supported.

7.4.3.2 Framing

Table 14. UART framing

Bit	Length	Value
Start bit (Sa)	1 bit	0
Data bits	8 bit	Data
Stop bit (So)	1 bit	1

Remark: For data and address bytes the LSB bit has to be sent first. No parity bit is used during transmission.

Read data: To read out data using the UART interface the flow described below has to be used. The first send byte defines both the mode itself and the address.The Trigger on pin IF3 has to be set, otherwise no read of data is possible.

Table 15. Byte Order to Read Data

Mode	byte 0	byte 1
RX	address	-
TX	-	data 0

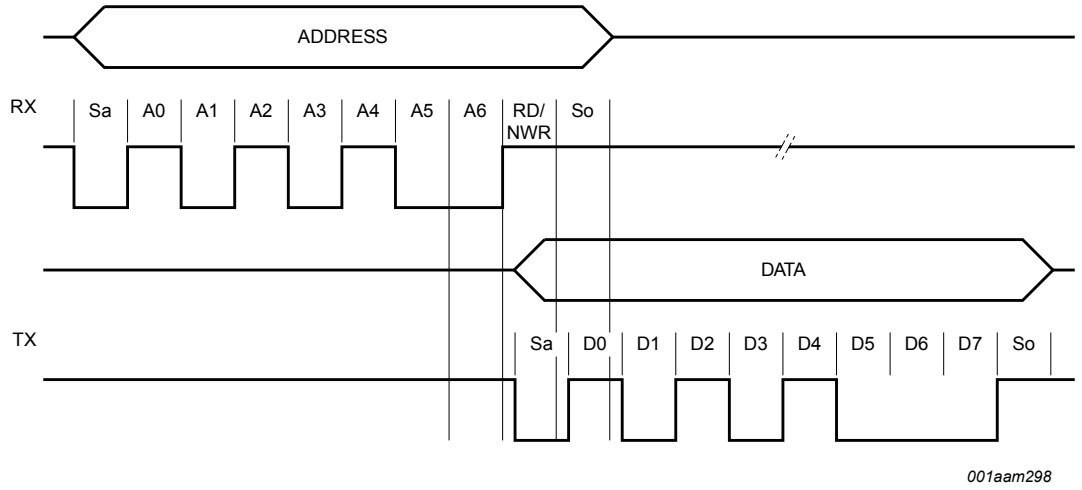


Figure 9. Example for UART Read

Write data:

To write data to the MFRC630 using the UART interface the following sequence has to be used.

The first send byte defines both, the mode itself and the address.

Table 16. Byte Order to Write Data

Mode	byte 0	byte 1
RX	address 0	data 0
TX		address 0

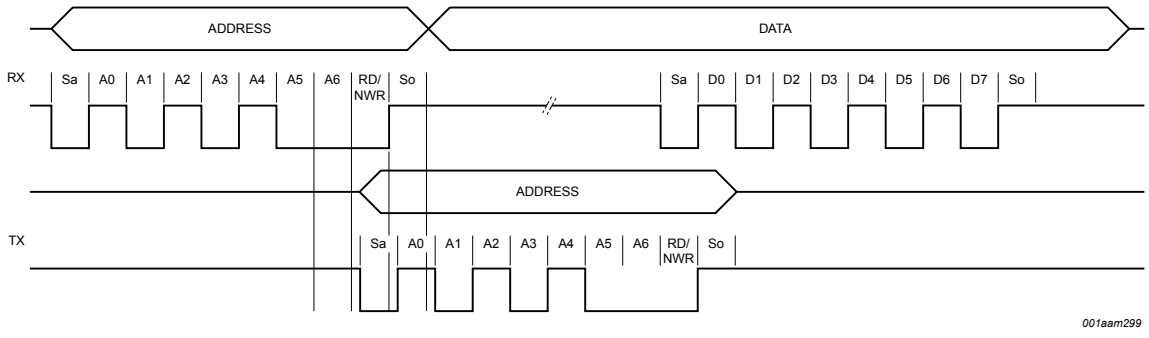


Figure 10. Example diagram for a UART write

Remark: Data can be sent before address is received.

7.4.4 I²C-bus interface

7.4.4.1 General

An Inter IC (I^2C) bus interface is supported to enable a low cost, low pin count serial bus interface to the host. The implemented I^2C interface is mainly implemented according the

NXP Semiconductors I²C interface specification, rev. 3.0, June 2007. The MFRC630 can act as a slave receiver or slave transmitter in standard mode, fast mode and fast mode plus.

The following features defined by the NXP Semiconductors I²C interface specification, rev. 3.0, June 2007 are not supported:

- The MFRC630 I2C interface does not stretch the clock
- The MFRC630 I2C interface does not support the general call. This means that the MFRC630 does not support a software reset
- The MFRC630 does not support the I2C device ID
- The implemented interface can only act in slave mode. Therefore no clock generation and access arbitration is implemented in the MFRC630.
- High speed mode is not supported by the MFRC630

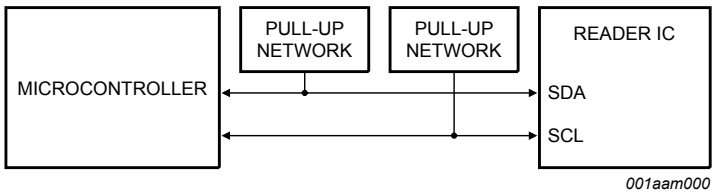


Figure 11. I²C-bus interface

The voltage level on the I2C pins is not allowed to be higher than PVDD.

SDA is a bidirectional line, connected to a positive supply voltage via a pull-up resistor. Both lines SDA and SCL are set to HIGH level if no data is transmitted. Data on the I²C-bus can be transferred at data rates of up to 400 kbit/s in fast mode, up to 1 Mbit/s in the fast mode+.

If the I²C interface is selected, a spike suppression according to the I²C interface specification on SCL and SDA is automatically activated.

For timing requirements refer to Table 196 "I²C-bus timing in fast mode and fast mode plus"

7.4.4.2 I²C Data validity

Data on the SDA line shall be stable during the HIGH period of the clock. The HIGH state or LOW state of the data line shall only change when the clock signal on SCL is LOW.

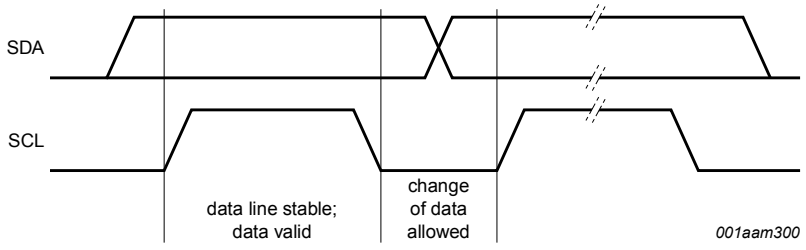


Figure 12. Bit transfer on the I²C-bus.

7.4.4.3 I²C START and STOP conditions

To handle the data transfer on the I²C-bus, unique START (S) and STOP (P) conditions are defined.

A START condition is defined with a HIGH-to-LOW transition on the SDA line while SCL is HIGH.

A STOP condition is defined with a LOW-to-HIGH transition on the SDA line while SCL is HIGH.

The master always generates the START and STOP conditions. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after the STOP condition.

The bus stays busy if a repeated START (Sr) is generated instead of a STOP condition. In this respect, the START (S) and repeated START (Sr) conditions are functionally identical. Therefore, the S symbol will be used as a generic term to represent both the START and repeated START (Sr) conditions.

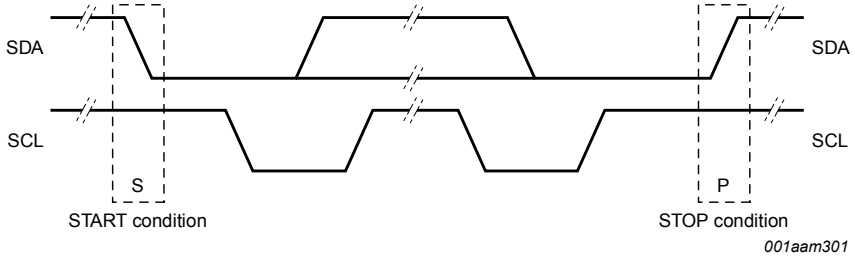


Figure 13. START and STOP conditions

7.4.4.4 I²C byte format

Each byte has to be followed by an acknowledge bit. Data is transferred with the MSB first, see Figure 13 "START and STOP conditions". The number of transmitted bytes during one data transfer is unrestricted but shall fulfil the read/write cycle format.

7.4.4.5 I²C Acknowledge

An acknowledge at the end of one data byte is mandatory. The acknowledge-related clock pulse is generated by the master. The transmitter of data, either master or slave, releases the SDA line (HIGH) during the acknowledge clock pulse. The receiver shall pull down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse.

The master can then generate either a STOP (P) condition to stop the transfer, or a repeated START (Sr) condition to start a new transfer.

A master-receiver shall indicate the end of data to the slave- transmitter by not generating an acknowledge on the last byte that was clocked out by the slave. The slave-transmitter shall release the data line to allow the master to generate a STOP (P) or repeated START (Sr) condition.

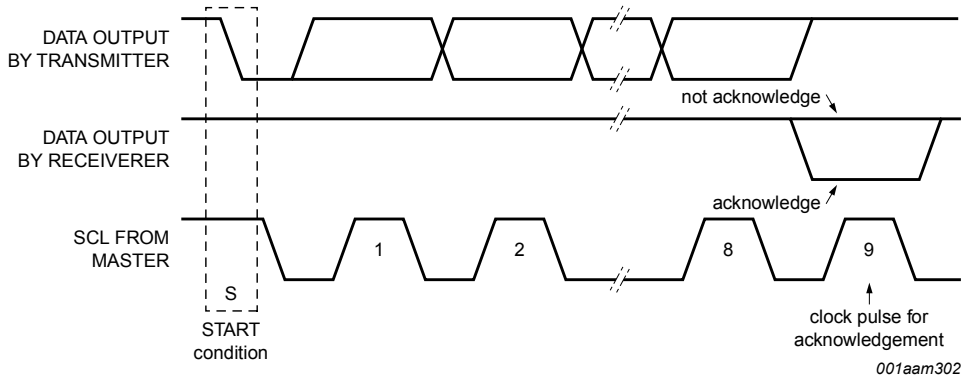


Figure 14. Acknowledge on the I²C- bus

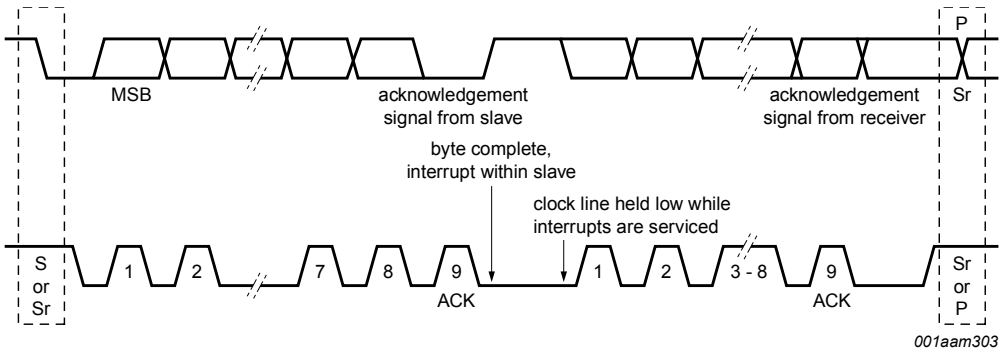


Figure 15. Data transfer on the I²C- bus

7.4.4.6 I²C 7-bit addressing

During the I²C-bus addressing procedure, the first byte after the START condition is used to determine which slave will be selected by the master.

Alternatively the I²C address can be configured in the EEPROM. Several address numbers are reserved for this purpose. During device configuration, the designer has to ensure, that no collision with these reserved addresses in the system is possible. Check the corresponding I²C specification for a complete list of reserved addresses.

For all MFRC630 devices the upper 5 bits of the device bus address are reserved by NXP and set to 01010(bin). The remaining 2 bits (ADR_2, ADR_1) of the slave address can be freely configured by the customer in order to prevent collisions with other I²C devices by using the interface pins (refer to Table 6) or the value of the I²C address EEPROM register (refer to Table 28).

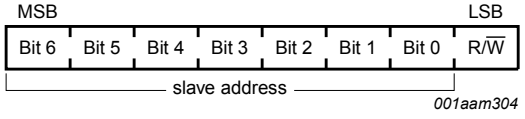


Figure 16. First byte following the START procedure

7.4.4.7 I²C-register write access

To write data from the host controller via I²C to a specific register of the MFRC630 the following frame format shall be used.

The read/write bit shall be set to logic 0.

The first byte of a frame indicates the device address according to the I²C rules. The second byte indicates the register address followed by up to n-data bytes. In case the address indicates the FIFO, in one frame all n-data bytes are written to the FIFO register address. This enables for example a fast FIFO access.

7.4.4.8 I²C-register read access

To read out data from a specific register address of the MFRC630 the host controller shall use the procedure:

First a write access to the specific register address has to be performed as indicated in the following frame:

The first byte of a frame indicates the device address according to the I²C rules. The second byte indicates the register address. No data bytes are added.

The read/write bit shall be logic 0.

Having performed this write access, the read access starts. The host sends the device address of the MFRC630. As an answer to this device address the MFRC630 responds with the content of the addressed register. In one frame n-data bytes could be read using the same register address. The address pointing to the register is incremented automatically (exception: FIFO register address is not incremented automatically). This enables a fast transfer of register content. The address pointer is incremented automatically and data is read from the locations [address], [address+1], [address+2]... [address+(n-1)]

In order to support a fast FIFO data transfer, the address pointer is not incremented automatically in case the address is pointing to the FIFO.

The read/write bit shall be set to logic 1.

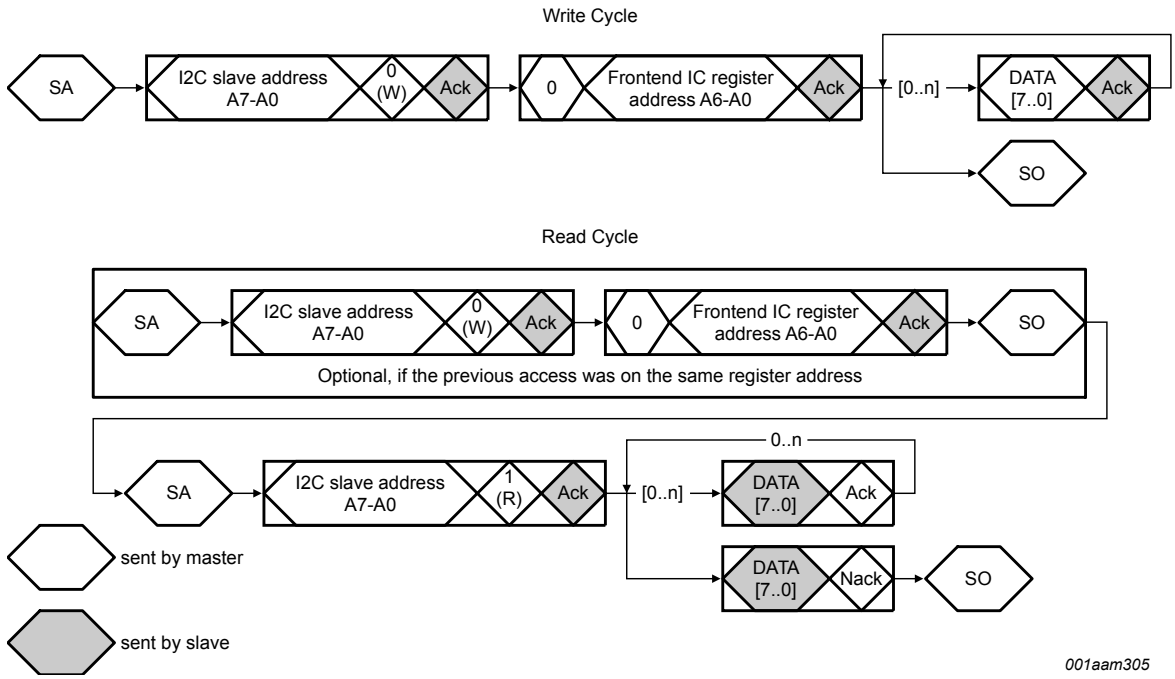


Figure 17. Register read and write access

7.4.4.9 I²CL-bus interface

The MFRC630 provides an additional interface option for connection of a SAM. This logical interface fulfills the I²C specification, but the rise/fall timings will not be compliant to the I²C standard. The I²CL interface uses standard I/O pads, and the communication speed is limited to 5 MBaud. The protocol itself is equivalent to the fast mode protocol of I²C. The SCL levels are generated by the host in push/pull mode. The MFRC630 does not stretch the clock. During the high period of SCL the status of the line is maintained by a bus keeper.

The address is 01010xxb, where the last two bits of the address can be defined by the application. The definition of this bits can be done by two options. With a pin, where the higher bit is fixed to 0 or the configuration can be defined via EEPROM. Refer to the EEPROM configuration in Section 7.7.

Table 17. Timing parameter I²CL

Parameter	Min	Max	Unit
f_{SCL}	0	5	MHz
$t_{HD;STA}$	80	-	ns
t_{LOW}	100	-	ns
t_{HIGH}	100	-	ns
$t_{SU;SDA}$	80	-	ns
$t_{HD;DAT}$	0	50	ns
$t_{SU;DAT}$	0	20	ns
$t_{SU;STO}$	80	-	ns
t_{BUF}	200	-	ns

The pull-up resistor is not required for the I^2CL interface. Instead, a on chip buskeeper is implemented in the MFRC630 for SDA of the I^2CL interface. This protocol is intended to be used for a point to point connection of devices over a short distance and does not support a bus capability.The driver of the pin must force the line to the desired logic voltage. To avoid that two drivers are pushing the line at the same time following regulations must be fulfilled:

SCL: As there is no clock stretching, the SCL is always under control of the Master.

SDA: The SDA line is shared between master and slave. Therefore the master and the slave must have the control over the own driver enable line of the SDA pin. The following rules must be followed:

- In the idle phase the SDA line is driven high by the master
- In the time between start and stop condition the SDA line is driven by master or slave when SCL is low. If SCL is high the SDA line is not driven by any device
- To keep the value on the SDA line a on chip buskeeper structure is implemented for the line

7.4.5 SAM interface

7.4.5.1 SAM functionality

The MFRC630 implements a dedicated I2C or SPI interface to integrate a MIFARE SAM (Secure Access Module) in a very convenient way into applications (e.g. a proximity reader).

The SAM can be connected to the microcontroller to operate like a cryptographic co-processor. For any cryptographic task, the microcontroller requests a operation from the SAM, receives the answer and sends it over a host interface (e.g. I2C, SPI) interface to the connected reader IC.

The MIFARE SAM supports a optimized method to integrate the SAM in a very efficient way to reduce the protocol overhead. In this system configuration, the SAM is integrated between the microprocessor and the reader IC, connected by one interface to the reader IC and by another interface to the microcontroller. In this application the microcontroller accesses the SAM using the T=1 protocol and the SAM accesses the reader IC using an I2C interface. The I2C SAM address is always defined by EEPROM register. Default value is 0101100. As the SAM is directly communicating with reader IC, the communication overhead is reduced. In this configuration, a performance boost of up to 40% can be achieved for a transaction time.

The MIFARE SAM supports applications using MIFARE product-based cards. For multi application purposes an architecture connecting the microcontroller additionally directly to the reader IC is recommended. This is possible by connecting the MFRC630 on one interface (SAM Interface SDA, SCL) with the MIFARE SAM AV2.6 (P5DF081XX/T1AR1070) and by connecting the microcontroller to the S2C or SPI interface.

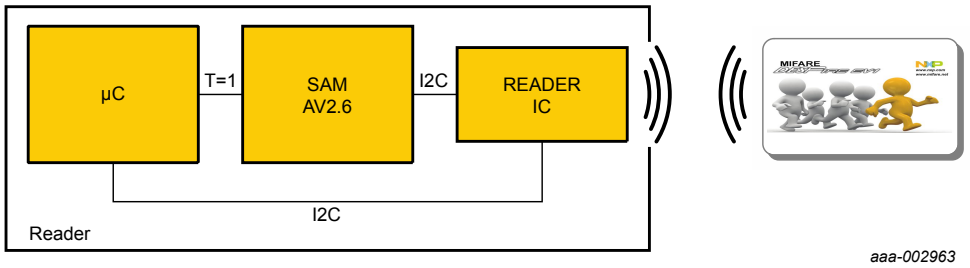


Figure 18. I2C interface enables convenient MIFARE SAM integration

7.4.5.2 SAM connection

The MFRC630 provides an interface to connect a SAM dedicated to the MFRC630. Both interface options of the MFRC630, I^2C, I^2CL or SPI can be used for this purpose. The interface option of the SAM itself is configured by a host command sent from the host to the SAM.

The I^2CL interface is intended to be used as connection between two IC's over a short distance. The protocol fulfills the I^2C specification, but does support a single device connected to the bus only.

The SPI block for SAM connection is identical with the SPI host interface block.

The pins used for the SAM SPI are described in Table 17.

Table 18. SPI SAM connection

SPI functionality	PIN
MISO	SDA2
SCL	SCL2
MOSI	IFSEL1
NSS	IFSEL0

7.4.6 Boundary scan interface

The MFRC630 provides a boundary scan interface according to the IEEE 1149.1. This interface allows to test interconnections without using physical test probes. This is done by test cells, assigned to each pin, which override the functionality of this pin.

To be able to program the test cells, the following commands are supported:

Table 19. Boundary scan command

Value (decimal)	Command	Parameter in	Parameter out
0	bypass	-	-
1	preload	data (24)	-
1	sample	-	data (24)
2	ID code (default)	-	data (32)
3	USER code	-	data (32)
4	Clamp	-	-