DATA SHEET

mifare®

MF RC530

ISO14443A Reader IC

Product Specification

Revision 3.2

PUBLIC

December 2005

Philips Semiconductors

PHILIPS

CONTENTS

MIFARE® is a registered trademark of Philips Electronics N.V.

1 GENERAL INFORMATION

1.1 Scope

This document describes the functionality of the MF RC530. It includes the functional and electrical specifications and gives details on how to design-in this device from system and hardware viewpoint.

1.2 General Description

The MF RC530 is member of a new family of highly integrated reader ICs for contactless communication at 13.56 MHz. This reader IC family utilises an outstanding modulation and demodulation concept completely integrated for all kinds of passive contactless communication methods and protocols at 13.56 MHz. The MF RC530 is pin- compatible to the MF RC500, the MF RC531, SL RC400 and CL RC632.

The MF RC530 supports all layers of the ISO14443A communication scheme.

The MF RC530 supports contactless communication using MIFARE® Higher Baudrates.

The internal transmitter part is able to drive an antenna designed for proximity operating distance (up to 100 mm) directly without additional active circuitry.

The receiver part provides a robust and efficient implementation of a demodulation and decoding circuitry for signals from ISO14443A compatible transponders.

The digital part handles the complete ISO14443A framing and error detection (Parity & CRC).
Additionally it supports the fast MIFARE® Classic security algorithm to authenticate MIFARE® Classic (e.g. MIFARE® Standard, MIFARE® Light) products.

A comfortable parallel interface, which can be directly connected to any 8-bit µ-Processor gives high flexibility for the reader/terminal design.
Additionally a SPI compatible interface is supported.

1.3 Features

- Highly integrated analog circuitry to demodulate and decode card response
- Buffered output drivers to connect an antenna with minimum number of external components
- Proximity operating distance (up to 100 mm)
- Supports ISO 14443A
- Supports MIFARE® Dual Interface Card ICs and supports MIFARE® Classic protocol
- Supports contactless communication with higher baudrates up to 424kbaud
- Crypto1 and secure non-volatile internal key memory
- Pin-compatible to the MF RC500, MF RC531, SL RC400 and CL RC632
- Parallel µ-Processor interface with internal address latch and IRQ line
- SPI compatible interface
- Flexible interrupt handling
- Automatic detection of parallel µ-Processor interface type
- Comfortable 64 byte send and receive FIFO-buffer
- Hard reset with low power function
- Power down mode per software
- Programmable timer
- Unique serial number
- User programmable start-up configuration
- Bit- and byte-oriented framing
- Independent power supply pins for digital, analog and transmitter part
- Internal oscillator buffer to connect 13.56 MHz quartz, optimised for low phase jitter
- Clock frequency filtering
- 3.3 V to 5 V operation for transmitter (antenna driver) in short range and proximity applications
- 3.3 V or 5V operation for the digital part

1.4 Ordering Information

Type Number	Package	
	Name	Description
MF RC530 01T/0FE	SO32	Small Outline Package; 32 leads

Table 1-1: MF RC530 Ordering Information

2 BLOCK DIAGRAM

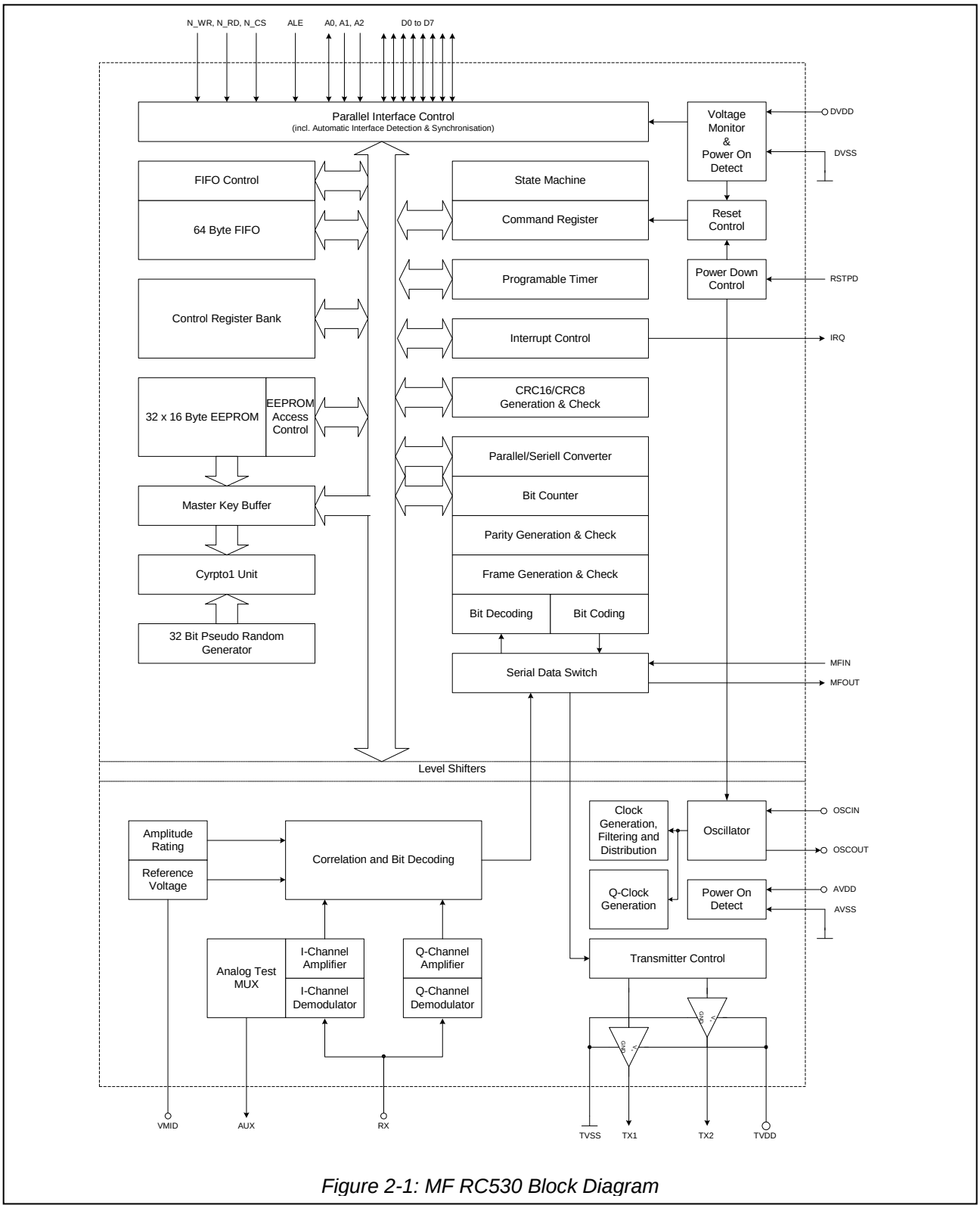


Figure 2-1: MF RC530 Block Diagram

3 PINNING INFORMATION

3.1 Pin Configuration

Pins denoted by bold letters are supplied by AVDD and AVSS. Pins drawn with bold lines are supplied by TVSS and TVDD. All other pins are supplied by DVDD and DVSS.

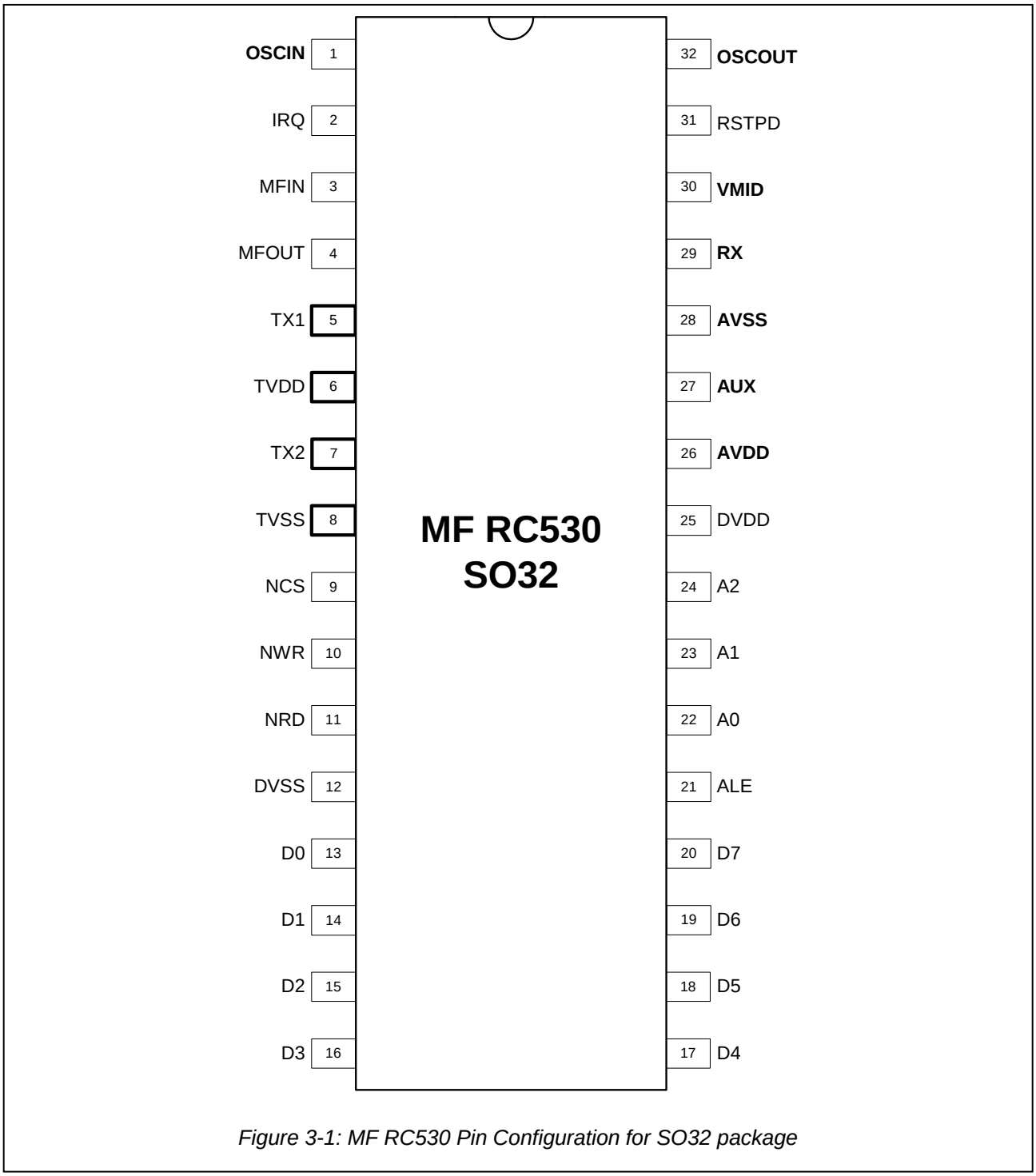


Figure 3-1: MF RC530 Pin Configuration for SO32 package

3.2 Pin Description

Pin Types: I...Input; O...Output; PWR...Power

PIN	SYMBOL	TYPE	DESCRIPTION
1	OSCIN	I	**Crystal Oscillator Input**: input to the inverting amplifier of the oscillator. This pin is also the input for an externally generated clock (f_{osc} = 13.56 MHz).
2	IRQ	O	**Interrupt Request**: output to signal an interrupt event
3	MFIN	I	**MIFARE® Interface Input**: accepts a digital, serial data stream according to ISO14443A (MIFARE®)
4	MFOUT	O	**MIFARE® Interface Output**: delivers a serial data stream according to ISO14443A (MIFARE®)
5	TX1	O	**Transmitter 1**: delivers the modulated 13.56 MHz energy carrier
6	TVDD	PWR	**Transmitter Power Supply**: supplies the output stage of TX1 and TX2
7	TX2	O	**Transmitter 2**: delivers the modulated 13.56 MHz energy carrier
8	TVSS	PWR	**Transmitter Ground**: supplies the output stage of TX1 and TX2
9	NCS	I	**Not Chip Select**: selects and activates the µ-Processor interface of the MF RC530
10[1]	NWR	I	**Not Write**: strobe to write data (applied on D0 to D7) into the MF RC530 register
	R/NW	I	**Read Not Write**: selects if a read or write cycle shall be performed.
	nWrite	I	**Not Write**: selects if a read or write cycle shall be performed
11[1]	NRD	I	**Not Read**: strobe to read data from the MF RC530 register (applied on D0 to D7)
	NDS	I	**Not Data Strobe**: strobe for the read and the write cycle
	nDStrb	I	**Not Data Strobe**: strobe for the read and the write cycle
12	DVSS	PWR	**Digital Ground**
13	D0	O	**Master In Slave Out (MISO),** SPI compatible interface
13	D0 to D7	I/O	**8 Bit Bi-directional Data Bus**
… 20[1]	AD0 to AD7	I/O	**8 Bit Bi-directional Address and Data Bus**
21[1]	ALE	I	**Address Latch Enable**: signal to latch AD0 to AD5 into the internal address latch when HIGH.
	AS	I	**Address Strobe**: strobe signal to latch AD0 to AD5 into the internal address latch when HIGH.
	nAStrb	I	**Not Address Strobe**: strobe signal to latch AD0 to AD5 into the internal address latch when LOW.
	NSS	I	**Not Slave Select:** strobe for the SPI communication
22[1]	A0	I	**Address Line 0**: Bit 0 of register address
	nWait	O	**Not Wait:** signals with LOW that an access-cycle may started and with HIGH that it may be finished.
	MOSI	I	**Master Out Slave In,** SPI compatible interface
23	A1	I	**Address Line 1**: Bit 1 of register address
24[1]	A2	I	**Address Line 2**: Bit 2 of register address
	SCK	I	**Serial Clock**: Clock for the SPI compatible interface
25	DVDD	PWR	**Digital Power Supply**
26	AVDD	PWR	**Analog Power Supply**

[1] These pins offer different functionality according to the selected µ-Processor interface type. For detailed information, refer to chapter 4.

PIN Description (continued)

PIN	SYMBOL	TYPE	DESCRIPTION
27	AUX	O	**Auxiliary Output**: This pin delivers analog test signals. The signal delivered on this output may be selected by means of the *TestAnaOutSel Register*.
28	AVSS	PWR	**Analog Ground**
29	RX	I	**Receiver Input**: Input pin for the cards response, which is the load modulated 13.56 MHz energy carrier, that is coupled out from the antenna circuit.
30	VMID	PWR	**Internal Reference Voltage**: This pin delivers the internal reference voltage. Note: It has to be supported by means of a 100 nF block capacitor.
31	RSTPD	I	**Reset and Power Down**: When HIGH, internal current sinks are switched off, the oscillator is inhibited, and the input pads are disconnected from the outside world. With a negative edge on this pin the internal reset phase starts.
32	OSCOUT	O	**Crystal Oscillator Output**: Output of the inverting amplifier of the oscillator.

Table 3-1: MF RC530 Pin Description

4 DIGITAL INTERFACE

4.1 Overview of Supported µ-Processor Interfaces

The MF RC530 supports direct interfacing of various µ-Processors. Alternatively the Enhanced Parallel Port (EPP) of personal computers can be connected directly. The following table shows the parallel interface signals supported by the MF RC530:

Bus Control Signals	Bus	Separated Address and Data Bus	Multiplexed Address and Data Bus
Separated Read and Write Strobes	**control**	NRD, NWR, NCS	NRD, NWR, NCS, ALE
	address	A0, A1, A2	AD0, AD1, AD2, AD3, AD4, AD5
	data	D0 … D7	AD0 … AD7
Common Read and Write Strobe	**control**	R/NW, NDS, NCS	R/NW, NDS, NCS, AS
	address	A0, A1, A2	AD0, AD1, AD2, AD3, AD4, AD5
	data	D0 … D7	AD0 … AD7
Common Read and Write Strobe with Handshake (EPP)	**control**	-	nWrite, nDStrb, nAStrb, nWait
	address		AD0, AD1, AD2, AD3, AD4, AD5
	data		AD0 … AD7

Table 4-1: Supported µ-Processor Interface Signals

4.2 Automatic µ-Processor Interface Type Detection

After every Power-On or Hard Reset, the MF RC530 also resets its parallel µ-Processor interface mode and checks the current µ-Processor interface type.
The MF RC530 identifies the µ-Processor interface by means of the logic levels on the control pins after the Reset Phase. This is done by a combination of fixed pin connections (see below) and a dedicated initialisation routine (see 11.4).

4.3 Connection to Different µ-Processor Types

The connection to different µ-Processor types is shown in the following table:

MF RC530	Parallel Interface Type				
	Separated Read/Write Strobe		Common Read/Write Strobe		
	Dedicated Address Bus	Multiplexed Address Bus	Dedicated Address Bus	Multiplexed Address Bus	Multiplexed Address Bus with Handshake
ALE	HIGH	ALE	HIGH	AS	nAStrb
A2	A2	LOW	A2	LOW	HIGH
A1	A1	HIGH	A1	HIGH	HIGH
A0	A0	HIGH	A0	LOW	nWait
NRD	NRD	NRD	NDS	NDS	nDStrb
NWR	NWR	NWR	R/NW	R/NW	nWrite
NCS	NCS	NCS	NCS	NCS	LOW
D7 ... D0	D7 ... D0	AD7 ... AD0	D7 ... D0	AD7 ... AD0	AD7 ... AD0

Table 4-2: Connection Scheme for Detecting the Parallel Interface Type

4.3.1 SEPARATED READ/WRITE STROBE

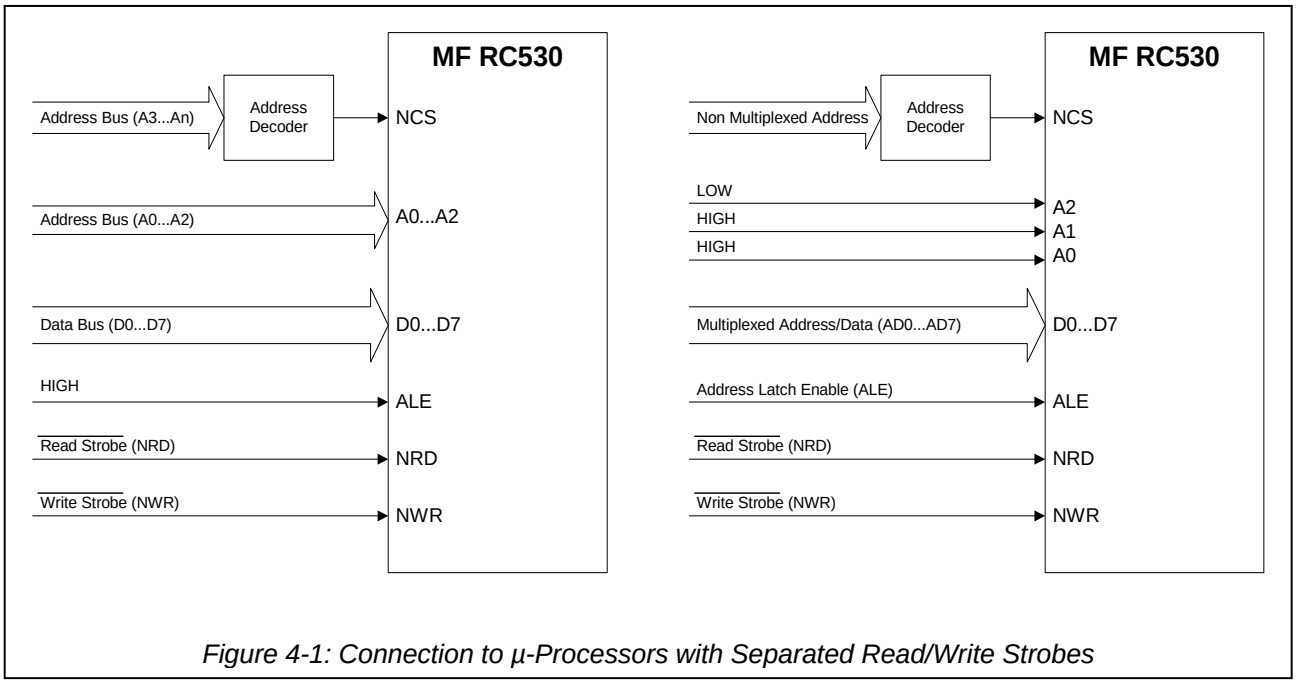


Figure 4-1: Connection to µ-Processors with Separated Read/Write Strobes

For timing specification refer to chapter 21.5.2.1.

4.3.2 COMMON READ/WRITE STROBE

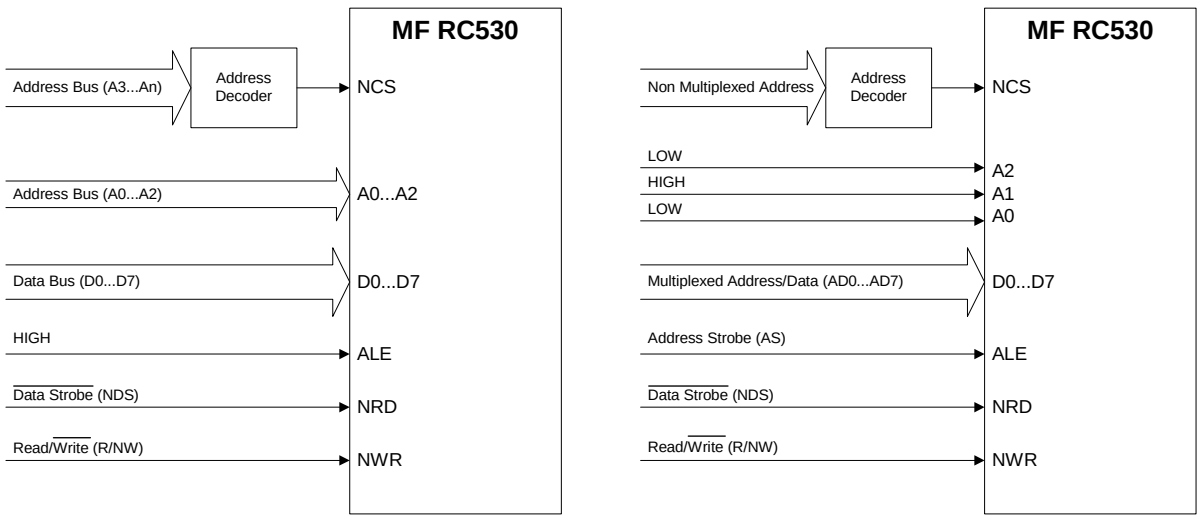


Figure 4-2: Connection to µ-Processors with Common Read/Write Strobes

For timing specification refer to chapter 21.5.2.2.

4.3.3 COMMON READ/WRITE STROBE AND HAND-SHAKE MECHANISM: EPP

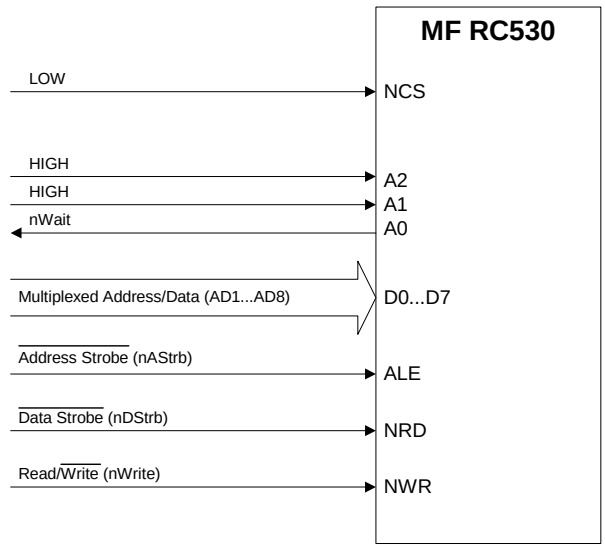


Figure 4-3: Connection to µ-Processors with Common Read/Write Strobes and Hand-Shake

For timing specification refer to chapter 21.5.2.3.

Remarks for EPP:
Although in the standard for the EPP no chip select signal is defined, the N_CS of the MF RC530 allows inhibiting the nDStrb signal. If not used, it shall be connected to DVSS.

After each Power-On or Hard Reset the nWait signal (delivered at pin A0) is high impedance. nWait will be defined at the first negative edge applied to nAStrb after the Reset Phase.

The MF RC530 does not support Read Address Cycle.

4.4 SPI Compatible interface

Additionally a serial peripheral interface (SPI) will be supported. The MF RC530 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 MF RC530 to the master.

MF RC530	SPI Interface
ALE	NSS
A2	SCK
A1	LOW
A0	MOSI
NRD	HIGH
NWR	HIGH
NCS	LOW
D7 ... D1	do not connect
D0	MISO

Table 4-3: SPI compatible interface

The following table shows the µ-Processor connection to the MF RC530 using the SPI compatible interface.

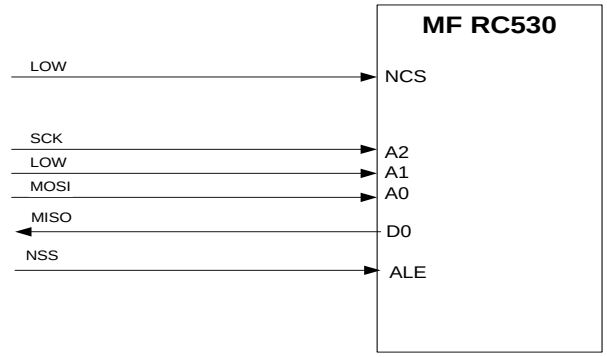


Figure 4-4: Connection to µ-Processors with SPI

Remarks for SPI:

The implemented SPI compatible interface is according to a standard SPI interface.
The MF RC530 can only be addressed as a slave.

Read data:

To read out data using the SPI compatible interface the following structure has to be used. It is possible to read out up to n-data bytes.
The first sent byte defines both, the mode itself and the address.

	byte 0	byte 1	byte 2	……..	byte n	byte n+1
MOSI	adr 0	adr 1	adr. 2	…….	adr n	00
MISO	XX	data 0	data 1	……	data n-1	data n

The address byte has to fulfil the following format. The MSB bit of the first byte sets the used mode. To read data from the MF RC530 the MSB bit is set to 1. The bits 6-1 define the address and the last bit should be set to 0.
According to scheme above, the last sent byte has been set to 0.

address (MOSI)	bit 7, MSB	bit 6 - bit 1	bit 0
byte 0	1	address	RFU (0)
byte 1 to byte n	RFU (0)	address	RFU (0)
byte n+1	0	0	0

Write data:

To write data to the MF RC530 using the SPI interface the following structure has to be used. It is possible to write out up to n-data bytes.

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

	byte 0	byte 1	byte 2	…….	byte n	byte n+1
MOSI	adr	data 0	data 1	…….	data n-1	data n
MISO	XX	XX	XX	…….	XX	XX

The address byte has to fulfil the following format. The MSB bit of the first byte sets the used mode. To write data to the MF RC530 the MSB bit is set to 0. The bits 6-1 define the address and the last bit should be set to 0.

The SPI write mode writes all data to the same address as defined in byte 0.
This allows an effective data writing to the MF RC530's FIFO buffer.

Address line (MOSI)	MSB	bit 6 - bit 1	bit 0
byte 0	0	address	RFU (0)
byte 1 to byte n+1	data		

Note:

The data bus pins D7…D1 have to be disconnected.

For timing specification refer to chapter 21.5.2.4

5 MF RC530 REGISTER SET

5.1 MF RC530 Registers Overview

Page	$Address_{hex}$	Register Name	Function
Page 0: Command and Status	0	Page	selects the register page
	1	Command	starts (and stops) the command execution
	2	FIFOData	in- and output of 64 byte FIFO buffer
	3	PrimaryStatus	status flags of the receiver and transmitter and of the FIFO buffer
	4	FIFOLength	number of bytes buffered in the FIFO
	5	SecondaryStatus	diverse status flags
	6	InterruptEn	control bits to enable and disable passing of interrupt requests
	7	InterruptRq	interrupt request flags
Page 1: Control and Status	8	Page	selects the register page
	9	Control	diverse control flags e.g.: timer, power saving
	A	ErrorFlag	error flags showing the error status of the last command executed
	B	CollPos	bit position of the first bit collision detected on the RF-interface
	C	TimerValue	actual value of the timer
	D	CRCResultLSB	LSB of the CRC-Coprocessor register
	E	CRCResultMSB	MSB of the CRC-Coprocessor register
	F	BitFraming	adjustments for bit oriented frames
Page 2: Transmitter and Coder Control	10	Page	selects the register page
	11	TxControl	controls the logical behaviour of the antenna driver pins TX1 and TX2
	12	CWConductance	selects the conductance of the antenna driver pins TX1 and TX2
	13	PreSet13	these values shall not be changed!
	14	CoderControl	sets the clock rate and the coding mode
	15	ModWidth	selects the width of the modulation pulse
	16	PreSet16	these values shall not be changed!
	17	PreSet17	these values shall not be changed!
Page 3: Receiver and Decoder Control	18	Page	selects the register page
	19	RxControl1	controls receiver behaviour
	1A	DecoderControl	controls decoder behaviour
	1B	BitPhase	selects the bit-phase between transmitter and receiver clock
	1C	RxThreshold	selects thresholds for the bit decoder
	1D	BPSKDemControl	Control BPSK receiver behaviour
	1E	RxControl2	controls decoder behaviour and defines the input source for the receiver
	1F	ClockQControl	controls clock generation for the 90° phase shifted Q-channel clock

MF RC530 Register Set (continued)

Page	Address$_{hex}$	Register Name	Function
Page 4: RF-Timing and Channel Redundancy	20	Page	selects the register page
	21	RxWait	selects the time interval after transmission, before receiver starts
	22	ChannelRedundancy	selects the kind and mode of checking the data integrity on the RF-channel
	23	CRCPresetLSB	LSB of the pre-set value for the CRC register
	24	CRCPresetMSB	MSB of the pre-set value for the CRC register
	25	PreSet25	these values shall not be changed
	26	MFOUTSelect	selects internal signal applied to pin MFOUT
	27	PreSet27	these values shall not be changed
Page 5: FIFO, Timer and IRQ-Pin Configuration	28	Page	selects the register page
	29	FIFOLevel	defines level for FIFO over– and underflow warning
	2A	TimerClock	selects the divider for the timer clock
	2B	TimerControl	selects start and stop conditions for the timer
	2C	TimerReload	defines the pre-set value for the timer
	2D	IRQPinConfig	configures the output stage of pin IRQ
	2E	PreSet2E	these values shall not be changed
	2F	PreSet2F	these values shall not be changed
Page 6: RFU	30	Page	selects the register page
	31	RFU	reserved for future use
	32	RFU	reserved for future use
	33	RFU	reserved for future use
	34	RFU	reserved for future use
	35	RFU	reserved for future use
	36	RFU	reserved for future use
	37	RFU	reserved for future use
Page 7: Test Control	38	Page	selects the register page
	39	RFU	reserved for future use
	3A	TestAnaSelect	selects analog test mode
	3B	RFU	reserved for future use
	3C	RFU	reserved for future use
	3D	TestDigiSelect	selects digital test mode
	3E	RFU	reserved for future use
	3F	RFU	reserved for future use

Table 5-1: MF RC530 Register Overview

5.1.1 REGISTER BIT BEHAVIOUR

Bits and flags for different registers behave differently, depending on their functions. In principle bits with same behaviour are grouped in common registers.

Abbreviation	Behaviour	Description
r/w	read and write	These bits can be written and read by the µ-Processor. Since they are used only for control means, there content is not influenced by internal state machines, e.g. the *TimerReload-Register* may be written and read by the µ-Processor. It will also be read by internal state machines, but never changed by them.
dy	dynamic	These bits can be written and read by the µ-Processor. Nevertheless, they may also be written automatically by internal state machines, e.g. the *Command-Register* changes its value automatically after the execution of the actual command.
r	read only	These registers hold flags, which value is determined by internal states only, e.g. the *ErrorFlag-Register* can not be written from external but shows internal states.
w	write only	These registers are used for control means only. They may be written by the µ-Processor but can not be read. Reading these registers returns an undefined value, e.g. the *TestAnaSelect-Register* is used to determine the signal on pin AUX, but it is not possible to read its content.

Table 5-2: Behaviour of Register Bits and its Designation

5.2 Register Description

5.2.1 PAGE 0: COMMAND AND STATUS

5.2.1.1 Page Register

Selects the register page.

Name: Page Address: 0x00, 0x08, 0x10, 0x18, 0x20, 0x28, 0x30, 0x38 Reset value: 10000000, 0x80

	7	6	5	4	3	2	1	0
	UsePage Select	0	0	0	0	PageSelect		
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7	UsePageSelect	If set to 1, the value of *PageSelect* is used as register address A5, A4, and A3. The LSBbits of the register address are defined by the address pins or the internal address latch, respectively. If set to 0, the whole content of the internal address latch defines the register address. The address pins are used as described in Table 4-2.
6-3	0000	Reserved for future use.
2-0	PageSelect	The value of *PageSelect* is used only if *UsePageSelect* is set to 1. In this case, it specifies the register page (which is A5, A4, and A3 of the register address).

5.2.1.2 Command Register

Starts and stops the command execution.

Name: Command Address: 0x01 Reset value:X0000000, 0xX0

	7	6	5	4	3	2	1	0
	IFDetect Busy	0	Command					
Access Rights	r	r	dy	dy	dy	dy	dy	dy

Description of the bits

Bit	Symbol	Function
7	IFDetectBusy	Shows the status of Interface Detection Logic: Set to 0 means ‘Interface Detection finished successfully’, Set to 1 signs ‘Interface Detection Ongoing’.
6	0	Reserved for future use.
5-0	Command	Activates a command according the Command Code. Reading this register shows, which command is actually executed.

5.2.1.3 FIFOData Register

In- and output of the 64 byte FIFO buffer.

Name: FIFOData Address: 0x02 Reset value: XXXXXXXX, 0xXX

	7	6	5	4	3	2	1	0
	FIFOData							
Access Rights	dy	dy	dy	dy	dy	dy	dy	dy

Description of the bits

Bit	Symbol	Function
7-0	FIFOData	Data input and output port for the internal 64 byte FIFO buffer. The FIFO buffer acts as parallel in/parallel out converter for all data stream in- and outputs.

5.2.1.4 PrimaryStatus Register

Status flags of the receiver, transmitter and the FIFO buffer.

Name: PrimaryStatus Address: 0x03 Reset value: 00000101, 0x05

	7	6	5	4	3	2	1	0
	0	ModemState			IRq	Err	HiAlert	LoAlert
Access Rights	r	r	r	r	r	r	r	r

Description of the bits

Bit	Symbol	Function
7	0	Reserved for future use.
6-4	ModemState	ModemState shows the state of the transmitter and receiver state machines. **State** / **Name of State** / **Description** 000 / Idle / Neither the transmitter nor the receiver is in operation, since none of them is started or since none of them has input data. 001 / TxSOF / Transmitting the 'Start Of Frame' Pattern. 010 / TxData / Transmitting data from the FIFO buffer (or redundancy check bits). 011 / TxEOF / Transmitting the 'End Of Frame' Pattern. 100 / GoToRx1 / Intermediate state, when receiver starts. GoToRx2 / Intermediate state, when receiver finishes. 101 / PrepareRx / Waiting until the time period selected in the *RxWait Register* is expired. 110 / AwaitingRx / Receiver activated; Awaiting an input signal at pin Rx. 111 / Receiving / Receiving data.
3	IRQ	This bit shows, if any interrupt source requests attention (with respect to the setting of the interrupt enable flags in the *InterruptEn Register*).
2	Err	This bit is set to 1, if any error flag in the *ErrorFlag Register* is set.
1	HiAlert	Is set to 1, when the number of bytes stored in the FIFO buffer fulfil the following equation: $HiAlert = (64 - FIFOLength) \leq WaterLevel$ Example: FIFOLength=60, WaterLevel=4 ⇒ HiAlert =1 FIFOLength=59, WaterLevel=4 ⇒ HiAlert =0
0	LoAlert	Is set to 1, when the number of bytes stored in the FIFO buffer fulfil the following equation: $LoAlert = FIFOLength \leq WaterLevel$ Example: FIFOLength=4, WaterLevel=4 ⇒ LoAlert =1 FIFOLength=5, WaterLevel=4 ⇒ LoAlert =0

5.2.1.5 FIFOLength Register

Number of bytes buffered in the FIFO.

Name: FIFOLength Address: 0x04 Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
	0	FIFOLength						
Access Rights	r	r	r	r	r	r	r	r

Description of the bits

Bit	Symbol	Function
7	0	Reserved for future use.
6-0	FIFOLength	Indicates the number of bytes stored in the FIFO buffer. Writing to the *FIFOData* Register increments, reading decrements *FIFOLength.*

5.2.1.6 SecondaryStatus Register

Diverse Status flags.

Name: SecondaryStatus Address: 0x05 Reset value: 01100000, 0x60

	7	6	5	4	3	2	1	0
	TRunning	E2Ready	CRCReady	0	0	RxLastBits		
Access Rights	r	r	r	r	r	r	r	r

Description of the bits

Bit	Symbol	Function
7	TRunning	If set to 1, the MF RC530's timer unit is running, e.g. the counter will decrement the *Timer Value Register* with the next timer clock.
6	E2Ready	If set to 1, the MF RC530 has finished programming the E2PROM.
5	CRCReady	If set to 1, the MF RC530 has finished calculating the CRC.
4-3	00	Reserved for future use.
2-0	RxLastBits	Show the number of valid bits in the last received byte. If zero, the whole byte is valid.

5.2.1.7 InterruptEn Register

Control bits to enable and disable passing of interrupt requests.

Name: InterruptEn Address: 0x06 Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
	SetIEn	0	TimerIEn	TxIEn	RxIEn	IdleIEn	HiAlertIEn	LoAlertIEn
Access Rights	w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7	SetIEn	Set to 1 *SetIEn* defines that the marked bits in the *InterruptEn Register* are set, Set to 0 clears the marked bits.
6	0	Reserved for future use.
5	TimerIEn	Allows the timer interrupt request (indicated by bit *TimerIRq*) to be propagated to pin IRQ. This bit can not be set or cleared directly but only by means of bit *SetIEn*.
4	TxIEn	Allows the transmitter interrupt request (indicated by bit *TxIRq*) to be propagated to pin IRQ. This bit can not be set or cleared directly but only by means of bit *SetIEn*.
3	RxIEn	Allows the receiver interrupt request (indicated by bit *RxIRq*) to be propagated to pin IRQ. This bit can not be set or cleared directly but only by means of bit *SetIEn*.
2	IdleIEn	Allows the idle interrupt request (indicated by bit *IdleIRq*) to be propagated to pin IRQ. This bit can not be set or cleared directly but only by means of bit *SetIEn*.
1	HiAlertIEn	Allows the high alert interrupt request (indicated by bit *HiAlertIRq*) to be propagated to pin IRQ. This bit can not be set or cleared directly but only by means of bit *SetIEn*.
0	LoAlertIEn	Allows the low alert interrupt request (indicated by bit *LoAlertIRq*) to be propagated to pin IRQ. This bit can not be set or cleared directly but only by means of bit *SetIEn*.

5.2.1.8 InterruptRq Register

Interrupt request flags.

Name: InterruptRq Address: 0x07 Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
	SetIRq	0	TimerIRq	TxIRq	RxIRq	IdleIRq	HiAlertIRq	LoAlertIRq
Access Rights	w	r/w	dy	dy	dy	dy	dy	dy

Description of the bits

Bit	Symbol	Function
7	SetIRq	Set to 1, *SetIRq* defines that the marked bits in the *InterruptRq Register* are set. Set to 0 *SetIRq* defines, that the marked bits in the *InterruptRq Register* are cleared.
6	0	Reserved for future use.
5	TimerIRq	Set to 1, when the timer decrements the *TimerValue Register* to zero.
4	TxIRq	Set to 1, when one of the following events occurs: *Transceive Command*: All data transmitted. *Auth1* and *Auth2 Command*: All data transmitted. *WriteE2 Command*: All data is programmed. *CalcCRC Command*: All data is processed.
3	RxIRq	This bit is set to 1, when the receiver terminates.
2	IdleIRq	This bit is set to 1, when a command terminates by itself e.g. when the *Command Register* changes its value from any command to the *Idle Command*. If an unknown command is started bit *IdleIRq* is set. Starting the *Idle Command* by the µ-Processor does not set bit *IdleIRq*.
1	HiAlertIRq	This bit is set to 1, when bit *HiAlert* is set. In opposite to *HiAlert*, *HiAlertIRq* stores this event and can only be reset by means of bit *SetIRq*.
0	LoAlertIRq	This bit is set to 1, when bit *LoAlert* is set. In opposite to *LoAlert*, *LoAlertIRq* stores this event and can only be reset by means of bit *SetIRq*.

5.2.2 PAGE 1: CONTROL AND STATUS

5.2.2.1 Page Register

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.2.2 Control Register

Diverse control flags, e.g.: timer, power saving.

Name: Control Address: 0x09 Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
	0	0	StandBy	PowerDown	Crypto1On	TStopNow	TStartNow	FlushFIFO
Access Rights	r/w	r/w	dy	dy	dy	w	w	w

Description of the bits

Bit	Symbol	Function
7-6	00	Reserved for future use
5	StandBy	Setting this bit to 1 enters the Soft PowerDown Mode. This means, internal current consuming blocks are switched off, the oscillator keeps running.
4	PowerDown	Setting this bit to 1 enters the Soft PowerDown Mode. This means, internal current consuming blocks are switched off including the oscillator.
3	Crypto1On	This bit indicates that the Crypto1 unit is switched on and therefore all data communication with the card is encrypted. This bit can only be set to 1 by a successful execution of the *Authent2 Command.*
2	TStopNow	Setting this bit to 1 stops the timer immediately. Reading this bit will always return 0.
1	TStartNow	Setting this bit to 1 starts the timer immediately. Reading this bit will always return 0.
0	FlushFIFO	Setting this bit to 1clears the internal FIFO-buffer's read- and write-pointer (*FIFOLength* becomes 0) and the flag *FIFOOvfl* immediately. Reading this bit will always return 0.

5.2.2.3 ErrorFlag Register

Error flags showing the error status of the last executed command.

Name: ErrorFlag Address: 0x0A Reset value: 01000000, 0x40

	7	6	5	4	3	2	1	0
	0	KeyErr	AccessErr	FIFOOvfl	CRCErr	FramingErr	ParityErr	CollErr
Access Rights	r	r	r	r	r	r	r	r

Description of the bits

Bit	Symbol	Function
7	0	Reserved for future use.
6	KeyErr	This bit is set to 1, if the LoadKeyE2 or the LoadKey Command recognises, that the input data is not coded according to the Key format definition. This bit is set to 0 starting the LoadkeyE2 or the LoadKey command.
5	AccessErr	This bit is set to 1, if the access rights to the E²PROM are violated. This bit is set to 0 starting an E²PROM related command.
4	FIFOOvfl	This bit is set to 1, if the µ-Processor or a MF RC530's internal state machine (e.g. receiver) tries to write data into the FIFO buffer although the FIFO buffer is already full.
3	CRCErr	This bit is set to 1, if *RxCRCEn* is set and the CRC fails. It is cleared to 0 automatically at receiver start phase during the state PrepareRx.
2	FramingErr	This bit is set to 1, if the SOF is incorrect. It is cleared automatically at receiver start (that is during the state PrepareRx).
1	ParityErr	This bit is set to 1, if the parity check has failed. It is cleared automatically at receiver start (that is during the state PrepareRx).
0	CollErr	This bit is set to 1, if a bit-collision is detected. It is cleared automatically at receiver start (that is during the state PrepareRx).

5.2.2.4 CollPos Register

Bit position of the first bit collision detected on the RF- interface.

Name: CollPos Address: 0x0B Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
	CollPos							
Access Rights	r	r	r	r	r	r	r	r

Description of the bits

Bit	Symbol	Function
7-0	CollPos	This register shows the bit position of the first detected collision in a received frame. Example: 0x00 indicates a bit collision in the start bit 0x01 indicates a bit collision in the 1st bit 0x08 indicates a bit collision in the 8th bit

5.2.2.5 TimerValue Register

actual value of the timer.

Name: TimerValue Address:0x0C Reset value: XXXXXXXX, 0xXX

	7	6	5	4	3	2	1	0
	TimerValue							
Access Rights	r	r	r	r	r	r	r	r

Description of the bits

Bit	Symbol	Function
7-0	TimerValue	This register shows the actual value of the timer counter.

5.2.2.6 CRCResultLSB Register

LSB of the CRC-Coprocessor register.

Name: CRCResultLSB Address: 0x0D Reset value: XXXXXXXX, 0xXX

	7	6	5	4	3	2	1	0
	CRCResultLSB							
Access Rights	r	r	r	r	r	r	r	r

Description of the bits

Bit	Symbol	Function
7-0	CRCResultLSB	This register shows the actual value of the least significant byte of the CRC register. It is valid only if bit *CRCReady* is set to 1.

5.2.2.7 CRCResultMSB Register

MSB of the CRC-Coprocessor register.

Name: CRCResultMSB Address: 0x0E Reset value: XXXXXXXX, 0xXX

	7	6	5	4	3	2	1	0
	CRCResultMSB							
Access Rights	r	r	r	r	r	r	r	r

Description of the bits

Bit	Symbol	Function
7-0	CRCResultMSB	This register shows the actual value of the most significant byte of the CRC register. It is valid only if bit *CRCReady* is set to 1. For 8-bit CRC calculation the registers value is undefined.

5.2.2.8 BitFraming Register

Adjustments for bit oriented frames.

Name: BitFraming Address: 0x0F Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
	0	RxAlign			0	TxLastBits		
Access Rights	r/w	dy	dy	dy	r/w	dy	dy	dy

Description of the bits

Bit	Symbol	Function
7	0	Reserved for future use
6-4	RxAlign	Used for reception of bit oriented frames: *RxAlign* defines the bit position for the first bit received to be stored in the FIFO. Further received bits are stored in the following bit positions. After reception, *RxAlign* is cleared automatically. Example: RxAlign = 0: the LSB of the received bit is stored at bit 0, second received bit is stored at bit position 1 RxAlign = 1: the LSB of the received bit is stored at bit 1, second received bit is stored at bit position 2 RxAlign = 7: the LSB of the received bit is stored at bit 7, second received bit is stored in the following byte at bit position 0
3	0	reserved for future use
2-0	TxLastBits	Used for transmission of bit oriented frames: *TxLastBits* defines the number of bits of the last byte that shall be transmitted. A 000 indicates that all bits of the last byte shall be transmitted. After transmission, *TxLastBits* is cleared automatically.

5.2.3 PAGE 2: TRANSMITTER AND CONTROL

5.2.3.1 Page Register

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.3.2 TxControl Register

Controls the logical behaviour of the antenna pin TX1 and TX2.

Name: TxControl Address: 0x11 Reset value: 01011000, 0x58

	7	6	5	4	3	2	1	0
	0	ModulatorSource		1	TX2Inv	TX2Cw	TX2RFEn	TX1RFEn
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7	0	This value shall not be changed
6-5	Modulator Source	Selects the source for the modulator input: 00: LOW 01: HIGH 10: Internal Coder 11: Pin MFIN
4	1	This value shall not be changed
3	TX2Inv	Set to 1, the output signal on pin TX2 will deliver an inverted 13.56 MHz energy carrier.
2	TX2Cw	Set to 1, the output signal on pin TX2 will deliver continuously the un-modulated 13.56 MHz energy carrier. Setting *TX2Cw* to 0 enables modulation of the 13.56 MHz energy carrier.
1	TX2RFEn	Set to 1, the output signal on pin TX2 will deliver the 13.56 MHz energy carrier modulated by the transmission data. If *TX2RFEn* is 0, TX2 drives a constant output level.
0	TX1RFEn	Set to 1, the output signal on pin TX1 will deliver the 13.56 MHz energy carrier modulated by the transmission data. If *TX1RFEn* is 0, TX1 drives a constant output level.

5.2.3.3 CwConductance Register

Selects the conductance of the antenna driver pins TX1 and TX2.

Name: CwConductance Address: 0x12 Reset value: 00111111, 0x3F

	7	6	5	4	3	2	1	0
	0	0	GsCfgCW					
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-6	00	These values shall not be changed
5-0	GsCfgCW	The value of this register defines the conductance of the output driver. This may be used to regulate the output power and subsequently current consumption and operating distance.

5.2.3.4 PreSet13 Register

Name: PreSet13 Address: 0x13 Reset value: 00111111, 0x3F

	7	6	5	4	3	2	1	0
	0	0	1	1	1	1	1	1
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Note: These values shall not be changed!

5.2.3.5 CoderControl Register

sets the clock rate and the coding mode

Name: CoderControl Address:0x14 Reset value: 00011001, 0x19

	7	6	5	4	3	2	1	0
	0	0	CoderRate			0	0	1
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-6	00	These values shall not be changed
5-3	CoderRate	This register defines the clock rate for Coder Circuit 000: MIFARE® 848 kBaud 001: MIFARE® 424 kBaud 010: MIFARE® 212 kBaud 011: MIFARE® 106 kBaud; ISO14443 A 100: RFU 101: RFU 110: RFU 111: RFU
2-0	001	These values shall not be changed

5.2.3.6 ModWidth Register

selects the width of the modulation pulse.

Name: ModWidth Address: 0x15 Reset value: 00010011, 0x13

	7	6	5	4	3	2	1	0
	ModWidth							
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-0	ModWidth	This register defines the width of the modulation pulse according to $T_{mod} = 2 \cdot (ModWidth + 1) / fc$.

5.2.3.7 PreSet16 Register

Name: PreSet16 Address: 0x16 Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Note: These values shall not be changed!

5.2.3.8 PreSet17 Register

Name: PreSet17 Address: 0x17 Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Note: These values shall not be changed!

5.2.4 PAGE 3: RECEIVER AND DECODER CONTROL

5.2.4.1 Page Register

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.4.2 RxControl1 Register

controls receiver behaviour.

Name: RxControl1 Address: 0x19 Reset value: 01110011, 0x73

	7	6	5	4	3	2	1	0
	SubCPulses			1	0	LPOff	Gain	
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-5	SubCPulses	Defines the number of subcarrier pulses per Bit 000: 1 Pulse 001: 2 Pulses 010: 4 Pulses 011: 8 Pulses MIFARE, ISO14443A 100: 16 Pulses 101: RFU 110: RFU 111: RFU
4-3	10	These values shall not be changed
2	LPOff	Switches off a LowPassFilter at the internal amplifier.
1-0	Gain	This register defines the receivers signal voltage gain factor: 00: 20 dB 01: 24 dB 10: 31 dB 11: 35 dB

5.2.4.3 DecoderControl Register

Controls decoder behaviour.

Name: DecoderControl Address: 0x1A Reset value: 00001000, 0x08

	7	6	5	4	3	2	1	0
	0	RxMultiple	ZeroAfter Coll	0	1	0	0	RxCoding
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7	0	This value shall not be changed
6	RxMultiple	If set to 0, after receiving of the Frame the receiver is deactivated If set to 1, it is possible to receive more than one Frame.
5	ZeroAfterColl	If set to 1, any bits received after a bit-collision are masked to zero. This eases resolving the anti-collision procedure defined in ISO14443A.
4-1	0100	These values shall not be changed
0	RxCoding	0: Manchester Coding 1: BPSK Coding

5.2.4.4 BitPhase Register

selects the bit-phase between transmitter and receiver clock.

Name: BitPhase Address: 0x1B Reset value: 10101101, 0xAD

	7	6	5	4	3	2	1	0
	BitPhase							
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-0	BitPhase	Defines the phase relation between transmitter and receiver clock. Note: The correct value of this register is essential for proper operation.

5.2.4.5 RxThreshold Register

selects thresholds for the bit decoder.

Name: RxThreshold Address: 0x1C Reset value: 11111111, 0xFF

	7	6	5	4	3	2	1	0
	MinLevel				CollLevel			
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-4	MinLevel	Defines the minimum signal strength at the decoder input that shall be accepted. If the signal strength is below this level, it is not evaluated.
3-0	CollLevel	Defines the minimum signal strength at the decoder input that has to be reached by the weaker half-bit of the Manchester-coded signal to generate a bit-collision relatively to the amplitude of the stronger half-bit.

5.2.4.6 BPSKDemControl

controls BPSK demodulation

Name: BPSKDemControl Address: 0x1D Reset value: 00011110, 0x1E

	7	6	5	4	3	2	1	0
	0	0	0	FilterAmp Det	TauD		TauB	
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-5	000	These values shall not be changed
4	FilterAmpDet	Switches on a HighPassFilter for amplitude detection
3-2	TauD	Change time-constant of internal PLL during data receiving
1-0	TauB	Change time-constant of internal PLL during burst

5.2.4.7 RxControl2 Register

controls decoder behaviour and defines the input source for the receiver.

Name:RxControl2 Address: 0x1E Reset value: 01000001, 0x41

	7	6	5	4	3	2	1	0
	RcvClkSelI	RxAutoPD	0	0	0	0	DecoderSource	
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7	RcvClkSelI	If set to 1, the I-clock is used for the receiver clock. Set to 0 indicates, that the Q-clock is used. I-clock and Q-clock are 90° phase shifted to each other
6	RxAutoPD	If set to 1, the receiver circuit is automatically switched on before receiving and switched off afterwards. This may be used to reduce current consumption. If set to 0, the receiver is always activated.
5-2	0000	These values shall not be changed
1-0	DecoderSource	Selects the source for the decoder input: 00: Low 01: Internal Demodulator 10: A subcarrier modulated Manchester coded signal at Pin MFIN 11: A baseband Manchester coded signal at Pin MFIN

5.2.4.8 ClockQControl Register

controls clock generation for the 90° phase shifted Q-channel clock.

Name: ClockQControl Address: 0x1F Reset value: 000XXXXX, 0xXX

	7	6	5	4	3	2	1	0
	ClkQ180Deg	ClkQCalib	0	ClkQDelay				
Access Rights	r	r/w	r/w	dy	dy	dy	dy	dy

Description of the bits

Bit	Symbol	Function
7	ClkQ180Deg	If the Q-clock is phase shifted more than 180° compared to the I-clock, the bit *ClkQ180Deg* is set to 1, otherwise it is 0.
6	ClkQCalib	If this bit is 0, the Q-clock is calibrated automatically after the Reset Phase and after data reception from the card. If this bit is set to 1, no calibration is performed automatically.
5	0	This value shall not be changed
4-0	ClkQDelay	This register shows the number of delay elements actually used to generate a 90°phase shift of the I-clock to obtain the Q-clock. It can be written directly by the µ-Processor or by the automatic calibration cycle.

5.2.5 PAGE 4: RF-TIMING AND CHANNEL REDUNDANCY

5.2.5.1 Page Register

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.5.2 RxWait Register

Selects the time interval after transmission, before receiver starts.

Name: RxWait Address: 0x21 Reset value: 00000101, 0x06

	7	6	5	4	3	2	1	0
	RxWait							
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-0	RxWait	After data transmission, the activation of the receiver is delayed for *RxWait* bit-clocks. During this ‘frame guard time’ any signal at pin Rx is ignored.

5.2.5.3 ChannelRedundancy Register

Selects kind and mode of checking the data integrity on the RF-channel.

Name: ChannelRedundancy Address: 0x22 Reset value: 00000011, 0x03

	7	6	5	4	3	2	1	0
	0	0	CRC 3309	CRC8	RxCRCEn	TxCRCEn	ParityOdd	ParityEn
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-6	90	This value shall not be changed
5	CRC3309	If set to 1, CRC-calculation is done according ISO/IEC3309 Note: For usage according to ISO14443A this bit has to be 0.
4	CRC8	If set to 1, an 8-bit CRC is calculated. If set to 0, a 16-bit CRC is calculated.
3	RxCRCEn	If set to 1, the last byte(s) of a received frame is/are interpreted as CRC byte/s. If the CRC itself is correct the CRC byte(s) is/are not passed to the FIFO. In case of an error, the *CRCErr* flag is set. If set to 0, no CRC is expected.
2	TxCRCEn	If set to 1, a CRC is calculated over the transmitted data and the CRC byte(s) are appended to the data stream. If set to 0, no CRC is transmitted.
1	ParityOdd	If set to 1, an odd parity is generated or expected, respectively. If set to 0 an even parity is generated or expected, respectively. Note: For usage according to ISO14443A this bit has to be 1.
0	ParityEn	If set to 1, a parity bit is inserted in the transmitted data stream after each byte and expected in the received data stream after each byte (MIFARE®, ISO14443A) If set to 0, no parity bit is inserted or expected.

5.2.5.4 CRCPresetLSB Register

LSB of the preset value for the CRC register.

Name: CRCPresetLSB Address: 0x23 Reset value: 01010011, 0x63

	7	6	5	4	3	2	1	0
	CRCPresetLSB							
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-0	CRCPresetLSB	*CRCPresetLSB* defines the starting value for CRC-calculation. This value is loaded into the CRC at the beginning of transmission, reception and the CalcCRC Command, if the CRC calculation is enabled.

5.2.5.5 CRCPresetMSB Register

MSB of the preset value for the CRC register.

Name: CRCPresetMSB Address: 0x24 Reset value: 01010011, 0x63

	7	6	5	4	3	2	1	0
	CRCPresetMSB							
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-0	CRCPresetMSB	*CRCPresetMSB* defines the starting value for CRC-calculation. This value is loaded into the CRC at the beginning of transmission, reception and the CalcCRC Command, if the CRC calculation is enabled. Note: This register is not relevant, if CRC8 is 1.

5.2.5.6 PreSet25 Register

Name: PreSet25 Address: 0x25 Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Note: These values shall not be changed!

5.2.5.7 MFOUTSelect Register

Selects internal signal applied to pin MFOUT.

Name: MFOUTSelect Address: 0x26 Reset value:00000000, 0x00

	7	6	5	4	3	2	1	0
	0	0	0	0	0	MFOUTSelect		
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-3	00000	These values shall not be changed
2-0	MFOUTSelect	*MFOUTSelect* defines which signal is routed to pin MFOUT. 000 Constant Low 001 Constant High 010 Modulation Signal (envelope) from internal coder, Miller coded 011 Serial data stream, not Miller coded 100 Output signal of the energy carrier demodulator (card modulation signal) Note: only valid MIFARE® and ISO14443 A at a baudrate of 106 kbaud. 101 Output signal of the subcarrier demodulator (Manchester coded card signal) Note: only valid MIFARE® and ISO14443 A at a baudrate of 106 kbaud. 110 RFU 111 RFU

5.2.5.8 PreSet27 Register

Name: PreSet27 Address: 0x27 Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x
Access Rights	w	w	w	w	w	w	w	w

Note: These values shall not be changed!

5.2.6 PAGE 5: FIFO, TIMER AND IRQ- PIN CONFIGURATION

5.2.6.1 Page Register

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.6.2 FIFOLevel Register

Defines the level for FIFO under- and overflow warning.

Name: FIFOLevel Address: 0x29 Reset value:00001000, 0x08

	7	6	5	4	3	2	1	0
	0	0	WaterLevel					
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-6	00	These values shall not be changed
5-0	WaterLevel	This register defines, the warning level of the MF RC530 for the µ-Processor for a FIFO-buffer over- or underflow: *HiAlert* is set to 1, if the remaining FIFO-buffer space is equal or less than *WaterLevel* bytes in the FIFO-buffer. *LoAlert* is set to 1, if equal or less than *WaterLevel* bytes are in the FIFO-buffer,.

5.2.6.3 TimerClock Register

Selects the divider for the timer clock.

Name: TimerClock Address: 0x2A Reset value: 00000111, 0x07

	7	6	5	4	3	2	1	0
	0	0	TAutoRestart	TPreScaler				
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-6	00	These values shall not be changed
5	TAutoRestart	If set to 1, the timer automatically restart its count-down from *TReloadValue*, instead of counting down to zero. If set to 0 the timer decrements to zero and the bit *TimerIRq* is set to 1.
4-0	TPreScaler	Defines the timer clock f_{Timer}. *TPreScaler* can be adjusted from 0 up to 21. The following formula is used to calculate f_{Timer} : $f_{Timer} = 13.56\ MHz / 2^{TPreScaler}$.

5.2.6.4 TimerControl Register

Selects start and stop conditions for the timer.

Name: TimerControl Address: 0x2B Reset value: 00000110, 0x06

	7	6	5	4	3	2	1	0
	0	0	0	0	TStopRxEnd	TStopRxBegin	TStartTxEnd	TStartTxBegin
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-4	0000	These values shall not be changed
3	TStopRxEnd	If set to 1, the timer stops automatically when data reception ends. 0 indicates, that the timer is not influenced by this condition.
2	TStopRxBegin	If set to 1, the timer stops automatically, when the first valid bit is received. 0 indicates, that the timer is not influenced by this condition.
1	TStartTxEnd	If set to 1, the timer starts automatically when data transmission ends. If the timer is already running, the timer restarts by loading *TReloadValue* into the timer. 0 indicates, that the timer is not influenced by this condition.
0	TStartTxBegin	If set to 1, the timer is starts automatically when the first bit is transmitted. If the timer is already running, the timer restarts by loading *TReloadValue* into the timer. 0 indicates, that the timer is not influenced by this condition.

5.2.6.5 TimerReload Register

Defines the preset value for the timer.

Name: TimerReload Address: 0x2C Reset value: 00001010, 0x0A

	7	6	5	4	3	2	1	0
	TReloadValue							
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-0	TreloadValue	With a start event the timer loads with the *TreloadValue*. Changing this register affects the timer only with the next start event. If *TReloadValue* is set to 0, the timer cannot start.

5.2.6.6 IRQPinConfig Register

Configures the output stage for pin IRQ.

Name: IRQPinConfig Address: 0x2D Reset value: 00000010, 0x02

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	IRQInv	IRQPushPull
Access Rights	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Description of the bits

Bit	Symbol	Function
7-2	000000	These values shall not be changed
1	IRQInv	If set to 1, the signal on pin IRQ is inverted with respect to bit *IRq*. 0 indicates, that the signal on pin IRQ is equal to bit *IRQ*.
0	IRQPushPull	If set to 1, pin IRQ works as standard CMOS output pad. 0 indicates, that pin IRQ works as open drain output pad.

5.2.6.7 PreSet2E

Name: PreSet2E Address: 0x2E Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x
Access Rights	w	w	w	w	w	w	w	w

5.2.6.8 Preset2F

Name: PreSet2F Address: 0x2F Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x
Access Rights	w	w	w	w	w	w	w	w

5.2.7 PAGE 6: RFU

5.2.7.1 Page Register

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.7.2 RFU Registers

Name: RFU Address: 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 037 Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x
Access Rights	w	w	w	w	w	w	w	w

Note: These registers are reserved for future use.

5.2.8 PAGE 7: TEST CONTROL

5.2.8.1 Page Register

Selects the register page. See 5.2.1.1 *Page Register*.

5.2.8.2 RFU Register

Name: RFU Address: 0x39 Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x
Access Rights	w	w	w	w	w	w	w	w

Note: This register is reserved for future use.

5.2.8.3 TestAnaSelect Register

Selects analog test signals.

Name: TestAnaSelect Address: 0x3A Reset value: 00000000, 0x00

	7	6	5	4	3	2	1	0
	0	0	0	0	TestAnaOutSel			
Access Rights	w	w	w	w	w	w	w	w

Description of the bits

Bit	Symbol	Function	
7-4	0000	These values shall not be changed	
3-0	TestAnaOutSel	This register selects the internal analog signal that is routed to pin AUX. For detailed information see 20.3	
		Value	Signal Name
		0	V_{mid}
		1	$V_{bandgap}$
		2	$V_{RxFollI}$
		3	$V_{RxFollQ}$
		4	V_{RxAmpI}
		5	V_{RxAmpQ}
		6	V_{CorrNI}
		7	V_{CorrNQ}
		8	V_{CorrDI}
		9	V_{CorrDQ}
		A	V_{EvalL}
		B	V_{EvalR}
		C	V_{Temp}
		D	RFU
		E	RFU
		F	RFU

5.2.8.4 RFU Register

Name: RFU Address: 0x3B Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x
Access Rights	w	w	w	w	w	w	w	w

Note: This register is reserved for future use.

5.2.8.5 RFU Register

Name: RFU Address: 0x3C Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x
Access Rights	w	w	w	w	w	w	w	w

Note: This register is reserved for future use.

5.2.8.6 TestDigiSelect Register

Selects digital test mode.

Name: TestDigiSelect Address:0x3D Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	SignalTo MFOUT	TestDigiSignalSel						
Access Rights	w	w	w	w	w	w	w	w

Description of the bits

Bit	Symbol	Function	
7	SignalToMFOUT	Set to 1, overrules the setting in *MFOUTSelect* and the digital test signal defined in *TestDigiSignalSel* is routed to pin MFOUT instead. Set to 0, *MFOUTSelect* defines the signal delivered at pin MFOUT.	
6-0	TestDigiSignalSel	Selects the digital test signal to be routed to pin MFOUT. For detailed information refer to chapter 20.4	
		TestDigiSignalSel	Signal Name
		$F4_{hex}$	s_data
		$E4_{hex}$	s_valid
		$D4_{hex}$	s_coll
		$C4_{hex}$	s_clock
		$B5_{hex}$	rd_sync
		$A5_{hex}$	wr_sync
		96_{hex}	int_clock
		83_{hex}	BPSK_out
		$E2_{hex}$	BPSK_sig

5.2.8.7 RFU Registers

Name: RFU Address: 0x3E Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x
Access Rights	w	w	w	w	w	w	w	w

Note: These registers are reserved for future use.

5.2.8.8 RFU Register

Name: RFU Address: 0x3F Reset value: xxxxxxxx, 0xxx

	7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x	x
Access Rights	w	w	w	w	w	w	w	w

Note: This register is reserved for future use.

5.3 MF RC530 Register Flags Overview

Flag(s)	Register	Address Register, Bit Position
AccessErr	ErrorFlag	0x0A, bit 5
BitPhase	BitPhase	0x1B, bits 7:0
ClkQ180Deg	ClockQControl	0x1F, bit 7
ClkQCalib	ClockQControl	0x1F, bit 6
ClkQDelay	ClockQControl	0x1F, bits 4:0
CoderRate	CoderControl	0x14, bits 5:3
CollErr	ErrorFlag	0x0A, bit 0
CollLevel	RxThreshold	0x1C, bits 3:0
CollPos	CollPos	0x0B, bits 7:0
Command	Command	0x01, bits 5:0
CRC3309	ChannelRedundancy	0x22, bit 5
CRC8	ChannelRedundancy	0x22, bit 4
CRCErr	ErrorFlag	0x0A, bit 3
CRCPresetLSB	CRCPresetLSB	0x23, bits 7:0
CRCPresetMSB	CRCPresetMSB	0x24, bits 7:0
CRCReady	SecondaryStatus	0x05 , bit 5
CRCResultMSB	CRCResultMSB	0x0E, bits 7:0
CRCResultLSB	CRCResultLSB	0x0D, bits 7:0
Crypto1On	Control	0x09, bit 3
DecoderSource	RxControl2	0x1E, bits 1:0
E2Ready	SecondaryStatus	0x05, bit 6
Err	PrimaryStatus	0x03, bit 2
FIFOData	FIFOData	0x02, bits 7:0
FIFOLength	FIFOLength	0x04, bits 7:0
FIFOOvfl	ErrorFlag	0x0A, bit 4
FilterAmpDet	BPSKDemControl	0x1D, bit 4
FlushFIFO	Control	0x09, bit 0
FramingErr	ErrorFlag	0x0A, bit 2
Gain	RxControl1	0x19, bits 1:0
GsCfgCW	CWConductance	0x12, bits 5:0
HiAlert	PrimaryStatus	0x03, bit 1
HiAlertIEn	InterruptEn	0x06, bit 1
HiAlertIRq	InterruptRq	0x07, bit 1

Flag(s)	Register	Address Register, Bit Position
IdleIEn	InterruptEn	0x06, bit 2
IdleIRq	InterruptRq	0x07, bit 2
IFDetectBusy	Command	0x01, bit 7
Irq	PrimaryStatus	0x03, bit 3
IRQInv	IRQPinConfig	0x2D, bit 1
IRQPushPull	IRQPinConfig	0x2D, bit 0
KeyErr	ErrorFlag	0x0A, bit 6
LoAlert	PrimaryStatus	0x03, bit 0
LoAlertIEn	InterruptEn	0x06, bit 0
LoAlertIRq	InterruptRq	0x07, bit 0
LPOff	RxControl1	0x19, bit 2
MFOUTSelect	MFOUTSelect	0x26, bits 2:0
MinLevel	RxThreshold	0x1C, bits 7:4
ModemState	PrimaryStatus	0x03 , bit 6:4
ModulatorSource	TxControl	0x11, bits 6:5
ModWidth	ModWidth	0x15, bits 7:0
PageSelect	Page	0x00, 0x08, 0x10, 0x18, 0x20, 0x28, 0x30, 0x38, bits 2:0
ParityEn	ChannelRedundancy	0x22, bit 0
ParityErr	ErrorFlag	0x0A, bit 1
ParityOdd	ChannelRedundancy	0x22 , bit 1
PowerDown	Control	0x09, bit4
RcvClkSell	RxControl2	0x1E, bit 7
RxAlign	BitFraming	0x0F, bits 6:4
RxAutoPD	RxControl2	0x1E, bit 6
RxCRCEn	ChannelRedundancy	0x22, bit 3
RxCoding	DecoderControl	0x1A, bit 0
RxIEn	InterruptEn	0x06, bit 3
RxIRq	InterruptRq	0x07, bit 3
RxLastBits	SecondaryStatus	0x05, bits 2:0
RxMultiple	DecoderControl	0x1A,bit 6
RxWait	RxWait	0x21, bits 7:0
SetIEn	InterruptEn	0x06, bit 67
SetIRq	InterruptRq	0x07, bit 7
SignalToMFOUT	TestDigiSelect	0x3D, bit 7

Flag(s)	Register	Address Register, Bit Position
StandBy	Control	0x09, bit 5
SubCPulses	RxControl1	0x19, bits 7:5
TauB	BPSKDemControl	0x1D, bits 1:0
TauD	BPSKDemControl	0x1D, bits 3:2
TautoRestart	TimerClock	0x2A, bit 5
TestAnaOutSel	TestAnaSelect	0x3A, bits 6:4
TestDigiSignalSel	TestDigiSelect	0x3D, bit 6:0
TimerIEn	InterruptEn	0x06, bit 5
TimerIRq	InterruptRq	0x07, bit 5
TimerValue	TimerValue	0x0C, bits 7:0
TpreScaler	TimerClock	0x2A, bits 4:0
TreloadValue	TimerReload	0x2C, bits 7:0
Trunning	SecondaryStatus	0x05, bit 7
TstartTxBegin	TimerControl	0x2B, bit 0
TstartTxEnd	TimerControl	0x2B, bit 1
TstartNow	Control	0x09, bit 1
TstopRxBegin	TimerControl	0x2B, bit 2
TstopRxEnd	TimerControl	0x2B, bit 3
TstopNow	Control	0x09, bit 2
TX1RFEn	TxControl	0x11, bit 0
TX2Cw	TxControl	0x11, bit 3
TX2Inv	TxControl	0x11, bit 3
TX2RFEn	TxControl	0x11, bit 1
TxCRCEn	ChannelRedundancy	0x22, bit 2
TxIEn	InterruptEn	0x06, bit 4
TxIRq	InterruptRq	0x07, bit 4
TxLastBits	BitFraming	0x0F, bits 2:0
UsePageSelect	Page	0x00, 0x08, 0x10, 0x18, 0x20, 0x28, 0x30, 0x38, bit 7
WaterLevel	FIFOLevel	0x29, bits 5:0
ZeroAfterColl	DecoderControl	0x1A, bit 5

5.4 Modes of Register Addressing

Three mechanisms are valid to operate with the MF RC530:

- Initiating functions and controlling data manipulation by executing *commands*
- Configuring electrical and functional behaviour via a set of *configuration bits*
- Monitoring the state of the MF RC530 by reading *status flags*

The commands, configuration bits and flags are accessed via the µ-Processor interface.
The MF RC530 can internally address 64 registers. This basically requires six address lines.

5.4.1 PAGING MECHANISM

The MF RC530 register set is segmented into 8 pages with 8 register each. The *Page-Register* can always be addressed, no matter which page is currently selected.

5.4.2 DEDICATED ADDRESS BUS

Using the MF RC530 with dedicated address bus, the µ-Processor defines three address lines via the address pins A0, A1, and A2. This allows addressing within a page. To switch between registers in different pages the paging mechanism needs then to be used.

The following table shows how the register address is assembled:

Register Bit: *UsePageSelect*	**Register-Address**					
1	*PageSelect2*	*PageSelect1*	*PageSelect0*	A2	A1	A0

Table 5-3: Dedicated Address Bus: Assembling the Register Address

5.4.3 MULTIPLEXED ADDRESS BUS

Using the MF RC530 with multiplexed address bus, the µ-Processor may define all 6 address lines at once. In this case either the paging mechanism or linear addressing may be used.

The following table shows how the register address is assembled:

Interface Bus Type	**Register Bit: *UsePageSelect***	**Register-Address**					
Multiplexed Address Bus (paging mode)	1	*PageSelect2*	*PageSelect1*	*PageSelect0*	AD2	AD1	AD0
Multiplexed Address Bus (linear addressing)	0	AD5	AD4	AD3	AD2	AD1	AD0

Table 5-4: Multiplexed Address Bus: Assembling the Register Address

6 MEMORY ORGANISATION OF THE E²PROM

6.1 Diagram of the E²PROM Memory Organisation

Block Number	Block Address	Byte Addresses	Access Rights	Memory Content	See Also
0	0	00 … 0F	r	Product Information Field	6.2
1	1	10 … 1F	r/w	Start Up Register Initialisation File	6.3.1
2	2	20 … 2F	r/w		
3	3	30 … 3F	r/w	Register Initialisation File	6.3.3
4	4	40 … 4F	r/w		
5	5	50 … 5F	r/w		
6	6	60 … 6F	r/w		
7	7	70 … 7F	r/w		
8	8	80 … 8F	w	Keys for Crypto1	6.4
9	9	90 … 9F	w		
10	A	A0 … AF	w		
11	B	B0 … BF	w		
12	C	C0 … CF	w		
13	D	D0 … DF	w		
14	E	E0 … EF	w		
15	F	F0 … FF	w		
16	10	100 … 10F	w		
17	11	110 … 11F	w		
18	12	120 … 12F	w		
19	13	130 … 13F	w		
20	14	140 … 14F	w		
21	15	150 … 15F	w		
22	16	160 … 16F	w		
23	17	170 … 17F	w		
24	18	180 … 18F	w		
25	19	190 … 19F	w		
26	1A	1A0 … 1AF	w		
27	1B	1B0 … 1BF	w		
28	1C	1C0 … 1CF	w		
29	1D	1D0 … 1DF	w		
30	1E	1E0 … 1EF	w		
31	1F	1F0 … 1FF	w		

Table 6-1: Diagram of E²PROM Memory Organisation

6.2 Product Information Field (Read Only)

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Meaning	Product Type Identification					RFU			Product Serial Number				Internal			CRC

Table 6-2: Product Information Field

PRODUCT TYPE IDENTIFICATION:

The MF RC530 is a member of a new family for highly integrated reader IC's. Each member of the product family has its unique Product Type Identification. The value of the Product Type Identification is shown in the table below:

	Product Type Identification				
Byte	0	1	2	3	4
Value	30_{hex}	88_{hex}	FE_{hex}	03_{hex}	XX_{hex}

Table 6-3: Product Type Identification Definition

Byte 4 indicates the current version number.

PRODUCT SERIAL NUMBER:

The MF RC530 holds a four bytes serial number that is unique for each device.

INTERNAL:

These 3 bytes hold internal trimming parameters.

CRC:

The content of the product information field is secured via a CRC-byte, which is checked during start up.

6.3 Register Initialisation Files (Read/Write)

Register initialisation in the register address range from 10_{hex} to $2F_{hex}$ is done automatically during the Initialising Phase (see 11.3), using the Start Up Register Initialisation File.
Furthermore, the user may initialise the MF RC530 registers with values from the Register Initialisation File executing the *LoadConfig-Command* (see 17.6.1).

Notes:

- The *Page-Register* (addressed with 10_{hex}, 18_{hex}, 20_{hex}, 28_{hex}) is skipped and not initialised.
- Make sure, that all *PreSet* registers are not changed.
- Make sure, that all register bits that are reserved for future use (RFU) are set to 0.

6.3.1 START UP REGISTER INITIALISATION FILE (READ/WRITE)

The content of the E²PROM memory bock address 1 and 2 are used to initialise the MF RC530 registers 10_{hex} to $2F_{hex}$ during the Initialising Phase automatically. The default values written into the E²PROM during production are shown chapter 6.3.2.

The assignment is the following:

E²PROM Byte Address	Register Address	Remark
10_{hex} (Block 1, Byte 0)	10_{hex}	Skipped
11_{hex}	11_{hex}	Copied
…	…	…
$2F_{hex}$ (Block 2, Byte 15)	$2F_{hex}$	Copied

Table 6-4: Byte Assignment for Register Initialisation at Start Up

6.3.2 SHIPMENT CONTENT OF START UP REGISTER INITIALISATION FILE

During production test, the Start Up Register Initialisation File is initialised with the values shown in the table below. With each power up these values are written into the MF RC530 register during the Initialising Phase.

E²PROM Byte Address	Reg. Address	Value	Description
10	10	00	*Page*: free for user
11	11	58	*TxControl:* Transmitter pins TX1 and TX2 switched off, bridge driver configuration, modulator driven from internal digital circuitry
12	12	3F	*CwConductance:* Source resistance of TX1 and TX2 to minimum.
13	13	3F	*PreSet13*
14	14	19	*CoderControl:* ISO14443A coding is set
15	15	13	*ModWidth*: Pulse width for Miller pulse coding is set to standard configuration.
16	16	00	*PreSet16*
17	17	3B	*PreSet17*
18	18	00	*Page:* free for user
19	19	73	*RxControl1:* ISO 14443A is set and internal amplifier gain is maximum.
1A	1A	08	*DecoderControl:* A bit-collision always evaluates to HIGH in the data bit stream.
1B	1B	AD	*BitPhase: BitPhase* is set to standard configuration.
1C	1C	FF	*RxThreshold: MinLevel* and *CollLevel* are set to maximum.
1D	1D	1E	*BPSKDemControl:* ISO14443A is set
1E	1E	41	*RxControl2:* Use Q-clock for the receiver, 'Automatic Receiver Off' is switched on, decoder is driven from internal analog circuitry.
1F	1F	00	*ClockQControl:* 'Automatic Q-clock Calibration' is switched on.
20	20	00	*Page:* free for user
21	21	06	*RxWait*: Frame Guard Time is set to six bit clocks.
22	22	03	*ChannelRedundancy:* Channel Redundancy is set according to ISO14443A.
23	23	63	*CRCPresetLSB:* CRC-Preset value is set according to ISO14443A.
24	24	63	*CRCPresetMSB:* CRC-Preset value is set according to ISO14443A.
25	25	00	*PreSet25*
26	26	00	*MFOUTSelect:* Pin MFOUT is set to LOW.
27	27	00	*PreSet27*
28	28	00	*Page:* free for user
29	29	08	*FIFOLevel*: *WaterLevel* FIFO buffer warning level is set to standard configuration.
2A	2A	07	*TimerClock: TPreScaler* is set to standard configuration, timer unit restart function is switched off.
2B	2B	06	*TimerControl:* Timer is started at the end of transmission, stopped at the beginning of reception.
2C	2C	0A	*TimerReload: TReloadValue:* the timer unit preset value is set to standard configuration.
2D	2D	02	*IRQPinConfig:* Pin IRQ is set to high impedance.
2E	2E	00	*PreSet2E*
2F	2F	00	*PreSet2F*

Table 6-5: Shipment Content of Start Up Configuration File

6.3.3 REGISTER INITIALISATION FILE (READ/WRITE)

The content of the E²PROM memory from block address 3 to 7 may be used to initialise the MF RC530 registers 10_{hex} to $2F_{hex}$ by execution of the *LoadConfig-Command* (see 17.6.1). It requires a two bytes argument, used as the two bytes long E²PROM starting byte address for the initialisation procedure.

The assignment is the following:

E²PROM Byte Address	Register Address	Remark
Starting Byte address for the E²PROM	10_{hex}	Skipped
Starting Byte address for the E²PROM +1	11_{hex}	Copied
…	…	…
Starting Byte address for the E²PROM + 31	$2F_{hex}$	Copied

Table 6-6: Byte Assignment for Register Initialisation at Start Up

The Register Initialisation File is big enough to hold the values for two initialisation sets and leaves one more block (16 bytes) for the user.

Note: The Register Initialisation File is read- and write-able for the user. Therefore, these bytes may also be used to store user specific data for other purposes.

6.4 Crypto1 Keys (Write Only)

6.4.1 KEY FORMAT

To store a key in the E²PROM, it has to be written in a specific format. Each key byte has to be split into the lower four bits k0 to k3 (lower nibble) and the higher four bits k4 to k7 (higher nibble). Each nibble is stored twice in one byte and one of the two nibbles is bit-wise inverted. This format is a precondition for successful execution of the *LoadKeyE2-* (see 17.8.1) and the *LoadKey-Command* (see 17.8.2).
With this format, 12 bytes of the E²PROM memory are needed to store a 6 byte long key.

This is shown in the following table:

Master Key Byte	0 (LSB)		1			5 (MSB)	
Master Key Bits	$\overline{k7\ k6\ k5\ k4}$ k7 k6 k5 k4	$\overline{k3\ k2\ k1\ k0}$ k3 k2 k1 k0	$\overline{k7\ k6\ k5\ k4}$ k7 k6 k5 k4	$\overline{k3\ k2\ k1\ k0}$ k3 k2 k1 k0		$\overline{k7\ k6\ k5\ k4}$ k7 k6 k5 k4	$\overline{k3\ k2\ k1\ k0}$ k3 k2 k1 k0
E²PROM Byte Address	n	n+1	n+2	n+3		n+10	n+11
Example	$5A_{hex}$	$F0_{hex}$	$5A_{hex}$	$E1_{hex}$		$5A_{hex}$	$A5_{hex}$

Table 6-7: Key Storage Format

Example: For the actual key A0 A1 A2 A3 A4 $A5_{hex}$ the value 5A F0 5A E1 5A D2 5A C3 5A B4 5A $A5_{hex}$ must be written into the E²PROM.

Note: Although it is possible to load data of any other format into the key storage location of the E²PROM, it is not possible to obtain a valid card authentication with such a key. The *LoadKeyE2-Command* (see 17.8.1) will fail.

6.4.2 STORAGE OF KEYS IN THE E²PROM

The MF RC530 reserves 384 bytes of memory area in the E²PROM to hold Crypto1 keys. It uses no memory segmentation to mirror the 12 bytes structure of key storage. Thus, every byte of the dedicated memory area may be the start of a key.

Example: If a key loading cycle starts at the last byte address of an E²PROM block, e.g. key byte 0 is stored at $12F_{hex}$, the following bytes are stored in the next E²PROM block , e.g. key byte 1 is stored at 130_{hex}, byte 2 at 131_{hex}, up to byte 11 at $13A_{hex}$.

With 384 bytes of memory and 12 bytes needed for one key, 32 different keys may be stored in the E²PROM.

Note: It is not possible to load a key exceeding the E²PROM byte location $1FF_{hex}$.

7 FIFO BUFFER

7.1 Overview

An 8x64 bit FIFO buffer is implemented in the MF RC530 acting as a parallel-to-parallel converter. It buffers the input and output data stream between the µ-Processor and the internals of the MF RC530. Thus, it is possible to handle data streams with lengths of up to 64 bytes without taking timing constraints into account.

7.2 Accessing the FIFO Buffer

7.2.1 ACCESS RULES

The FIFO-buffer input and output data bus is connected to the *FIFOData Register*. Writing to this register stores one byte in the FIFO-buffer and increments the internal FIFO-buffer write-pointer. Reading from this register shows the FIFO-buffer contents stored at the FIFO-buffer read-pointer and increments the FIFO-buffer read-pointer. The distance between the write- and read-pointer can be obtained by reading the *FIFOLength Register*.

When the µ-Processor starts a command, the MF RC530 may, while the command is in progress, access the FIFO-buffer according to that command. Physically only one FIFO-buffer is implemented, which can be used in input- and output direction. Therefore the µ-Processor has to take care, not to access the FIFO-buffer in an unintended way.

The following table gives an overview on FIFO access during command processing:

Active Command	µ-Processor is allowed to		Remark
	Write to FIFO	Read from FIFO	
StartUp	-	-	
Idle	-	-	
Transmit	✓	-	
Receive	-	✓	
Transceive	✓	✓	µ-Processor has to know the actual state of the command (transmitting or receiving)
WriteE2	✓	-	
ReadE2	✓	✓	The µ-Processor has to prepare the arguments, afterwards only reading is allowed
LoadKeyE2	✓	-	
LoadKey	✓	-	
Authent1	✓	-	
Authent2	-	-	
LoadConfig	✓	-	
CalcCRC	✓	-	

Table 7-1: Allowed Access to the FIFO-Buffer

7.3 Controlling the FIFO-Buffer

Besides writing to and reading from the FIFO-buffer, the FIFO-buffer pointers might be reset by setting the bit *FlushFIFO*. Consequently, *FIFOLength* becomes zero, *FIFOOvfl* is cleared, the actually stored bytes are not accessible anymore and the FIFO-buffer can be filled with another 64 bytes again.

7.4 Status Information about the FIFO-Buffer

The µ-Processor may obtain the following data about the FIFO-buffers status:

- Number of bytes already stored in the FIFO-buffer: *FIFOLength*
- Warning, that the FIFO-buffer is quite full: *HiAlert*
- Warning, that the FIFO-buffer is quite empty: *LoAlert*
- Indication, that bytes were written to the FIFO-buffer although it was already full: *FIFOOvfl*
  FIFOOvfl can be cleared only by setting bit *FlushFIFO*.

The MF RC530 can generate an interrupt signal

- If *LoAlertIRq* is set to 1 it will activate Pin IRQ when *LoAlert* changes to 1.
- If *HiAlertIRq* is set to 1 it will activate Pin IRQ when *HiAlert* changes to 1.

The flag *HiAlert* is set to 1 if only *WaterLevel* bytes or less can be stored in the FIFO-buffer. It is generated by the following equation:

$$HiAlert = (64 - FIFOLength) \leq WaterLevel$$

The flag *LoAlert* is set to 1 if *WaterLevel* bytes or less are actually stored in the FIFO-buffer. It is generated by the following equation:

$$LoAlert = FIFOLength \leq WaterLevel$$

7.5 Register Overview FIFO Buffer

The following table shows the related flags of the FIFO buffer in alphabetic order.

Flags	**Register**	**Address** Register, bit position
FIFOLength	FIFOLength	0x04, bits 6-0
FIFOOvfl	ErrorFlag	0x0A, bit 4
FlushFIFO	Control	0x09, bit 0
HiAlert	PrimaryStatus	0x03, bit 1
HiAlertIEn	InterruptIEn	0x06, bit 1
HiAlertIRq	InterruptIRq	0x07, bit 1
LoAlert	PrimaryStatus	0x03, bit 0
LoAlertIEn	InterruptIEn	0x06, bit 0
LoAlertIRq	InterruptIRq	0x07, bit 0
WaterLevel	FIFOLevel	0x29, bits 5-0

Table 7-2. Registers associated with the FIFO Buffer

8 INTERRUPT REQUEST SYSTEM

8.1 Overview

The MF RC530 indicates certain events by setting bit *IRq* in the *PrimaryStatus-Register* and, in addition, by activating pin IRQ. The signal on pin IRQ may be used to interrupt the µ-Processor using its interrupt handling capabilities. This allows the implementation of efficient µ-Processor software.

8.1.1 INTERRUPT SOURCES OVERVIEW

The following table shows the integrated interrupt flags, the related source and the condition for its setting.
The interrupt flag *TimerIRq* indicates an interrupt set by the timer unit. The setting is done when the timer decrements from 1 either down to zero (*TAutoRestart flag disabled*) or to the TPreLoad value if TAutoRestart is enabled.
The *TxIRq* bit indicates interrupts from different sources. If the transmitter is active and the state changes from sending data to transmitting the end of frame pattern, the transmitter unit sets automatically the interrupt bit. The CRC coprocessor sets *TxIRq* after having processed all data from the FIFO buffer. This is indicated by the flag *CRCReady* = 1. If the E^2Prom programming has finished the *TxIRq* bit is set, indicated by the bit *E2Ready* = 1.
The *RxIRq* flag indicates an interrupt when the end of the received data is detected.
The flag *IdleIRq* is set if a command finishes and the content of the command register changes to idle.
The flag *HiAlertIRq* is set to 1 if the *HiAlert* bit is set to one, that means the FIFO buffer has reached the level indicated by the bit *WaterLevel*, see chapter 7.4.
The flag *LoAlertIRq* is set to 1 if the *LoAlert* bit is set to one, that means the FIFO buffer has reached the level indicated by the bit *WaterLevel*, see chapter 7.4.

Interrupt Flag	Interrupt Source	Is set automatically, when
TimerIRq	Timer Unit	the timer counts from 1 to 0
TxIRq	Transmitter	a data stream, transmitted to the card, ends
	CRC-Coprocessor	all data from the FIFO buffer has been processed
	E^2PROM	all data from the FIFO buffer has been programmed
RxIRq	Receiver	a data stream, received from the card, ends
IdleIRq	*Command Register*	a command execution finishes
HiAlertIRq	FIFO-buffer	the FIFO-buffer is getting full
LoAlertIRq	FIFO-buffer	the FIFO-buffer is getting empty

Table 8-1: Interrupt Sources

8.2 Implementation of Interrupt Request Handling

8.2.1 CONTROLLING INTERRUPTS AND THEIR STATUS

The MF RC530 informs the µ-Processor about the interrupt request source by setting the according bit in the *InterruptRq Register*. The relevance of each interrupt request bit as source for an interrupt may be masked with the interrupt enable bits of the *InterruptEn Register*.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
InterruptEn	SetIEn	rfu	TimerIEn	TxIEn	RxIEn	IdleIEn	HiAlertIEn	LoAlertIEn
InterruptRq	SetIRq	rfu	TimerIRq	TxIRq	RxIRq	IdleIRq	HiAlertIRq	LoAlertIRq

Table 8-2: Interrupt Control Registers

If any interrupt request flag is set to 1 (showing that an interrupt request is pending) and the corresponding interrupt enable flag is set the status flag *IRq* in the *PrimaryStatus Register* is set to 1. Furthermore different interrupt sources can be set active simultaneously. Therefore, all interrupt request bits are ‘OR’ed and connected to the flag *IRq* and forwarded to pin IRQ.

8.2.2 ACCESSING THE INTERRUPT REGISTERS

The interrupt request bits are set automatically by the internal state machines of the MF RC530. Additionally the µ-Processor has access in order to set or to clear them.
A special implementation of the *InterruptRq* and the *InterruptEn* Register allows the change a single bit status without influencing the other ones. If a specific interrupt register shall be set to one, the bit *SetIxx* has to be set to 1 and simultaneously the specific bit has to be set to 1 too. Vice versa, if a specific interrupt flag shall be cleared, a zero has to be written to the *SetIxx* and simultaneously the specific address of the interrupt register has to be set to 1. If a bit content shall not be changed during the setting or clearing phase a zero has to be written to the specific bit location.

Example: writing $3F_{hex}$ to the *InterruptRq Register* clears all bits as *SetIRq* in this case is set to 0 and all other bits are set to 1. Writing 81_{hex} sets bit *LoAlertIRq* to 1 and leaves all other bits untouched.

8.3 Configuration of Pin IRQ

The logic level of the status flag *IRq* is visible at pin IRQ. In addition, the signal on pin IRQ may be controlled by the following bits of the *IRQPinConfig Register*:

- *IRQInv*: if set to 0, the signal on pin IRQ is equal to the logic level of bit *IRq*.
  If set to 1, the signal on pin IRQ is inverted with respect to bit *IRq*.
- *IRQPushPull*: if set to 1, pin IRQ has standard CMOS output characteristics
  otherwise it is an open drain output and an external resistor is necessary to achieve a HIGH level at this pin.

Note: During the Reset Phase (see 11.2) *IRQInv* is set to 1 and *IRQPushPull* to 0. This results in a high impedance at pin IRQ.

8.4 Register Overview Interrupt Request System

The following table shows the related flags of the Interrupt Request System in alphabetic order.

Flags	Register	Address Register, bit position
HiAlertIEn	InterruptEn	0x06, bit 1
HiAlertIRq	InterruptRq	0x07, bit 1
IdleIEn	InterruptEn	0x06, bit 2
IdleIRq	InterruptRq	0x07, bit 2
IRq	PrimaryStatus	0x03, bit 3
IRQInv	IRQPinConfig	0x07, bit 1
IRQPushPull	IRQPinConfig	0x07, bit 0
LoAlertIEn	InterruptEn	0x06, bit 0
LoAlertIRq	InterruptRq	0x07, bit 0
RxIEn	InterruptEn	0x06, bit 3
RxIRq	InterruptRq	0x07, bit 3
SetIEn	InterruptEn	0x06, bit 7
SetIRq	InterruptRq	0x07, bit 7
TimerIEn	InterruptEn	0x06, bit 5
TimerIRq	InterruptRq	0x07, bit 5
TxIEn	InterruptEn	0x06, bit 4
TxIRq	InterruptRq	0x07, bit 4

Table 8-3 Registers associated with the Interrupt Request System

9 TIMER UNIT

9.1 Overview

A timer is implemented in the MF RC530. It derives its clock from the 13.56 MHz chip-clock. The µ-Processor may use this timer to manage timing relevant tasks.

The timer unit may be used in one of the following configurations:

- Timeout-Counter
- Watch-Dog Counter
- Stop Watch
- Programmable One-Shot
- Periodical Trigger

The timer unit can be used to measure the time interval between two events or to indicate that a specific event occurred after a specific time. The timer can be triggered by events which will be explained in the following, but the timer itself does not influence any internal event (e.g. A timeout during data receiving does not influence the receiving process automatically). Furthermore, several timer related flags are set and these flags can be used to generate an interrupt.

9.2 Implementation of the Timer Unit

9.2.1 BLOCK DIAGRAM

The following block diagram shows the timer module.

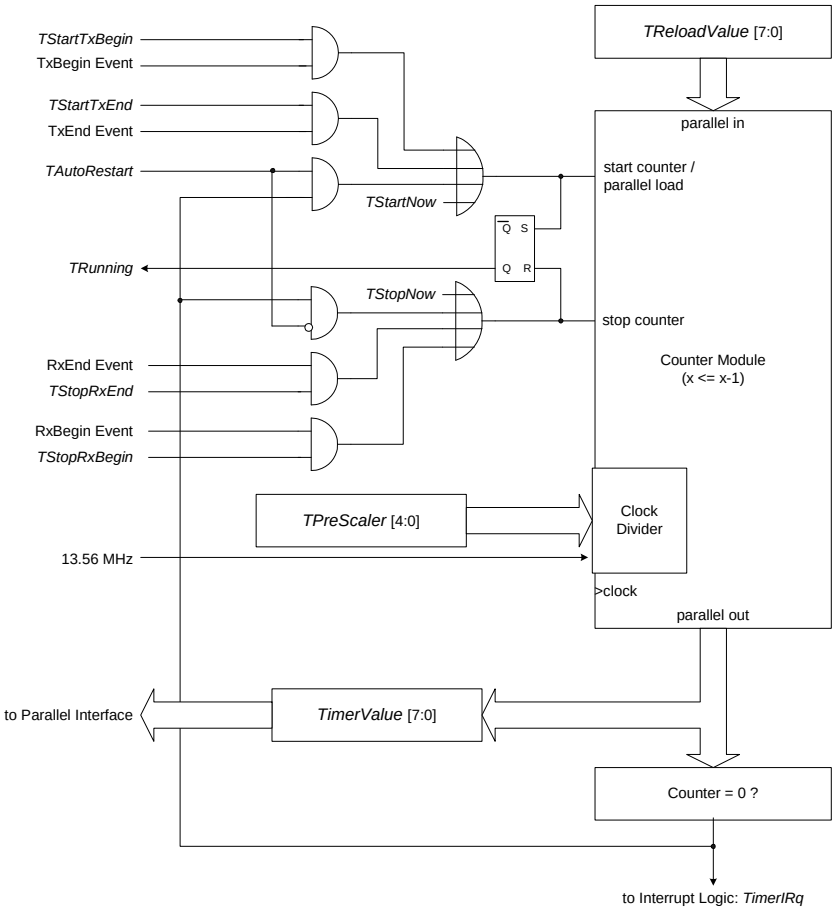


Figure 9-1: Timer Module Block Diagram

The timer unit is designed in a way, that several events in combination with enabling flags start or stop the counter. For example, setting the bit *TStartTxBegin* to 1 enables to control the receiving of data using the timer unit. In addition, the first received bit is indicated by *TxBeginEvent*. This combination starts the counter at the defined *TReloadValue*.
The timer stops either automatically if the counter value is equal to zero, or if a defined stop event happens.

9.2.2 CONTROLLING THE TIMER UNIT

The main part of the timer unit is a down-counter. As long as the down-counter value is unequal zero, it decrements its value with each timer clock.
If *TAutoRestart* is enabled the timer does not decrement down to zero. Having reached the value 1 the timer reloads with the next clock with the *TimerReload* value.
The timer is started immediately by loading a value from the *TimerReload Register* into the counter module. This may be triggered by one of the following events:

- Transmission of the first bit to the card (TxBegin Event) and bit *TStartTxBegin* is 1
- Transmission of the last bit to the card (TxEnd Event) and bit *TStartTxEnd* is 1
- Bit *TStartNow* is set to 1 (by the µ-Processor)

Note: Every start event reloads the timer from the *TimerReload Register*. Thus, the timer unit is re-triggered.

The timer can be configured to stop with one of the following events:

- Reception of the first valid bit from the card (RxBegin Event) and bit *TStopRxBegin* is set to 1
- Reception of the last bit from the card (RxEnd event) and bit *TStopRxEnd* is set to 1
- The counter module has decrement down to zero and bit *TAutoRestart* is set to 0
- Bit *TStopNow* is set to 1 (by the µ-Processor)

Loading a new value, e.g. zero, into the *TimerReload* Register does not immediately influence the counter, since the *TimerReload* Register affects the counter units content only with the next start event. Thus, the *TimerReload* Register may be changed even if the timer unit is already counting. The consequence of changing the *TimerReload* Register will be visible after the next start event.

If the counter is stopped by setting bit *TstopNow,* no *TimerIRq* is signalled.

9.2.3 TIMER UNIT CLOCK AND PERIOD

The clock of the timer unit is derived from the 13.56 MHz chip clock via a programmable divider. The clock selection is done with the *TPreScaler* Register that defines the timer unit clock frequency according to the following formula:

$$T_{TimerClock} = \frac{1}{f_{TimerClock}} = \frac{2^{TPreScaler}}{13.56MHz}$$

The possible values for the *TPreScaler* Register range from 0 up to 21 resulting in minimum time $T_{TimerClock}$ of about 74 ns up to about 150 ms.

The time period elapsed since the last start event is calculated with

$$T_{Timer} = \frac{TReLoadValue - TimerValue}{f_{TimerClock}}$$

resulting in a minimum time T_{Timer} of about 74 ns up to about 40 s.

9.2.4 STATUS OF THE TIMER UNIT

The *TRunning* bit in the *SecondaryStatus* Register shows the timer's current status. Any configured start event starts the timer at the *TReloadValue* and changes the status flag *TRunning* to 1, any configured stop event stops the timer and sets the status flag *TRunning* back to 0. As long as status flag *TRunning* is set to 1, the *TimerValue Register* changes with the next timer unit clock.
The actual timer unit content can be read directly via the *TimerValue Register*.

9.3 Usage of the Timer Unit

9.3.1 TIME-OUT- AND WATCH-DOG-COUNTER

Having started the timer by setting *TReloadValue* the timer unit decrements the *TimerValue Register* beginning with a certain start event. If a certain stop event occurs e.g. a bit is received from the card, the timer unit stops (no interrupt is generated).
On the other hand, if no stop event occurs, e.g. the card does not answer in the expected time, the timer unit decrements down to zero and generates a timer interrupt request. This signals the µ-Processor that the expected event has not occurred in the given time T_{Timer}.

9.3.2 STOP WATCH

The time T_{Timer} between a certain start- and stop event may be measured by the µ-Processor by means of the MF RC530 timer unit. Setting *TReloadValue* the timer starts to decrement. If the defined stop event occurs the timers stops. The time between start and stop can be calculated by

$$\Delta T = \left(T\,\mathrm{Re}load_{\ value} - Timer_{value}\right) * T_{Timer}$$

if the timer does not decrements down to zero.

9.3.3 PROGRAMMABLE ONE-SHOT TIMER

The µ-Processor starts the timer unit and waits for the timer interrupt. After the specified time T_{Timer} the interrupt will occur.

9.3.4 PERIODICAL TRIGGER

If the µ-Processor sets bit *TAutoRestart*, it will generate an interrupt request periodically after every T_{Timer}.

9.4 Register Overview Timer Unit

The following table shows the related flags of the Timer Unit in alphabetic order.

Flags	Register	Address
TautoRestart	TimerClock	0x2A, bit 5
TimerValue	TimerValue	0x0C, bits 7-0
TimerReloadValue	TimerReload	0x2C, bits 7-0
TpreScaler	TimerClock	0x2A, bits 4-0
Trunning	SecondaryStatus	0x05, bit 7
TstartNow	Control	0x09, bit 1
TstartTxBegin	TimerControl	0x2B, bit 0
TstartTxEnd	TimerControl	0x2B, bit 1
TstopNow	Control	0x09, bit 2
TstopRxBegin	TimerControl	0x2B, bit 2
TstopRxEnd	TimerControl	0x2B, bit 3

Table 9-1 Registers associated with the Timer Unit

10 POWER REDUCTION MODES

10.1 Hard Power Down

A Hard Power Down is enabled with HIGH on pin RSTPD. This turns off all internal current sinks including the oscillator. All digital input buffers are separated from the input pads and defined internally (except pin RSTPD itself). The output pins are frozen at a certain value.

This is shown in the following table.

SYMBOL	PIN	TYPE	DESCRIPTION
OSCIN	1	I	Not separated from input, pulled to AVSS
IRQ	2	O	High impedance
MFIN	3	I	Separated from Input
MFOUT	4	O	LOW
TX1	5	O	HIGH , if Tx1RfEn=1
			Low , if Tx1RfEn=0
TX2	7	O	HIGH, only if Tx2RfEn=1 and Tx2Inv=0
			LOW
NWR	9	I	Separated from Input
NRD	10	I	Separated from Input
NCS	11	I	Separated from Input
D0 to D7	13 to 20	I/O	Separated from Input
ALE	21	I	Separated from Input
A0	22	I/O	Separated from Input
A1	23	I	Separated from Input
A2	24	I	Separated from Input
AUX	27	O	High impedance
RX	29	I	Not changed
VMID	30	A	Pulled to AVDD
RSTPD	31	I	Not changed
OSCOUT	32	O	HIGH

Table 10-1: Signal on Pins during Hard Power Down

10.2 Soft Power Down

The Soft Power Down-mode is entered immediately by setting bit *PowerDown* in the *Control-Register*. All internal current sinks are switched off (including the oscillator buffer).
In difference to the Hard Power Down-mode, the digital input-buffers are not separated by the input pads and keep their functionality. The digital output pins do not change their state.
After resetting bit *PowerDown* in the *Control-Register* it needs 512 clocks until the Soft Power Down mode is left indicated by the *PowerDown* bit itself. Resetting it does not immediately clear it. It is cleared automatically by the MF RC530 when the Soft Power Down-Mode is left.

Note: If the internal oscillator is used, you have to take into account that it is supplied by AVDD and it will take a certain time t_{osc} until the oscillator is stable and the clock cycles can be detected by the internal logic.

10.3 Stand By Mode

The Stand By-mode is entered immediately by setting bit *StandBy* in the *Control-Register*. All internal current sinks are switched off (including the internal digital clock buffer but except the oscillator buffer).
Different from the Hard Power Down-Mode, the digital input-buffers are not separated by the input pads and keep their functionality. The digital output pins do not change their state.
Different from the Soft Power Down-Mode, the oscillator does not need time to wake up.
After resetting bit *StandBy* in the *Control-Register* it needs 4 clocks on pin OSCIN until the Stand By-Mode is left indicated by the *StandBy* bit itself. Resetting it does not immediately clear it. It is cleared automatically by the MF RC530 when the Stand By-Mode is left.

10.4 Receiver Power Down

It is power saving to switch off the receiver circuit when it is not needed and switched it on again right before data is to be received from the card. This is done automatically by setting bit *RxAutoPD* to 1. If it is set to 0 the receiver is continuously switched on.

11 START UP PHASE

The phases executed during the start up are shown in the following figure.

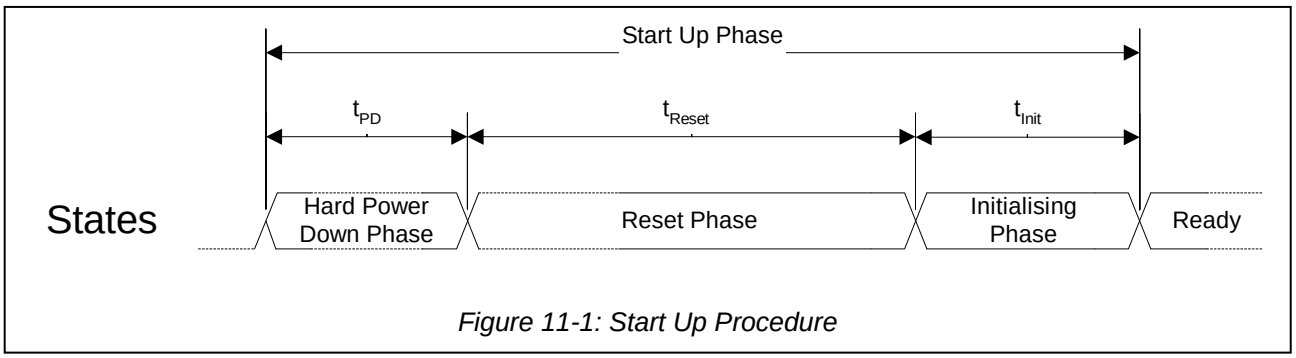


Figure 11-1: Start Up Procedure

11.1 Hard Power Down Phase

The Hard Power Down Phase is active during the following cases:

- Power On Reset caused by power up at pin DVDD
  (active while DVDD is below the digital reset threshold)
- Power On Reset caused by power up at pin AVDD
  (active while AVDD is below the analog reset threshold)
- A HIGH level on pin RSTPD
  (active while pin RSTPD is HIGH)

Note:

In case three, HIGH level on pin RSTPD, has to be at least 100µs long (t_{PD} >= 100µs). Shorter phases will not necessarily result in the reset phase $t_{Reset.}$

The slew rate of rising/falling edge on pin RSTPD is not critical because pin RSTPD is a schmitt-trigger input.

11.2 Reset Phase

The Reset Phase follows the Hard Power Down Phase automatically. One's the oscillator is running stable, it takes 512 clocks. During the Reset Phase, some of the register bits are pre-set by hardware. The respective reset values are given in the description of each register (see 5.2.).

Note: If the internal oscillator is used, you have to take into account that it is supplied by AVDD and that it will take a certain time t_{osc} until the oscillator is stable.

11.3 Initialising Phase

The Initialising Phase follows the Reset Phase automatically. It takes 128 clocks. During the Initialising Phase the content of the E²PROM blocks 1 and 2 is copied into the registers 10_{hex} to $2F_{hex}$. (see 6.3)

Note: At production test, the MF RC530 is initialised with default configuration values. This reduces the µ-Processors effort for configuring the device to a minimum.

11.4 Initialising the Parallel Interface-Type

For the different connections for the different µ-Processor interface types (see 4.3), a certain initialising sequence shall be applied to enable a proper µ-Processor interface type detection and to synchronise the µ-Processor's and the MF RC530's Start Up.

During the Start Up Phase the Command value reads as $3F_{hex}$, after the oscillator delivers a stable clock frequency with an amplitude of >90% of the nominal 13.56MHz clock. At the end of the Initialising Phase the MF RC530 enters the *Idle Command* automatically. Consequently the *Command* value changes to 00_{hex}.

To ensure proper detection of the µ-Processor interface, the following sequence shall be executed:

- Read from the *Command-Register* until the 6 bit register value for *Command* is 00_{hex}.
  The internal initialisation phase is now completed and the MF RC530 is ready to be controlled.
- Write the value 80_{hex} to the *Page-Register* to initialise the µ-Processor interface.
- Read the *Command-Register*. If its value is 00_{hex} the µ-Processor interface initialisation was successful.

Having done the interface initialisation, the linear addressing mode can be activated by writing 0x00 to the page register(s).

12 OSCILLATOR CIRCUITRY

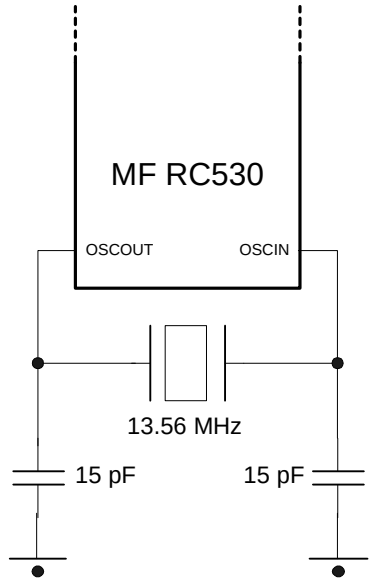


Figure 12-1: Quartz Connection

The clock applied to the MF RC530 acts as time basis for the coder and decoder of the synchronous system. Therefore stability of the clock frequency is an important factor for proper performance. To obtain highest performance, clock jitter has to be as small as possible. This is best achieved by using the internal oscillator buffer with the recommended circuitry. If an external clock source is used, the clock signal has to be applied to pin OSCIN. In this case special care for clock duty cycle and clock jitter is needed and the clock quality has to be verified. It needs to be in accordance with the specifications in chapter 21.5.3.

Remark: We do not recommend to use an external clock source.

13 TRANSMITTER PINS TX1 AND TX2

The signal delivered on TX1 and TX2 is the 13.56 MHz energy carrier modulated by an envelope signal. It can be used to drive an antenna directly, using a few passive components for matching and filtering (see chapter 19). For that, the output circuitry is designed with a very low impedance source resistance. The signal of TX1 and TX2 can be controlled via the *TxControl Register*.

13.1 Configuration of TX1 and TX2

The configuration possibilities of TX1 are described in the table below:

Register Configuration in *TxControl*	**Envelope**	**Signal on TX1**
TX1RFEn		
0	X	LOW
1	0	LOW
1	1	13.56 MHz energy carrier

Table 13-1: Configurations of Pin TX1

The configuration possibilities of TX2 are described in the table below:

Register Configuration in *TxControl*			**Envelope**	**Signal on TX2**
TX2RFEn	***TX2CW***	***InvTX2***		
0	X	X	X	LOW
1	0	0	0	LOW
			1	13.56 MHz energy carrier
		1	0	HIGH
			1	13.56 MHz energy carrier, 180° phase shift relative to TX1
	1	0	X	13.56 MHz energy carrier
		1	X	13.56 MHz energy carrier, 180° phase shift relative to TX1

Table 13-2: Configurations of Pin TX2

13.2 Operating Distance versus Power Consumption

The user has the possibility to find a trade-off between maximum achievable operating distance and power consumption using different antenna matching circuits by varying the supply voltage at the antenna driver supply pin TVDD. Different antenna matching circuits are described in the Application Note, *MIFARE® Design of MF RC500 Matching Circuit and Antennas*.

13.3 Pulse Width

The envelope carries the information of the data signal that shall be transmitted to the card done by coding the data signal according to the Miller code. Furthermore, each pause of the Miller coded signal again is coded as a pulse of certain length. The width of this pulse can be adjusted by means of the *ModWidth Register*. The pulse length is calculated by

$$T_{Pulse} = 2\frac{ModWidth + 1}{f_C}$$

where f_c = 13.56MHz.

14 RECEIVER CIRCUITRY

14.1 General

The MF RC530 employs an integrated quadrature-demodulation circuit giving the possibility to detect an ISO 14443A compliant subcarrier signal applied to pin RX. The ISO14443A sub-carrier signal is defined as a Manchester coded ASK-modulated signal. The quadrature-demodulator uses two different clocks, Q- and I-clock, with a phase shift of 90° between them. Both resulting sub-carrier signals are amplified, filtered and forwarded to a correlation circuitry. The correlation results are evaluated, digitised and passed to the digital circuitry.

For all processing units various adjustments can be made to obtain optimum performance.

14.2 Block Diagram

Figure 14-1 shows the block diagram of the receiver circuitry. The receiving process includes several steps. First the quadrature demodulation of the carrier signal of 13.56 MHz is done. To achieve an optimum in performance an automatic clock Q calibration is recommended (see 14.3.1). The demodulated signal is amplified by an adjustable amplifier. A correlation circuit calculates the degree of similarity between the expected and the received signal. The bit phase register allows to align the position of the correlation intervals with the bit grid of the received signal. In the evaluation and digitizer circuitry the valid bits are detected and the digital results are send to the FIFO register. Several tuning steps in this circuit are possible.

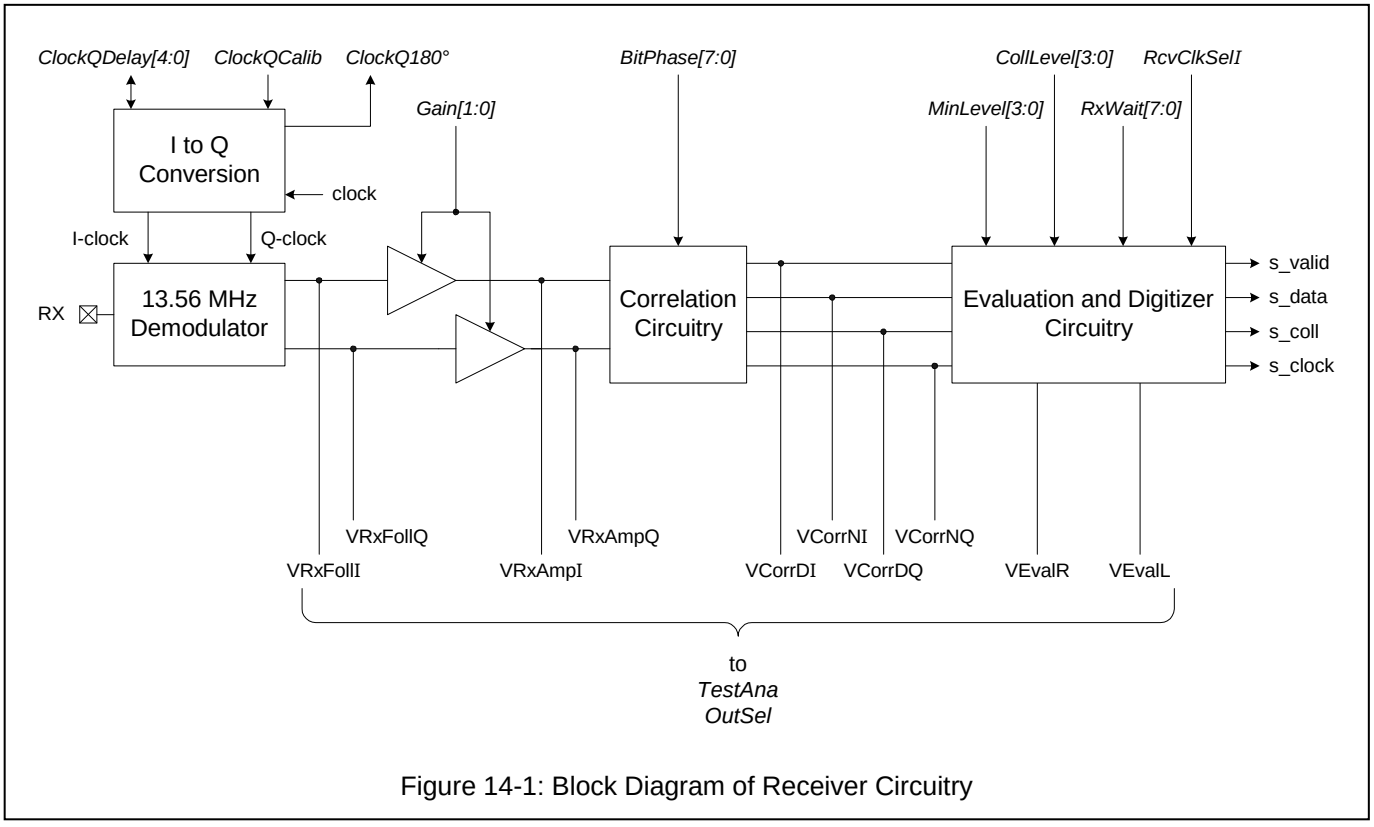


Figure 14-1: Block Diagram of Receiver Circuitry

The user may observe the signal on its way through the receiver as shown in the block diagram above. One signal at a time may be routed to pin AUX using the *TestAnaSelect-Register* as described in 20.3.

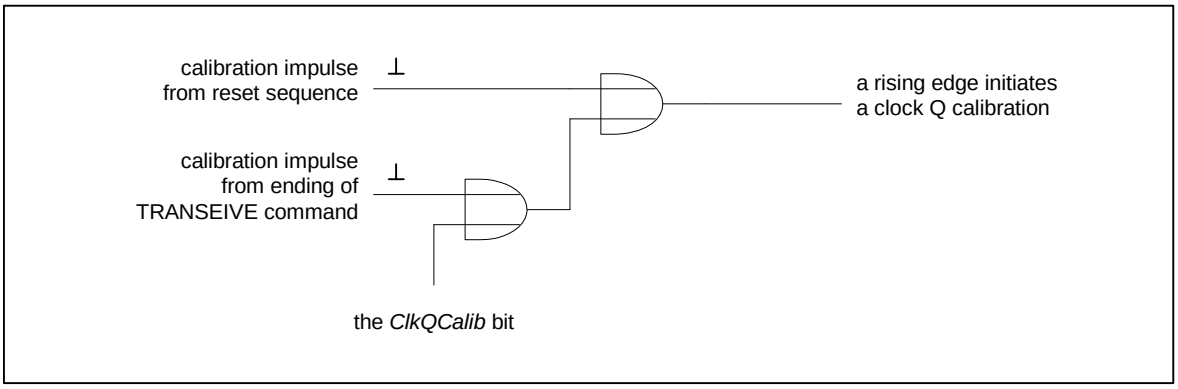
14.3 Putting the Receiver into Operation

In general, the default settings programmed in the Start Up Initialisation File are suitable to use the MF RC530 for data communication with MIFARE® cards. However, in some environments specific user settings may achieve better performance.

14.3.1 AUTOMATIC CLOCK-Q CALIBRATION

The quadrature demodulation concept of the receiver generates a phase signal I-clock and a 90° shifted quadrature signal Q-clock. To achieve an optimum demodulator performance, the Q- and the I-clock have to have a difference in phase of 90°. After the reset phase of the MF RC530, a calibration procedure is done automatically. It is possible to have an automatic calibration done at the ending of each Transceive command. To do so, the *ClkQCalib* bit has to be configured to a value of 0.
Configuring this bit to a constant value of 1 disables all automatic calibrations except the one after the reset sequence.
It is also possible to initiate one automatic calibration by software. This is done with a 0 to 1 transition of bit *ClkQCalib.*

The details:



Note: The duration of the automatic clock Q calibration takes 65 oscillator periods which is approx. 4,8µs.

The value of *ClkQDelay* is proportional to the phase shift between the Q- and the I-clock. The status flag *ClkQ180Deg* shows, that the phase shift between the Q- and the I-clock is greater than 180°.

Notes:

- The startup configuration file enables an automatically Q-clock calibration after the reset.
- While *ClkQCalib* is 1, no automatic calibration is done. Therefore leaving this bit 1 can be used to permanently disable the automatic calibration.
- It is possible to write data to *ClkQDelay* via the µ-Processor. The aim could be a disabling of the automatic calibration and to pre-set the delay by software. But notice, that configuring the delay value by software requires that bit *ClkQCalib* has already been set to 1 before and that a time interval of at least 4.8µs has elapsed since then. Each delay value must be written with the *ClkQCalib* bit set to 1. If *ClkQCalib* is 0 the configured delay value will be overwritten by the next interval automatic calibration.

14.3.2 AMPLIFIER

The demodulated signal has to be amplified with the variable amplifier to achieve the best performance. The gain of the amplifiers can be adjusted by means of the register bits *Gain[1:0]*. The following gain factors are selectable:

Register Setting	Gain Factor [dB] (Simulation Results)
0	20
1	24
2	31
3	35

Table 14-1: Gain Factors for the Internal Amplifier

14.3.3 CORRELATION CIRCUITRY

The correlation circuitry calculates the degree of matching between the received and an expected signal. The output is a measure for the amplitude of the expected signal in the received signal. This is done for both, the Q- and the I-channel. The correlator delivers two outputs for each of the two input channels, resulting in four output signals in total.

For optimum performance, the correlation circuitry needs the phase information for the signal coming from the card. This information has to be defined by the µ-Processor by means of the register *BitPhase[7:0]*. This value defines the phase relation between the transmitter and receiver clock in multiples of $t_{BitPhase}$ = 1/13.56 MHz.

14.3.4 EVALUATION AND DIGITIZER CIRCUITRY

For each bit-half of the Manchester coded signal the correlation results are evaluated. The evaluation and digitizer circuit decides from the signal strengths of both bit-halves, whether the current bit is valid, and, if it is valid, the value of the bit itself or whether the current bit-interval contains a collision.

To do this in an optimum way, the user may select the following levels:

- *MinLevel*: Defines the minimum signal strength of the stronger bit-half's signal for being considered valid.
- *CollLevel*: Defines the minimum signal strength that has to be exceeded by the weaker half-bit of the Manchester-coded signal to generate a bit-collision. If the signal's strength is below this value, a 1 and 0 can be determined unequivocally.
  CollLevel defines the minimum signal strength relative to the amplitude of the stronger half-bit.

After transmission of data, the card is not allowed to send its response before a certain time period, called frame guard time in the standard ISO14443A. The length of this time period after transmission shall be set in the *RxWait-Register*. The *RxWait-Register* defines when the receiver is switched on after data transmission to the card in multiples of one bit-duration.

If register bit *RcvClkSelI* is set to 1, the I-clock is used to clock the correlator and evaluation circuits. If set to 0, the Q-clock is used.

Note: It is recommended to use the Q-clock.

15 SERIAL SIGNAL SWITCH

15.1 General

Two main blocks are implemented in the MF RC530. A digital circuitry, comprising state machines, coder and decoder logic and so on and an analog circuitry with the modulator and antenna drivers, receiver and amplification circuitry. The interface between these two blocks can be configured in the way, that the interfacing signals may be routed to the pins MFIN and MFOUT.

This topology supports, that the analog part of the one MF RC530 may be connected to the digital part of another device.
Note: The MFIN pin can only be accessed by 106 kbaud. The Manchester with Subcarrier- and the Manchester signal can only be accessed by the MFOUT pin at 106 kbaud.

15.2 Block Diagram

Figure 15-1 describes the serial signal switches. Three different switches are implemented in the serial signal switch in order to use the MF RC530 in different configurations.
The serial signal switch may also be used during the design In phase or for test purposes to check the transmitted and received data. Chapter 20.2, describes analog test signals as well as measurements at the serial signal switch.

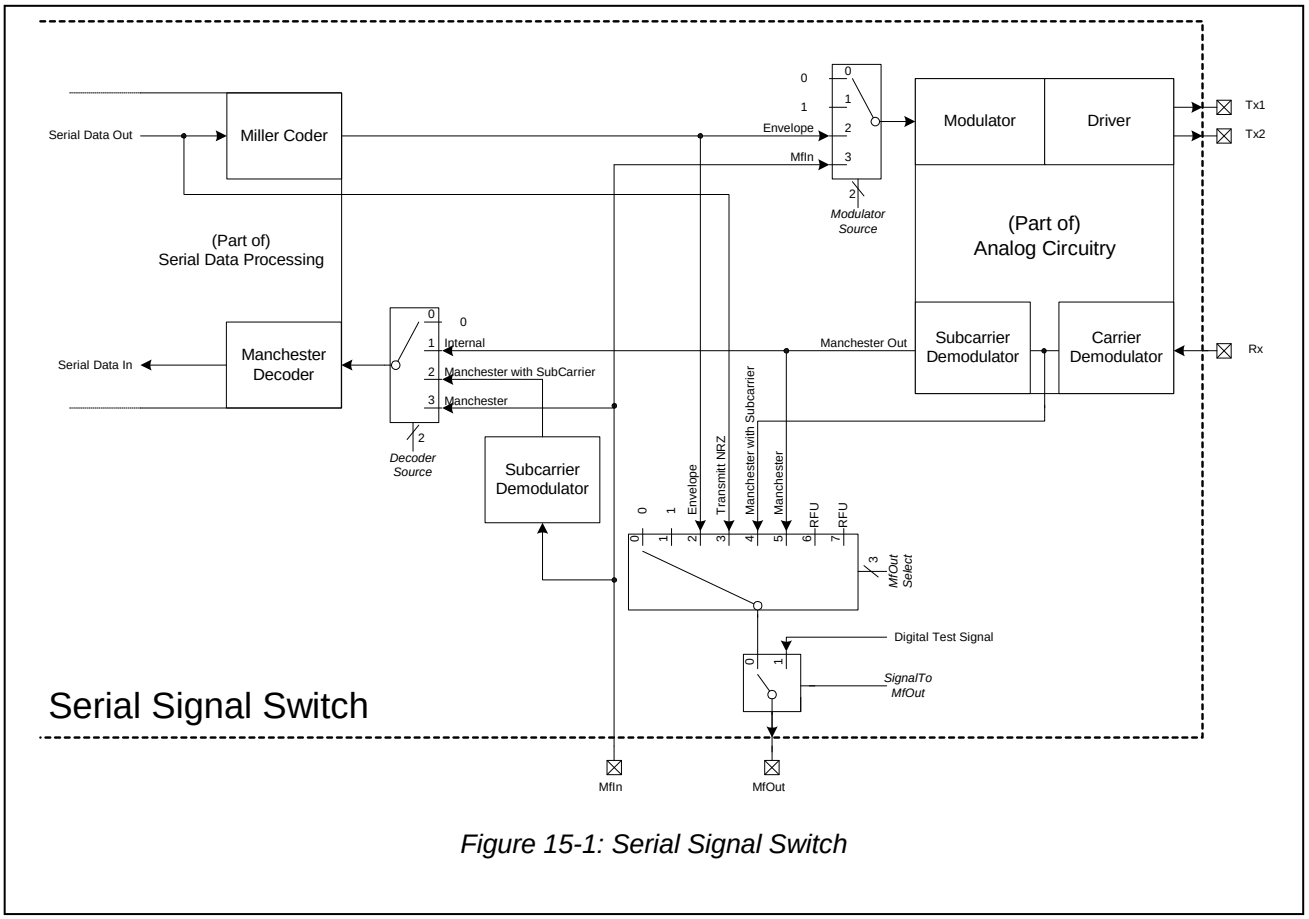


Figure 15-1: Serial Signal Switch

The following chapters describe the relevant registers used to configure and control the serial signal switch.

15.3 Registers Relevant for the Serial Signal Switch

The flags *DecoderSource* define the input signal for the internal Manchester decoder in the following way:

DecoderSource	**Input Signal for Decoder**
0	Constant 0
1	Output of the analog part. This is the default configuration.
2	Direct connection to MFIN, expecting a 847.5 kHz sub-carrier signal modulated by a Manchester coded signal.
3	Direct connection to MFIN, expecting a Manchester coded signal.

Table 15-1: Values for DecoderSource

ModulatorSource defines the signal that modulates the transmitted 13.56 MHz energy carrier. The modulated signal drives the pins TX1 and TX2.

ModulatorSource	**Input Signal for Modulator**
0	Constant 0 (energy carrier off at pin TX1 and TX2).
1	Constant 1 (continuous energy carrier delivered at pin TX1 and TX2).
2	Modulation signal (envelope) from the internal coder. This is the default configuration.
3	Direct connection to MFIN, expecting a Miller pulse coded signal.

Table 15-2: Values for ModulatorSource

MFOUTSelect selects the output signal which is routed to the pin MFOUT.

MFOUTSelect	**Signal Routed to Pin MFOUT**
0	Constant Low
1	Constant High
2	Modulation signal (envelope) from the internal coder.
3	Serial data stream that is to be transmitted (same as for *MFOUTSelect* = 2, but not coded by the Miller pulse coder yet).
4	Output signal of the receiver circuit (card modulation signal regenerated and delayed)
5	Output signal of the subcarrier demodulator (Manchester-coded card signal)
6	RFU
7	RFU

Table 15-3: Values for MFOUTSelect

Note: To use *MFOUTSelect*, the value of test signal control bit *SignalToMFOUT* has to be 0.

15.4 Usage of the MFIN and MFOUT

15.4.1 ACTIVE ANTENNA CONCEPT

The MF RC530 analog circuitry may be used via the pins MFIN and MFOUT. To do so, the following register settings have to be made:

Register	Value	Signal	At MF RC530 Pin
ModulatorSource	3	Miller Pulse Coded	MFIN
MFOUTSelect	4	Manchester Coded with sub-carrier	MFOUT
DecoderSource	X	-	-

Table 15-4: Register setting to use the MF RC530 analog circuitry only

On the other hand, the MF RC530 digital circuitry may be used via the pins MFIN and MFOUT. To do so, the following register settings have to be made:

Register	Value	Signal	At MF RC530 Pin
ModulatorSource	X	-	-
MFOUTSelect	2	Miller Pulse Coded	MFOUT
DecoderSource	2	Manchester Coded with sub-carrier	MFIN

Table 15-5: Register setting to use the MF RC530 digital circuitry only

Two MF RC530 devices configured in the above described way may be connected to each other via the pins MFOUT and MFIN.
Note: The usage of the active antenna concept is only possible with a baudrate of 106 kbaud.

15.4.2 DRIVING TWO RF-PARTS

It is possible, to connect a ‘passive antenna’ to pins TX1, TX2 and RX (via the appropriate filter and matching circuit) and at the same time an Active Antenna to the pins MFOUT and MFIN.

In this configuration, two RF-parts may be driven (one after another) by one µ-Processor.

16 MIFARE® HIGHER BAUDRATES

The MIFARE® Classic system is specified with a fix Baud-rate of 106 kBaud for the communication on the RF interface. ISO 14443A in the existing version also defines 106 kBaud at least for the initial phase of a communication between PICC and PCD.

To speed up the communication between a terminal and a card to cover requirements for large data transmission the MF RC530 supports the MIFARE® higher baudrates communication in combination with e.g.a µController IC like the MIFARE® ProX.

Communication direction	Baudrates [kbaud]
MF RC530 based PCD→ µC PICC supporting higher baudrates	106, 212, 424
µC PICC supporting higher baudrates → MF RC530 based PCD	106, 212, 424

Table 16-1 MIFARE® Higher Baudrates

The MIFARE® Higher Baudrates' concept will be described in the Application Note: '*MIFARE® Implementation of Higher Baudrates'.* This Application Note will cover also the integration a MIFARE® Higher Baudrates communication concept in current applications.

17 MF RC530 COMMAND SET

17.1 General Description

The MF RC530 behaviour is determined by an internal state machine capable to perform a certain set of commands. The commands can be started by writing the according command-code to the *Command-Register*.

Arguments and/or data necessary to process a command are mainly exchanged via the FIFO buffer.

17.2 General Behaviour

- Each command, that needs a data stream (or data byte stream) as input will immediately process the data it finds in the FIFO buffer.
- Each command that needs a certain number of arguments will start processing only when it has received the correct number of arguments via the FIFO buffer.
- The FIFO buffer is not cleared automatically at command start. Therefore, it is also possible to write the command arguments and/or the data bytes into the FIFO buffer and start the command afterwards.
- Each command (except the *StartUp-Command*) may be interrupted by the µ-Processor by writing a new command code into the *Command-Register* e.g.: the *Idle-Command*.

17.3 MF RC530 Commands Overview

Command	Code	Action	Arguments and Data passed via FIFO	Returned Data via FIFO	see Chapter
StartUp	$3F_{hex}$	Runs the Reset- and Initialisation Phase. Note: This command can not be activated by software, but only by a Power-On or Hard Reset	-	-	17.3.2
Idle	00_{hex}	No action; cancels current command execution.	-	-	17.3.3
Transmit	$1A_{hex}$	Transmits data from the FIFO buffer to the card.	Data Stream	-	17.4.1
Receive	16_{hex}	Activates receiver circuitry. Note: Before the receiver actually starts, the state machine waits until the time configured in the register *RxWait* has passed. Note: This command may be used for test purposes only, since there is no timing relation to the *Transmit-Command*.	-	Data Stream	17.4.2

MF RC530 Commands Overview Continued

Command	Code	Action	Arguments and Data passed via FIFO	Returned Data via FIFO	see Chapter
Transceive	$1E_{hex}$	Transmits data from FIFO buffer to the card and activates automatically the receiver after transmission. Note: Before the receiver actually starts, the MF RC530 waits until the time configured in the register *RxWait* has passed. Note: This command is the combination of Transmit and Receive	Data Stream	Data Stream	17.4.3
WriteE2	01_{hex}	Gets data from FIFO buffer and writes it to the internal E²PROM.	Start Address LSB Start Address MSB Data Byte Stream	-	17.5.1
ReadE2	03_{hex}	Reads data from the internal E²PROM and puts it into the FIFO buffer. Note: Keys cannot be read back	Start Address LSB Start Address MSB Number of Data Bytes	Data Bytes	17.5.2
LoadKeyE2	$0B_{hex}$	Copies a key from the E²PROM into the key buffer.	Start Address LSB Start Address MSB	-	17.8.1
LoadKey	19_{hex}	Reads a key from the FIFO buffer and puts it into the key buffer. Note: The key has to be prepared in a specific format (refer to 6.4.1, key format)	Byte0 (LSB) Byte1 … Byte 10 Byte11 (MSB)	-	17.8.2
Authent1	$0C_{hex}$	Performs the first part of the Crypto1 card authentication.	Card's Auth-Command Card's Block Address Card's Serial Number LSB Card's Serial Number Byte1 Card's Serial Number Byte2 Card's Serial Number MSB	-	17.8.3
Authent2	14_{hex}	Performs the second part of the card authentication using the Crypto1 algorithm.	-	-	17.8.4
LoadConfig	07_{hex}	Reads data from E²PROM and initialises the MF RC530 registers.	Start Address LSB Start Address MSB	-	17.6.1
CalcCRC	12_{hex}	Activates the CRC-Coprocessor. Note: The result of the CRC calculation can be read from the registers *CRCResultLSB* and *CRCResultMSB*	Data Byte-Stream	-	17.6.2

Table 17-1: MF RC530 Command Overview

17.3.1 BASIC STATES

17.3.2 STARTUP COMMAND $3F_{HEX}$

Command	$Code_{hex}$	Action	Arguments and Data	Returned Data
StartUp	3F	Runs the Reset- and Initialisation Phase Note: This command can not be activated by software, but only by a Power-On or Hard Reset	-	-

The *StartUp-Command* runs the Reset- and Initialisation Phase. It does not need or return any data. It can not be activated by the µ-Processor but is started automatically after one of the following events:

- Power On Reset caused by power up at Pin DVDD
- Power On Reset caused by power up at Pin AVDD
- Negative Edge at Pin RSTPD

The Reset-Phase defines certain register bits by an asynchronous reset. The Initialisation-Phase defines certain registers with values taken from the E²PROM.

When the *StartUp-Command* has finished, the *Idle-Command* is entered automatically.

Notes:

- The µ-Processor must not write to the MF RC530 as long as the MF RC530 is busy executing the *StartUp-Command*. To ensure this, the µ-Processor shall poll for the *Idle-Command* to determine the end of the Initialisation Phase (see also chapter 11.4).
- As long as the *StartUp-Command* is active, only reading from page 0 of the MF RC530 is possible.
- The *StartUp-Command* can not be interrupted by the µ-Processor.

17.3.3 IDLE COMMAND 00_{HEX}

Command	$Code_{hex}$	Action	Arguments and Data	Returned Data
Idle	00	No action, cancels current command execution	-	-

The *Idle-Command* switches the MF RC530 to its inactive state. In this Idle-state it waits for the next command. It does not need or return any data. The device automatically enters the Idle-state when a command finishes. In this case the MF RC530 simultaneously initiates an interrupt request by setting bit *IdleIRq*. Triggered by the µ-Processor, the *Idle-Command* may be used to stop execution of all other commands (except the *StartUp Command*). In that case no *IdleIRq* is generated.

Remark: Stopping a command with the *Idle Command* does not clear the FIFO buffer content.

17.4 Commands for Card Communication

The MF RC530 is a ISO 14443A compliant reader IC. Therefore, the command set of this IC allows more flexibility and more generalised commands compared to MIFARE® dedicated reader ICs. The following chapter describes the command set for card communication in general ending with the MIFARE® related authentication procedure.

17.4.1 TRANSMIT COMMAND $1A_{HEX}$

Command	$Code_{hex}$	Action	Arguments and Data	Returned Data
Transmit	1A	Transmits data from FIFO buffer to the card	Data Stream	-

The *Transmit-Command* takes data from the FIFO buffer and forwards it to the transmitter. It does not return any data. The *Transmit-Command* can only be started by the µ-Processor.

17.4.1.1 Working with the Transmit Command

To transmit data one of the following sequences may be used:

1. All data, that shall be transmitted to the card is written to the FIFO while the *Idle-Command* is active. After that, the command code for the *Transmit-Command* is written to the *Command-Register*.
   Note: This is possible for transmission of data with a length of up to 64 bytes.
2. The command code for the *Transmit-Command* is written to the *Command-Register* first. Since no data is available in the FIFO, the command is only enabled but transmission is not triggered yet.
   Data transmission really starts with the first data byte written to the FIFO. To generate a continuous data stream on the RF-interface, the µ-Processor has to put the next data bytes to the FIFO in time.
   Note: This allows transmission of data of any length but requires that data is available in the FIFO in time.
3. A part of the data, that shall be transmitted to the card is written to the FIFO while the *Idle-Command* is active. After that, the command code for the *Transmit-Command* is written to the *Command-Register*.
   While the *Transmit-Command* is active, the µ-Processor may feed further data to the FIFO, causing the transmitter to append it to the transmitted data stream.
   Note: This enables transmission of data of any length but requires that data is available in the FIFO in time.

When the transmitter requests the next data byte to keep the data stream on the RF-interface continuous but the FIFO buffer is empty, the *Transmit-Command* automatically terminates. This causes the internal state machine to change its state from Transmit to Idle.

If data transmission to the card is finished, the MF RC530 sets the flag *TxIRq* to signal it to the µ-Processor.

Remark: If the µ-Processor overwrites the transmit code in the *Command-Register* with the *Idle-Command* or any other command, transmission stops immediately with the next clock cycle. This may produce output signals that are not according to ISO14443A.

17.4.1.2 RF-Channel Redundancy and Framing

Each transmitted frame consists of a SOF (start of frame) pattern, followed by the data stream and is closed by an EOF (end of frame) pattern. These different phases of the transmit sequence may be monitored by watching *ModemStat*e of *PrimaryStatus-Register* (see 17.4.4).

Depending on the setting of bit *TxCRCEn* in the *ChannelRedundancy-Register* a CRC is calculated and appended to the data stream. The CRC is calculated according the settings in the *ChannelRedundancy Register*. Parity generation is handled according the settings in the *ChannelRedundancy-Register* (bits *ParityEn* and *ParityOdd)*.

17.4.1.3 Transmission of Bit Oriented Frames

The transmitter may be configured to send an incomplete last byte. To achieve this *TxLastBits* has to be set to a value unequal zero. This is shown in the figure below.

Figure 17-1: Transmitting Bit Oriented Frames

The figure shows the data stream if *ParityEn* is set in *ChannelRedundancy-Register*. All fully transmitted bytes are followed by a parity check bit, but the incomplete byte is not followed by a parity check bit. After transmission, *TxLastBits* is cleared automatically.

Note: If *TxLastBits* is not equal to zero CRC generation has to be disabled. This is done by clearing the bit *TxCRCEn* in the *ChannelRedundancy Register*.

17.4.1.4 Transmission of Frames with more than 64 Bytes

To generate frames with more than 64 bytes, the µ-Processor has to write data into the FIFO buffer while the *Transmit Command* is active. The state machine checks the FIFO status when it starts transmitting the last bit of the actual data stream (the check time is marked below with arrows).

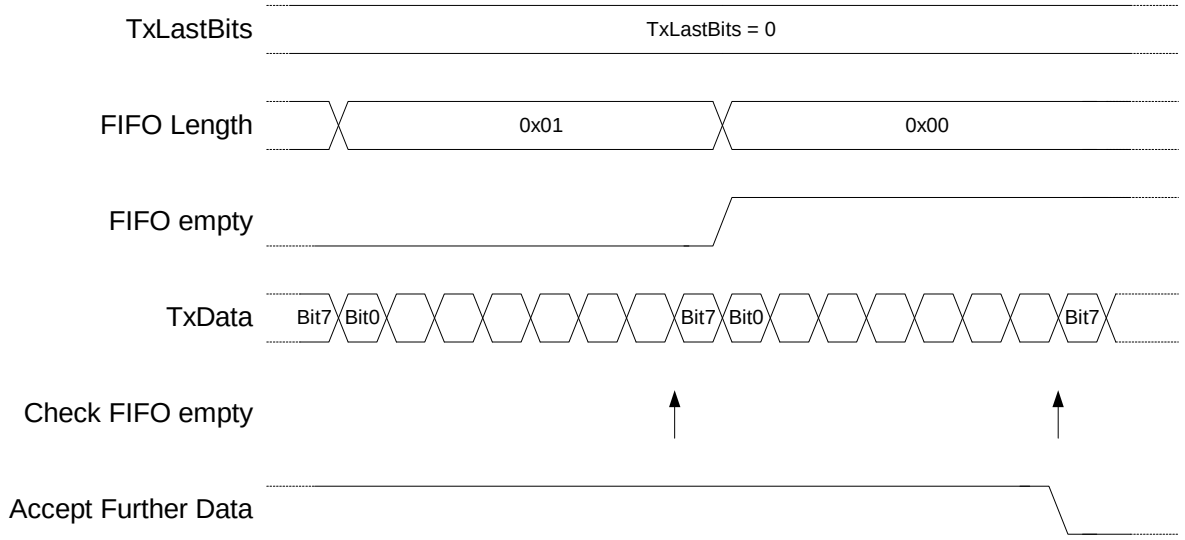


Figure 17-2: Timing for Transmitting Byte Oriented Frames

As long as the internal signal 'Accept Further Data' is 1 further data may be loaded to the FIFO. The MF RC530 appends this data to the data stream transmitted via the RF-interface.
If the internal signal 'Accept Further Data' is 0 the transmission will terminate. All data written into the FIFO buffer after 'Accept Further Data' went 0 will not be transmitted anymore, but remain in the FIFO buffer.

Remark: If parity generation is enabled (*ParityEn* bit is 1) the parity bit is the last bit to be transmitted. This delays the signal 'Accept Further Data' for one bit duration.

If *TxLastBits* is unequal zero the last byte is not transmitted completely, but only the number of bits set in *TxLastBits* are transmitted (starting with the least significant bit).
Thus, the internal state machine has to check the FIFO status at an earlier point in time (shown in the figure below).

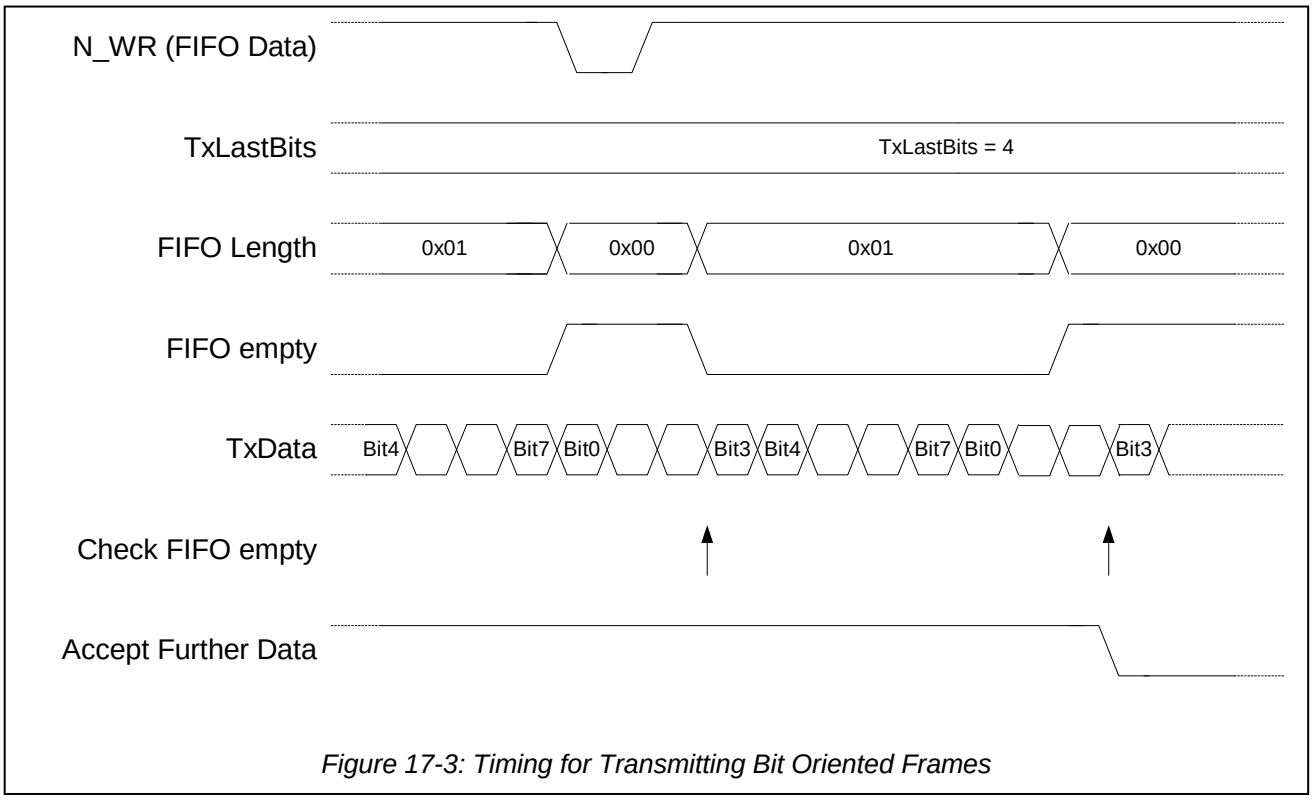


Figure 17-3: Timing for Transmitting Bit Oriented Frames

Since *TxLastBits* = 4 in this example, transmission stops after Bit 3 is transmitted. If configured, the frame is completed with an EOF.

The figure above also shows a write access to the *FIFOData Register* right before the FIFO's status is checked. This leads to 'FIFO empty' going to 0 again and therefore 'Accept Further Data' stays active. The new byte written is transmitted via the RF-interface.

'Accept Further Data' is changed only by the 'Check FIFO empty' function. This function verifies 'FIFO empty' one bit duration before the last expected bit transmission.

Frame Definition	**Verification at:**
8 Bit with Parity	8^{th} Bit
8 Bit without Parity	7^{th} Bit
x Bit without Parity	$(x-1)^{th}$ Bit

17.4.2 RECEIVE COMMAND 16_{HEX}

Command	$Code_{hex}$	Action	Arguments and Data	Returned Data
Receive	16	Activates Receiver Circuitry	-	Data Stream

The *Receive-Command* activates the receiver circuitry. All data received from the RF interface is returned via the FIFO buffer. The *Receive-Command* can be started either by the µ-Processor or automatically during execution of the *Transceive-Command*.

Note: This command may be used for test purposes only, since there is no timing relation to the *Transmit-Command*.

17.4.2.1 Working with the Receive Command

After starting the *Receive Command* the internal state machine decrements the value set in the *RxWait-Register* with every bit-clock. From 3 down to 1 the analog receiver circuitry is prepared and activated. When the counter reaches 0, the receiver starts monitoring the incoming signal at the RF-interface. If the signal strength reaches a level higher than the value set in the *MinLevel-Register* it finally starts decoding. The decoder stops, if no more signal can be detected on the receiver input pin Rx. The decoder indicates termination of operation by setting bit *RxIRq*.

The different phases of the receive sequence may be monitored by watching *ModemState* of the *PrimaryStatus-Register* (see 17.4.4).

Note: Since the counter values from 3 to 0 are necessary to initialise the analog receiver circuitry the minimum value for *RxWait* is 3.

17.4.2.2 RF-Channel Redundancy and Framing

The decoder expects a SOF pattern at the beginning of each data stream. If a SOF is detected, it activates the serial to parallel converter and gathers the incoming data bits. Every completed byte is forwarded to the FIFO. If an EOF pattern is detected or the signal strength falls below *MinLevel* set in the *RxThreshold Register*, the receiver and the decoder stop, the *Idle-Command* is entered and an appropriate response for the µ-Processor is generated (interrupt request activated, status flags set).

If bit *RxCRCEn* in the *ChannelRedundancy Register* is set a CRC block is expected. The CRC block may be one byte or two bytes according to bit *CRC8* in the *ChannelRedundancy Register*.

Remark: The received CRC block is not forwarded to the FIFO buffer if it is correct. This is realised by shifting the incoming data bytes through an internal buffer of either one or two bytes (depending on the defined CRC). The CRC block remains in this internal buffer. As a consequence all data bytes are available in the FIFO buffer one or two bytes delayed.
If the CRC fails all received bytes are forwarded to the FIFO buffer (including the faulty CRC itself).

If *ParityEn* is set in the *ChannelRedundancy Register* a parity bit is expected after each byte. If bit *ParityOdd* is set to 1, the expected parity is an odd parity, otherwise an even parity is expected.

17.4.2.3 Collision Detection

If more than one card is within the RF-field during the card selection phase, they will respond simultaneously. The MF RC530 supports the algorithm defined in ISO14443A to resolve data-collisions of cards serial numbers by doing the so-called anti-collision procedure. The basis for this is the ability to detect bit-collisions.

Bit-collision detection is supported by the used bit-coding scheme, namely the Manchester-coding. If in the first and second half-bit of a bit a sub-carrier modulation is detected, instead of forwarding a 1 or a 0 a bit collision will be signalled. To distinguish a 1 or 0-bit from a bit-collision, the MF RC530 uses the setting of *CollLevel*. If the amplitude of the half-bit with smaller amplitude is larger than defined by *CollLevel*, the MF RC530 indicates a bit-collision.

If a bit-collision is detected, the error flag *CollErr* is set. If a bit-collision is detected in a parity bit, the flag *ParityErr* is set indicating a parity error.

Independent from the detected collision the receiver continues receiving the incoming data stream. In case of a bit-collision, the decoder forwards 1 at the collision position.

Note: As an exception, if bit *ZeroAfterColl* is set, all bits received after the first bit-collision are forced to zero, regardless whether a bit-collision or an unequivocal state has been detected. This feature eases for the software to carry out the anti-collision procedure defined in ISO14443A.

When the first bit collision in a frame is detected, the bit position of this collision is stored in the *CollPos Register*.

The collision position follows the table below:

Collision in Bit	Value of CollPos
SOF	0
LSBit of LSByte	1
…	…
MSBit of LSByte	8
LSBit of second Byte	9
…	…
MSBit of second Byte	16
LSBit of third Byte	17
…	…

Table 17-2: Returned Values for Bit Collision Positions

The parity bits are not counted in *CollPos*, since a bit-collision in a parity bit per definition succeeds a bit-collision in the data bits. If a collision is detected in the SOF a frame error is reported and no data is forwarded to the FIFO buffer. In this case the receiver continues to monitor the incoming signal and generates the correct notifications to the µ-Processor when the ending of the faulty input stream is detected. This helps the µ-Processor to determine the time when it is allowed next to send anything to the card.

17.4.2.4 Receiving Bit Oriented Frames

The receiver can handle byte streams with incomplete bytes, resulting in bit oriented frames. To support this, the following values may be used:

- *RxAlign* selects a bit offset for the first incoming byte, e.g. if *RxAlign* is set to 3, the first 5 bits received are forwarded to the FIFO buffer. Further bits are packed into bytes and forwarded. After reception, *RxAlign* is cleared automatically.
  If *RxAlign* is set to zero, all incoming bits are packed into one byte.
- *RxLastBits* returns the number of bits valid in the last received byte, e.g. if *RxLastBits* evaluates to 5 at the end of the receiving command, the 5 least significant bits are valid.
  RxlastBits evaluates to zero if the last byte is complete.

RxLastBits is valid only, if no frame error is indicated by the flag *FrameErr*. If *RxAlign* is set to a value other than zero and also *ParityEn* is active, the first parity bit is not checked but ignored.

17.4.2.5 Communication Errors

The following table shows which event causes the setting of error flags:

Cause	**Bit, that is set**
Received data did not start with a SOF pattern.	FramingErr
The CRC block is not equal the expected value.	CRCErr
The received data is shorter than the CRC block.	CRCErr
The parity bit is not equal the expected value (e. g. a bit collision occurs when a parity is expected)	ParErr
A collision is detected.	CollErr

Table 17-3: Communication Error Table

17.4.3 TRANSCEIVE COMMAND $1E_{HEX}$

Command	$Code_{hex}$	Action	Arguments and Data	Returned Data
Transceive	1E	Transmits data from FIFO buffer to the card and then activates automatically the receiver	Data Stream	Data Stream

The *Transceive-Command* first executes the *Transmit-Command* (see 17.4.1) and then automatically starts the *Receive-Command* (see 17.4.2). All data that shall be transmitted is forwarded via the FIFO buffer and all data received is returned via the FIFO buffer. The *Transceive-Command* can be started only by the µ-Processor.

Note: To adjust the timing relation between transmitting and receiving, the *RxWait Register* is used to define the time delay from the last bit transmitted until the receiver is activated. Furthermore, the *BitPhase Register* determines the phase-shift between the transmitter and the receiver clock.

17.4.4 STATES OF THE CARD COMMUNICATION

The actual state of the transmitter and receiver state machine can be fetched from *ModemState* in the *PrimaryStatus Register*.

The assignment of *ModemState* to the internal action is shown in the following table:

ModemState	Name of State	Description
000	Idle	Neither the transmitter nor the receiver is in operation, since none of them is started or the transmitter has not got input data
001	TxSOF	Transmitting the ‘Start Of Frame’ Pattern
010	TxData	Transmitting data from the FIFO buffer (or redundancy check bits)
011	TxEOF	Transmitting the ‘End Of Frame’ Pattern
100	GoToRx1	Intermediate state passed, when receiver starts
	GoToRx2	Intermediate state passed, when receiver finishes
101	PrepareRx	Waiting until the time period selected in the *RxWait Register* has expired
110	AwaitingRx	Receiver activated; Awaiting an input signal at pin Rx
111	Receiving	Receiving data

Table 17-4: Meaning of ModemState

17.4.5 STATE DIAGRAM FOR THE CARD COMMUNICATION

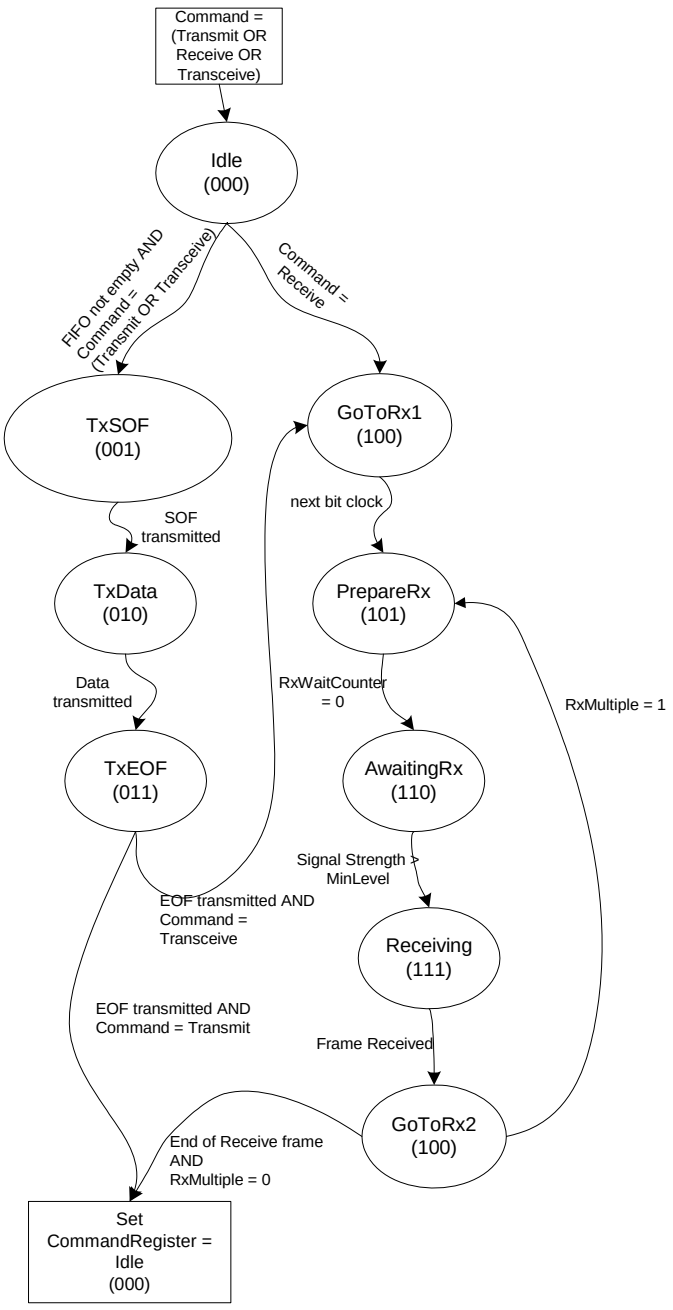


Figure 17-4: State Diagram: Card Communication

17.5 Commands to Access the E²PROM

17.5.1 WRITEE2 COMMAND 01_{HEX}

17.5.1.1 Overview

Command	$Code_{hex}$	Action	Arguments and Data passed via FIFO	Returned Data via FIFO
WriteE2	01	Get data from FIFO buffer and write it to the E²PROM	Start Address LSB Start Address MSB Data Byte Stream	-

The *WriteE2-Command* interprets the first two bytes in the FIFO buffer as E²PROM starting byte-address. Any further bytes are interpreted as data bytes and are programmed into the E²PROM, starting from the given E²PROM starting byte-address. This command does not return any data.
The *WriteE2-Command* can only be started by the µ-Processor. It will not stop automatically but has to be stopped explicitly by the µ-Processor by issuing the *Idle-Command*.

17.5.1.2 Programming Process

One byte up to 16 byte can be programmed into the EEPROM in one programming cycle. The time needed will be in any case about 5.8ms.

The state machine copies all data bytes prepared in the FIFO buffer to the E²PROM input buffer. The internal E²PROM input buffer is 16 byte long, which is equal the block size of the E²PROM. A programming cycle is started either if the last position of the E²PROM input buffer is written or if the last byte of the FIFO buffer has been fetched.

As long as there are unprocessed bytes in the FIFO buffer or the E²PROM programming cycle still is in progress, the flag *E2Ready* is 0. If all data from the FIFO buffer are programmed into the E²PROM, the flag *E2Ready* is set to1. Together with the rising edge of *E2Ready* the interrupt request flag *TxIRq* indicates a 1. This may be used to generate an interrupt when programming of all data is finished.

After the *E2Ready* bit is set to 1, the *WriteE2-Command* may be stopped by the µ-Processor by issuing the *Idle-Command*.

Note: During the E2PROM programming indicated by *E2Ready = 0* the WRITEE2 command cannot be stopped by any other command.

17.5.1.3 Timing Diagram

The following diagram shows programming of 5 bytes into the E²PROM:

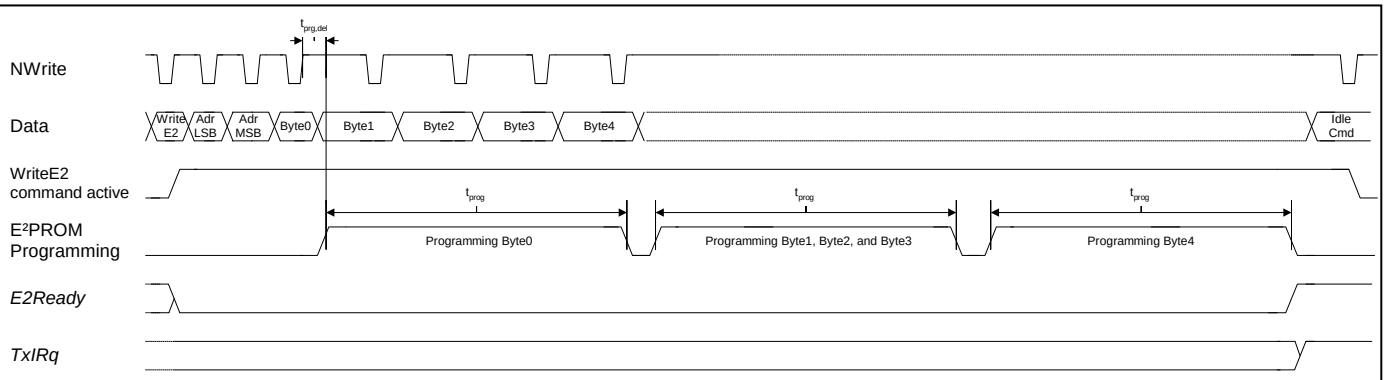


Figure 17-5: Timing Diagram for E²PROM programming

Explanation: It is assumed, that the MF RC530 finds and reads Byte 0 before the µ-Processor is able to write Byte 1 ($t_{prog,del}$ = 300 ns). This causes the MF RC530 to start the programming cycle, which needs about t_{prog} = 5.8 ms. In the meantime the µ-Processor stores Byte 1 to Byte 4 to the FIFO buffer. Assuming, that the E²PROM starting byte-address is e.g. $16C_{hex}$ then Byte 0 is stored exactly there. The MF RC530 copies the following data bytes into the E²PROM input buffer. Copying Byte 3, it detects, that this data byte has to be programmed at the E²PROM byte-address $16F_{hex}$. Since this is the end of the memory block, the MF RC530 automatically starts a programming cycle. In the next turn, Byte 4 will be programmed at the E²PROM byte-address 170_{hex}. Since this is the last data byte, the flags (*E2Ready* and *TxIRq*) that indicate the end of the E²PROM programming activity will be set.

Although all data has been programmed into the E2PROM, the MF RC530 stays in the *WriteE2-Command*. Writing further data to the FIFO would lead to further E²PROM programming, continuing at the E²PROM byte-address 171_{hex}. The command is stopped using the *Idle-Command*.

17.5.1.4 Error Flags for the WriteE2 Command

Programming is inhibited for the E²PROM blocks 0 (E²PROM's byte-address 00_{hex} to $0F_{hex}$). Programming to these addresses sets the flag *AccessErr*. No programming cycle is started. Addresses above $1FF_{hex}$ are taken modulo 200_{hex} (for the E²PROM memory organisation, refer to chapter 6.).

17.5.2 READE2 COMMAND 03_{HEX}

17.5.2.1 Overview

Command	$Code_{hex}$	Action	Arguments	Returned Data
ReadE2	03	Reads data from E²PROM and puts it to the FIFO buffer	Start Address LSB Start Address MSB Number of Data Bytes	Data Bytes

The *ReadE2-Command* interprets the first two bytes found in the FIFO buffer as E²PROM starting byte-address. The next byte specifies the number of data bytes that shall be returned.
When all three argument-bytes are available in the FIFO buffer, the specified number of data bytes is copied from the E²PROM into the FIFO buffer, starting from the given E²PROM starting byte-address.
The *ReadE2-Command* can be triggered only by the µ-Processor. It stops automatically when all data has been delivered.

17.5.2.2 Error Flags for the ReadE2 Command

Reading is inhibited for the E²PROM blocks 8_{hex} up to $1F_{hex}$ (key memory area). Reading from these addresses sets the flag *AccessErr* to 1. Addresses above $1FF_{hex}$ are taken modulo 200_{hex} (for the E²PROM memory organisation, refer to chapter 6).

17.6 Diverse Commands

17.6.1 LOADCONFIG COMMAND 07_{HEX}

17.6.1.1 Overview

Command	$Code_{hex}$	Action	Arguments and Data	Returned Data
LoadConfig	07	Reads data from E²PROM and initialises the registers	Start Address LSB Start Address MSB	-

The *LoadConfig-Command* interprets the first two bytes found in the FIFO buffer as E²PROM starting byte-address. When the two argument-bytes are available in the FIFO buffer, 32 bytes from the E²PROM are copied into the MF RC530 control and configuration registers, starting at the given E²PROM starting byte-address. The *LoadConfig-Command* can only be started by the µ-Processor. It stops automatically when all relevant registers have been copied.

17.6.1.2 Register Assignment

The 32 bytes of E²PROM content, beginning with the E²PROM starting byte-address, is written to the MF RC530 register 10_{hex} up to register $2F_{hex}$ (for the E²PROM memory organisation see also 6).

Note: The procedure for the register assignment is the same as it is for the Start Up Initialisation (see 11.3). The difference is, that the E²PROM starting byte-address for the Start Up Initialisation is fixed to 10_{hex} (Block 1, Byte 0). With the *LoadConfig-Command* it can be chosen.

17.6.1.3 Relevant Error Flags for the LoadConfig-Command

Valid E²PROM starting byte-addresses are in the range from 10_{hex} up to 60_{hex}.

Copying from block 8_{hex} up to $1F_{hex}$ (keys) is inhibited. Reading from these addresses sets the flag *AccessErr* to 1. Addresses above $1FF_{hex}$ are taken modulo 200_{hex} (for the E²PROM memory organisation refer to chapter 6).

17.6.2 CALCCRC COMMAND 12_{HEX}

17.6.2.1 Overview

Command	$Code_{hex}$	Action	Arguments and Data	Returned Data
CalcCRC	12	Activates the CRC-Coprocessor	Data Byte-Stream	-

The *CalcCRC-Command* takes all data from the FIFO buffer as input bytes for the CRC-Coprocessor. All data stored in the FIFO buffer before the command is started will be processed. This command does not return any data via the FIFO buffer, but the content of the CRC-register can be read back via the *CRCResultLSB-register* and the *CRCResultMSB-register*. The *CalcCRC-Command* can only be started by the µ-Processor. It does not stop automatically but has to be stopped explicitly by the µ-Processor with the *Idle-Command*. If the FIFO buffer is empty, the *CalcCRC-Command* waits for further input from the FIFO buffer.

17.6.2.2 CRC-Coprocessor Settings

For the CRC-Coprocessor the following parameters may be configured:

Parameter	Value	Bit	Register
CRC Register Length	8 Bit or 16 Bit CRC	CRC8	ChannelRedundancy
CRC Algorithm	Algorithm according ISO14443A or according ISO/IEC3309	CRC3309	ChannelRedundancy
CRC Preset Value	Any	CRCPresetLSB, CRCPresetMSB	CRCPresetLSB, CRCPresetMSB

Table 17-5: CRC-Coprocessor Parameters

The CRC polynomial for the 8-bit CRC is fixed to $x^8 + x^4 + x^3 + x^2 + 1$.
The CRC polynomial for the 16-bit CRC is fixed to $x^{16} + x^{12} + x^5 + 1$.

17.6.2.3 Status Flags of the CRC-Coprocessor

The status flag *CRCReady* indicates, that the CRC-Coprocessor has finished processing of all data bytes found in the FIFO buffer. With the *CRCReady* flag setting to 1, an interrupt is requested with *TxIRq* being set. This supports interrupt driven usage of the CRC-Coprocessor.

When *CRCReady* and *TxIRq* are set to 1, respectively, the content of the *CRCResultLSB*- and *CRCResultMSB-register* and the flag *CRCErr* is valid.

The *CRCResultLSB*- and *CRCResultMSB-register* hold the content of the CRC register, the *CRCErr* flag indicates CRC validity for the processed data.

17.7 Error Handling during Command Execution

If any error is detected during command execution, this is shown by setting the status flag *Err* in the *PrimaryStatus Register*. For information about the cause of the error, the µ-Processor may evaluate the status flags in the *ErrorFlag Register*.

Error Flag of the *ErrorFlag Register*	Related to Command
KeyErr	*LoadKeyE2, LoadKey*
AccessError	*WriteE2, ReadE2, LoadConfig*
FIFOOvl	*No specific commands*
CRCErr	*Receive, Transceive, CalcCRC*
FramingErr	*Receive, Transceive*
ParityErr	*Receive, Transceive*
CollErr	*Receive, Transceive*

Table 17-6: Error Flags Overview

17.8 MIFARE® Classic Security Commands

17.8.1 LOADKEYE2 COMMAND $0B_{HEX}$

17.8.1.1 Overview

Command	$Code_{hex}$	Action	Arguments and Data	Returned Data
LoadKeyE2	0B	Reads a key from the E²PROM and puts it into the internal key buffer	Start Address LSB Start Address MSB	-

The *LoadKeyE2-Command* interprets the first two bytes found in the FIFO buffer as E²PROM starting byte-address. The E²PROM bytes starting from the given starting byte-address are interpreted as key, stored in the correct key format as described in chapter 6.4.1. When all two argument-bytes are available in the FIFO buffer, the command execution starts. The *LoadKeyE2-Command* can be started only by the µ-Processor. It stops automatically after having copied the key from the E²PROM into the key buffer.

17.8.1.2 Relevant Error Flags for the LoadKeyE2-Command

If the key format is not correct (see chapter 6.4.1) an undefined value is copied into the key buffer and the flag *KeyError* is set.

17.8.2 LOADKEY COMMAND 19_{HEX}

17.8.2.1 Overview

Command	$Code_{hex}$	Action	Arguments and Data	Returned Data
LoadKey	19	Reads a key from the FIFO buffer and puts it into the key buffer	Byte0 (LSB) Byte1 … Byte10 Byte11 (MSB)	-

The *LoadKey-Command* interprets the first twelve bytes it finds in the FIFO buffer as key, stored in the correct key format as described in chapter 6.4.1.
When the twelve argument-bytes are available in the FIFO buffer they are checked and, if valid, are copied into the key buffer (see also 18.2).
The *LoadKey-Command* can only be started by the µ-Processor. It stops automatically after having copied the key from the FIFO buffer into the key buffer.

17.8.2.2 Relevant Error Flags for the LoadKey-Command

All bytes requested are copied from the FIFO buffer to the key buffer. If the key format is not correct (see chapter 6.4.1) an undefined value is copied into the key buffer and the flag *KeyError* is set.

17.8.3 AUTHENT1 COMMAND $0C_{HEX}$

17.8.3.1 Overview

Command	$Code_{hex}$	Action	Arguments and Data	Returned Data
Authent1	0C	Performs the first part of the Crypto1 (MIFARE® Classic) card authentication	Card Auth-Command Card Block Address Card Serial Number LSB Card Serial Number Byte1 Card Serial Number Byte2 Card Serial Number MSB	-

The *Authent1-Command* is a special *Transceive-Command*: it takes six argument bytes which are sent to the card. The card's response is not forwarded to the µ-Processor, but is used to check the authenticity of the card and to prove authenticity of the MF RC530 to the card.
The *Authent1-Command* can be triggered only by the µ-Processor. The sequence of states for this command is the same as for the *Transceive-Command* (see 17.4.3).

17.8.4 AUTHENT2 COMMAND 14_{HEX}

17.8.4.1 Overview

Command	$Code_{hex}$	Action	Arguments and Data	Returned Data
Authent2	14	Performs the second part of the card authentication using the Crypto1 algorithm.	-	-

The *Authent2-Command* is a special *Transceive-Command.* It does not need any argument byte but all necessary data which has to be sent to the card is assembled by the MF RC530 itself. The card response is not forwarded to the µ-Processor, but is used to check the authenticity of the card and to prove authenticity of the MF RC530 to the card.
The *Authent2-Command* can only be started by the µ-Processor. The logical sequence for this command is the same as for the Transceive-Command (see 17.4.3).

17.8.4.2 Effect of the Authent2-Command

If the *Authent2-Command* was successful, authenticity of card and MF RC530 is proved. In this case, the control bit *Crypto1On* is set automatically. When bit *Crypto1On* is set, all further card communication is done encrypted, using the Crypto1 security algorithm. If the Authent2-Command fails, bit *Crypto1On* is cleared.

Note: The flag *Crypto1On* can not be set by the µ-Processor but only through a successfully performed *Authent2-Command*. The µ-Processor may clear the bit *Crypto1On* to continue with plain card communication.

Note: The *Authent2-Command* has to be executed immediately after a successful *Authent1-Command* (see 17.8.3). Furthermore, the keys stored in the key buffer and those on the card have to match.

18 MIFARE® CLASSIC AUTHENTICATION AND CRYPTO1

18.1 General

The security algorithm implemented in MIFARE® Classic products is called Crypto1. It is based on a proprietary stream cipher with a key length of 48 bits. To access data of a MIFARE® Classic card, the knowledge of the according key is necessary. For successful card authentication and subsequent access to the card's data stored in the EEPROM, the correct key has to be available in the MF RC530. After a card is selected as defined in ISO14443A the user may continue with the MIFARE® Classic protocol. In this case it is mandatory to perform a card authentication. The Crypto1 authentication is a 3-pass authentication. This procedure is done automatically with the execution of *Authent1-* (see 17.8.3) and the *Authent2-Commands* (see 17.8.4). During the card authentication procedure, the security algorithm is initialised. The communication with a MIFARE® Classic card following a successful authentication is encrypted.

18.2 Crypto1 Key Handling

During the authentication command the MF RC530 reads the key from the internal key buffer. The key is always taken from the key buffer. Therefore, the commands for Crypto1 authentication do not require addressing of a key. The user has to ensure, that the correct key is prepared in the key buffer before the card authentication is triggered.

The key buffer can be loaded

- from the E²PROM with the *LoadKeyE2-Command* (see 17.8.1)
- directly from the µ-Processor via the FIFO-Buffer with the *LoadKey-Command* (see 17.8.2)

This is shown in the following figure:

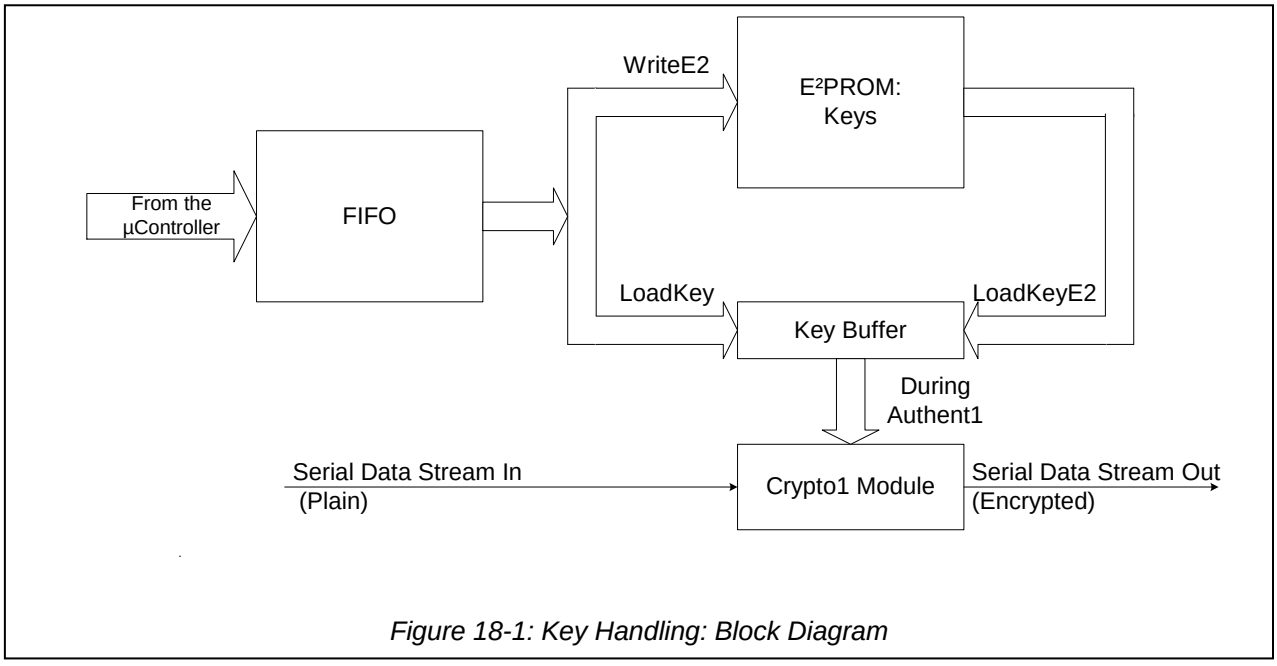


Figure 18-1: Key Handling: Block Diagram

18.3 Performing MIFARE® Classic Authentication

To enable authentication of MIFARE® Classic cards the Crypto1 security algorithm is implemented. To obtain valid authentication, the correct key has to be available in the key buffer of the MF RC530.

- ⇨ Step 1: Load the internal key buffer by means of the *LoadKeyE2-* (see 17.8.1) or the *LoadKey-Command* (see 17.8.2).
- ⇨ Step 2: Start the *Authent1-Command* (see 17.8.3). When finished, check the error flags to obtain the status of the command execution.
- ⇨ Step 3: Start the *Authent2-Command* (see 17.8.4). When finished, check the error flags and bit *Crypto1On* to obtain the status of the command execution.

19 TYPICAL APPLICATION

19.1 Circuit Diagram

The figure below shows a typical application, where the antenna is directly connected to the MF RC530:

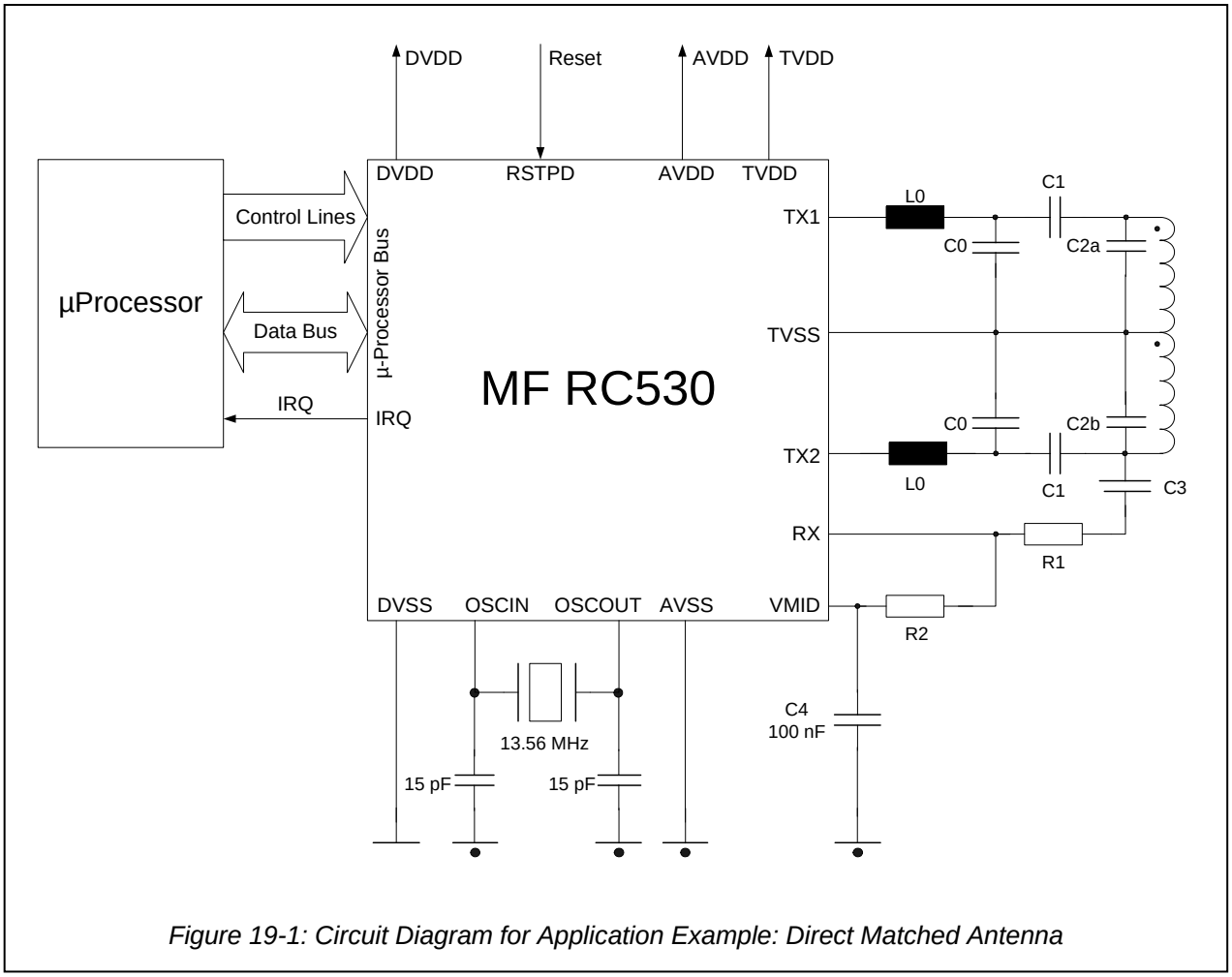


Figure 19-1: Circuit Diagram for Application Example: Direct Matched Antenna

19.2 Circuit Description

The matching circuit consists of an EMC low pass filter (L0 and C0), a matching circtuitry (C1 and C2), and a receiving circuit (R1, R2, C3 and C4), and the antenna itself.

For more detailed information about designing and tuning an antenna please refer to the Application Note

'MIFARE® and I CODE MICORE reader IC family; Directly Matched Antenna Design' and

'MIFARE® (14443A) 13,56 MHz RFID Proximity Antennas'.

19.2.1 EMC LOW PASS FILTER

The MIFARE® system operates at a frequency of 13.56 MHz. This frequency is generated by a quartz oscillator to clock the MF RC530 and is also the basis for driving the antenna with the 13.56 MHz energy carrier. This will not only cause emitted power at 13.56 MHz but will also emit power at higher harmonics. The international EMC regulations define the amplitude of the emitted power in a broad frequency range. Thus, an appropriate filtering of the output signal is necessary to fulfil these regulations.

A multi-layer board it is recommended to implement a low pass filter as shown in the circuit above. The low pass filter consists of the components L0 and C0. The recommended values are given in the above mentioned application notes.

Note: To achieve best performance all components shall have at least the quality of the recommended ones.

Note: The layout has a major influence on the overall performance of the filter.

19.2.2 ANTENNA MATCHING

Due to the impedance transformation of the given low pass filter, the antenna coil has to be matched to a certain impedance. The matching elements C1 and C2 can be estimated and have to be fine tuned depending on the design of the antenna coil.

The correct impedance matching is important to provide the optimum performance. The overall Quality factor has to be considered to guarantee a proper ISO14443 communication scheme. Environmental influences have to considered as well as common EMC design rules.

For details refer to the above mentioned application notes.

Note: Do not exceed the current limits I_{TVDD}, otherwise the chip might be destroyed.

Note: The overall 13.56MHz RFID proximity antenna design with the MF RC500 chip is straight forward and doesn't require a special RF-know how. However, all relevant parameters have to be considered to guarantee an overall optimum performance together with international EMC compliance.

19.2.3 RECEIVING CIRCUIT

The internal receiving concept of the MF RC530 makes use of both side-bands of the sub-carrier load modulation of the card response. No external filtering is required.

It is recommended to use the internally generated VMID potential as the input potential of pin RX. This DC voltage level of VMID has to be coupled to the Rx-pin via R2. To provide a stable DC reference voltage a capacitance C4 has to be connected between VMID and ground.

Considering the (AC) voltage limits at the Rx-pin the AC voltage divider of R1 + C3 and R2 has to be designed. Depending on the antenna coil design and the impedance matching the voltage at the antenna coil varies from antenna design to antenna design. Therefore the recommended way to design the receiving circuit is to use the given values for R1, R2, and C3 from the above mentioned application note, and adjust the voltage at the Rx-pin by varying R1 within the given limits.

Note: R2 is AC-wise connected to ground (via C4).

19.2.4 ANTENNA COIL

The precise calculation of the antenna coils' inductance is not practicable but the inductance can be **estimated** using the following formula. We recommend designing an antenna either with a circular or rectangular shape.

l_1 ..... Length of one turn of the conductor loop

D_1 ..... Diameter of the wire or width of the PCB conductor respectively

K..... Antenna Shape Factor (K = 1,07 for circular antennas and K = 1,47 for square antennas)

N_1 ..... Number of turns

ln ..... Natural logarithm function

The actual values of the **antenna inductance, resistance, and capacitance at 13.56 MHz** depend on various parameters like:

- antenna construction (Type of PCB)
- thickness of conductor
- distance between the windings
- shielding layer
- metal or ferrite in the near environment

Therefore a measurement of those parameters under real life conditions, or at least a rough measurement and a tuning procedure is recommended to guarantee the optimum performance. For details refer to the above mentioned application notes.

20 TEST SIGNALS

20.1 General

The MF RC530 allows different kind of signal measurements. These measurements can be used to check the internally generated and received signals using the possibilities of the serial signal switch as described in chapter 15.
Furthermore, with the MF RC530 the user may select internal analog signals to measure them at pin AUX and internal digital signals to observe them on pin MFOUT by register selections. These measurements can be helpful during the design-in phase to optimise the receiver's behaviour or for test purpose.

20.2 Measurements Using the Serial Signal Switch

Using the serial signal switch at pin MFOUT the user may observe data send to the card or data received from the card. The following tables give an overview of the different signals available.

SignalToMFOUT	MFOUTSelect	Signal routed to MFOUT pin
0	0	LOW
0	1	HIGH
0	2	Envelope
0	3	Transmit NRZ
0	4	Manchester with Subcarrier
0	5	Manchester
0	6	RFU
0	7	RFU
1	X	Digital Test signal

Table 20-1 Signal routed to MFOUT pin

Note: The routing of the Manchester and the Manchester with Subcarrier signal to the MFOUT is only possible at 106 kbaud datarate.

20.2.1 TX-CONTROL

The following plot shows the signal measured at MFOUT using the serial signal switch to control the data sent to the card .Setting the flag *MFOUTSelect* to 3 data sent to the card is shown NRZ coded. *MFOUTSelect* set to 2 shows the Miller coded signal.

The RFOut signal is measured directly on the antenna showing the pulse shape of the RF signal. For detail information concerning the pulse of the RF signal please refer to the application note *'MIFARE® Design of MF RC 500 Matching Circuits and Antennas'*

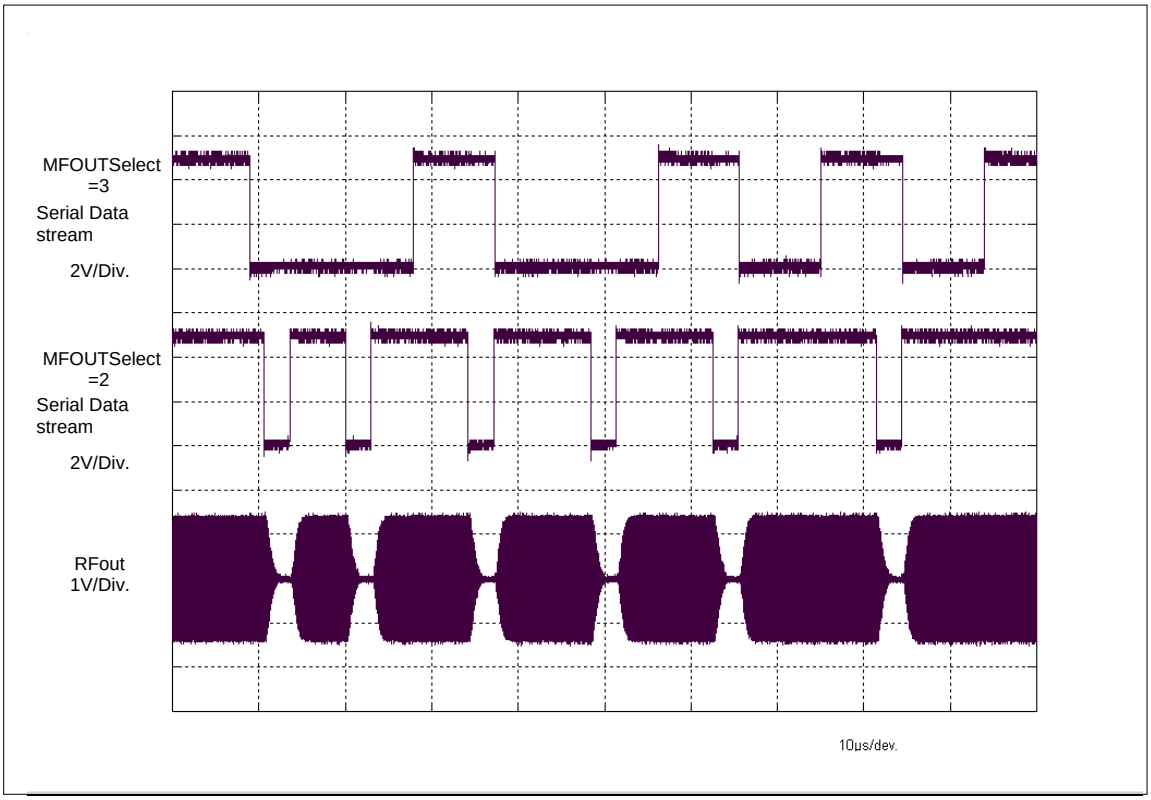


Figure 19 TX Control Signals

20.2.2 RX-CONTROL

The following plot shows the beginning of a cards answer to a request signal. The signal RF shows the RF voltage measured directly on the antenna so that the cards load modulation is visible. *MFOUTSelect* set to 4 shows the Manchester decoded signal with subcarrier. *MFOUTSelect* set to 5 shows the Manchester decoded signal.

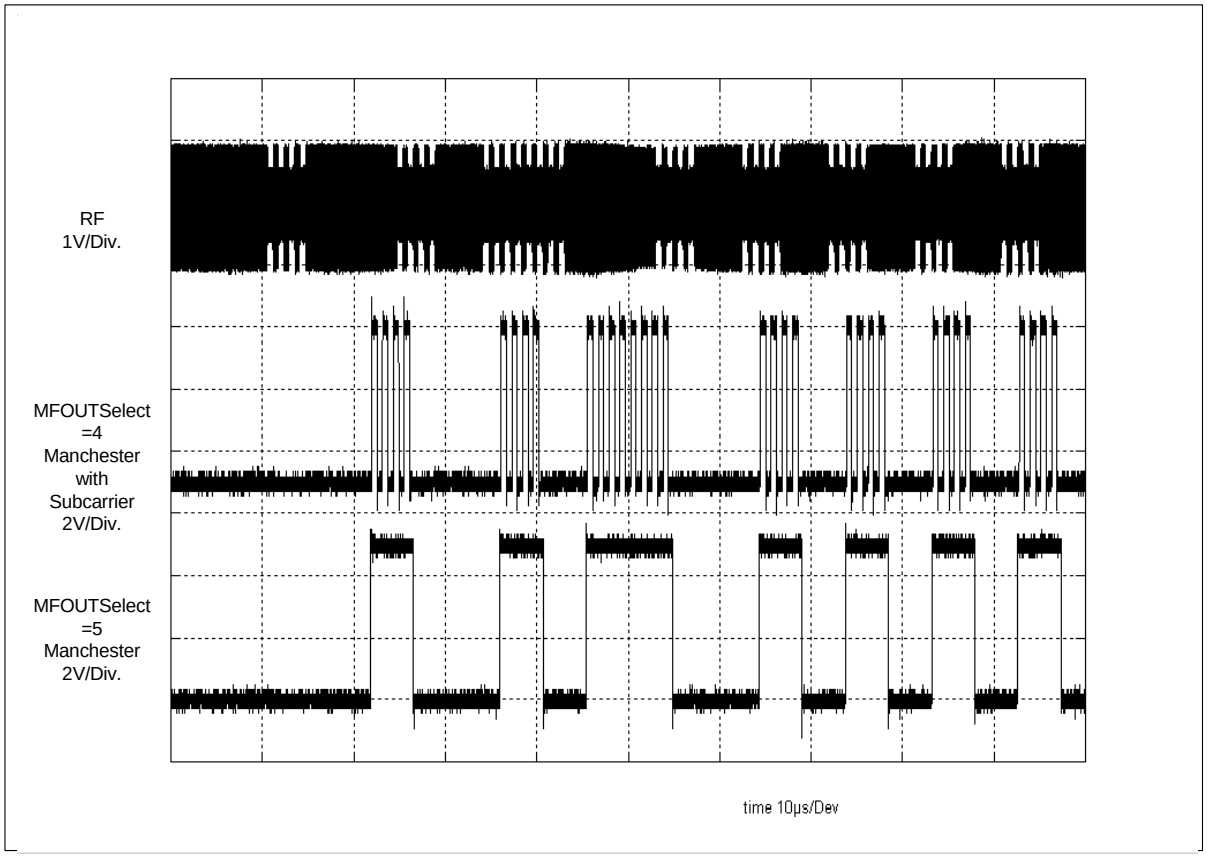


Figure 20 RX Control Signals

20.3 Analog Test-Signals

The analog test signals may be routed to pin AUX by selecting them with the register bits *TestAnaOutSel*.

Value	Signal Name	Description
0	V_{mid}	Voltage at internal node Vmid
1	$V_{bandgap}$	Internal reference voltage generated by the band gap.
2	$V_{RxFollI}$	Output signal from the demodulator using the I-clock.
3	$V_{RxFollQ}$	Output signal from the demodulator using the Q-clock.
4	V_{RxAmpI}	I-channel subcarrier signal amplified and filtered.
5	V_{RxAmpQ}	Q-channel subcarrier signal amplified and filtered.
6	V_{CorrNI}	Output signal of N-channel correlator fed by the I-channel subcarrier signal.
7	V_{CorrNQ}	Output signal of N-channel correlator fed by the Q-channel subcarrier signal.
8	V_{CorrDI}	Output signal of D-channel correlator fed by the I-channel subcarrier signal.
9	V_{CorrDQ}	Output signal of D-channel correlator fed by the Q-channel subcarrier signal.
A	V_{EvalL}	Evaluation signal from the left half bit.
B	V_{EvalR}	Evaluation signal from the right half bit.
C	V_{Temp}	Temperature voltage derived from band gap.
D	rfu	Reserved for future use
E	rfu	Reserved for future use
F	rfu	Reserved for future use

Table 20-2: Analog Test Signal Selection

20.4 Digital Test-Signals

Digital test signals may be routed to pin MFOUT by setting bit *SignalToMFOUT* to 1. A digital test signal may be selected via the register bits *TestDigiSignalSel* in Register *TestDigiSelect*.

The signals selected by a certain *TestDigiSignalSel* setting is shown in the table below:

TestDigiSignalSel	Signal Name	Description
$F4_{hex}$	s_data	Data received from the card.
$E4_{hex}$	s_valid	Shows with 1, that the signals s_data and s_coll are valid.
$D4_{hex}$	s_coll	Shows with 1, that a collision has been detected in the current bit.
$C4_{hex}$	s_clock	Internal serial clock: during transmission, this is the coder-clock and during reception this is the receiver clock.
$B5_{hex}$	rd_sync	Internal synchronised read signal (derived from the parallel µ-Processor interface).
$A5_{hex}$	wr_sync	Internal synchronised write signal (derived from the parallel µ-Processor interface).
96_{hex}	int_clock	Internal 13.56 MHz clock.
83_{hex}	BPSK_out	BPSK signal output
$E2_{hex}$	BPSK_sig	BPSK signal's amplitude detected
00_{hex}	no test signal	output as defined by *MFOUTSelect* are routed to pin MFOUT.

Table 20-3: Digital Test Signal Selection

If no test signals are used, the value for the *TestDigiSelect-Register* shall be 00_{hex}.

Note: All other values of *TestDigiSignalSel* are for production test purposes only.

20.5 Examples of Analog- and Digital Test Signals

Fig. 22 shows a MIFARE® Classic Card's answer to a request command using the Qclock receiving path.

RX –Reference is given to show the Manchester modulated signal at the RX pin. This signal is demodulated and amplified in the receiver circuitry VRXAmpQ shows the amplified side band signal having used the Q-Clock for demodulation. The signals VCorrDQ and VCorrNQ generated in the correlation circuitry are evaluated and digitised in the evaluation and digitizer circuitry. VEvalR and VEvalL show the evaluation signal of the right and left half bit. Finally, the digital test-signal S_data shows the received data which is send to the internal digital circuit and S_valid indicates that the received data stream is valid.

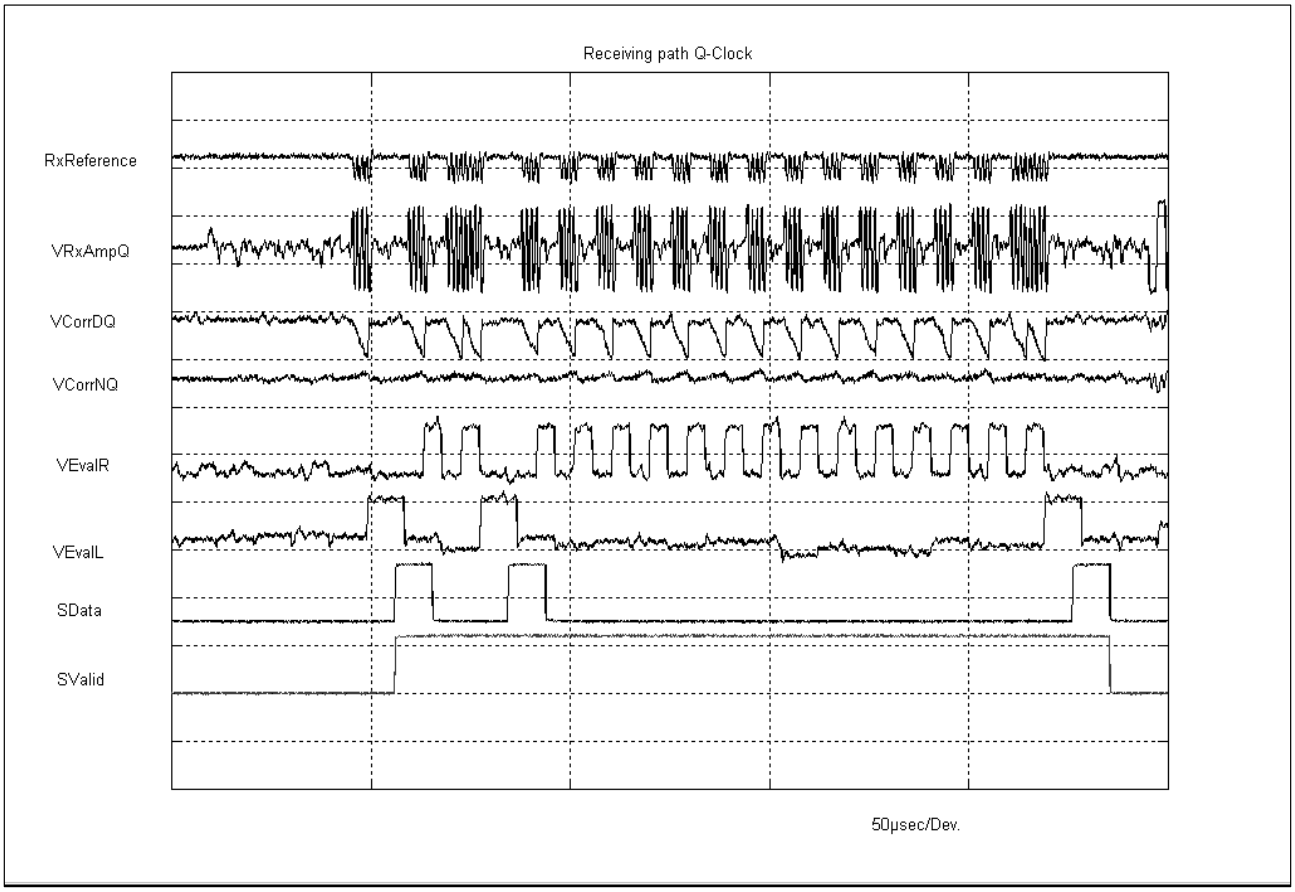


Figure 21. Receiving path Q-Clock

21 ELECTRICAL CHARACTERISTICS

21.1 Absolute Maximum Ratings

SYMBOL	PARAMETER	MIN	MAX	UNIT
$T_{amb,abs}$	Ambient or Storage Temperature Range	-40	+150	°C
DVDD AVDD TVDD	DC Supply Voltages	-0.5	6	V
$V_{in,abs}$	Absolute voltage on any digital pin to DVSS	-0.5	DVDD + 0.5	V
$V_{RX,abs}$	Absolute voltage on RX pin to AVSS	-0.5	AVDD + 0.5	V

Table 21-1: Absolute Maximum Ratings

21.2 Operating Condition Range

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
T_{amb}	Ambient Temperature	-	-25	+25	+85	°C
DVDD	Digital Supply Voltage	DVSS = AVSS = TVSS = 0V	3.0	3.3	3.6	V
			4.5	5.0	5.5	V
AVDD	Analog Supply Voltage	DVSS = AVSS = TVSS = 0V	4.5	5.0	5.5	V
TVDD	Transmitter Supply Voltage	DVSS = AVSS = TVSS = 0V	3.0	5.0	5.5	V

Table 21-2: Operating Condition Range

21.3 Current Consumption

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
I_{DVDD}	Digital Supply Current	Idle Command		8	11	mA
		Stand By Mode		3	5	mA
		Soft Power Down Mode		800	1000	µA
		Hard Power Down Mode		1	10	µA
I_{AVDD}	Analog Supply Current	Idle Command, Receiver On		25	40	mA
		Idle Command, Receiver Off		12	15	mA
		Stand By Mode		10	13	mA
		Soft Power Down Mode		1	10	µA
		Hard Power Down Mode		1	10	µA
I_{TVDD}	Transmitter Supply Current	Continuous Wave			150	mA
		TX1 and TX2 unconnected *TX1RFEn*, *TX2RFEn* = 1		5.5	7	mA
		TX1 and TX2 unconnected *TX1RFEn*, *TX2RFEn* = 0		65	130	µA

Table 21-3: Current Consumption

21.4 Pin Characteristics

21.4.1 INPUT PIN CHARACTERISTICS

Pins D0 to D7, A0, and A1 have TTL input characteristics and behave as defined in the following table.

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
I_{Leak}	Input Leakage Current		-1.0	+1.0	µA
V_T	Threshold	CMOS: DVDD < 3.6 V	0.35 DVDD	0.65 DVDD	V
		TTL: 4.5 < DVDD	0.8	2.0	V

Table 21-4: Standard Input Pin Characteristics

The digital input pins NCS, NWR, NRD, ALE, A2, and MFIN have Schmitt-Trigger characteristics, and behave as defined in the following table.

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
I_{Leak}	Input Leakage Current		-1.0	+1.0	µA
V_{T+}	Positive-Going Threshold	TTL: 4.5 < DVDD	1.4	2.0	V
		CMOS: DVDD < 3.6 V	0.65 DVDD	0.75 DVDD	V
V_{T-}	Negative-Going Threshold	TTL: 4.5 < DVDD	0.8	1.3	V
		CMOS: DVDD < 3.6 V	0.25 DVDD	0.4 DVDD	V

Table 21-5: Schmitt-Trigger Input Pin Characteristics

Pin RSTPD has Schmitt-Trigger CMOS characteristics. In addition, it is internally filtered with an RC-low-pass filter, which causes a relevant propagation delay for the reset signal:

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
I_{Leak}	Input Leakage Current		-1.0	+1.0	µA
V_{T+}	Positive-Going Threshold	CMOS: DVDD < 3.6 V	0.65 DVDD	0.75 DVDD	V
V_{T-}	Negative-Going Threshold	CMOS: DVDD < 3.6 V	0.25 DVDD	0.4 DVDD	V
$t_{RSTPD,p}$	Propagation Delay			20	µs

Table 21-6: RSTPD Input Pin Characteristics

The analog input pin RX has the following input capacitance:

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
C_{RX}	Input Capacitance			15	pF

Table 21-7: RX Input Capacitance

The analog input pin RX has the following input voltage range:

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
$VI_{N,RX}$	Dynamical Voltage input range	AVDD=5V, T=25°C	1,1V	4,4	V

Table 21-*8*: *RX Input voltage range*

21.4.2 DIGITAL OUTPUT PIN CHARACTERISTICS

Pins D0 to D7, MFOUT and IRQ have CMOS output characteristics and behave as defined in the following table.

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	Output Voltage HIGH	DVDD = 5 V, I_{OH} = -1 mA	2.4	4.9		V
		DVDD = 5 V, I_{OH} = -10 mA	2.4	4.2		V
V_{OL}	Output Voltage LOW	DVDD = 5 V, I_{OL} = 1 mA		25	400	mV
		DVDD = 5 V, I_{OL} = 10 mA		250	400	mV
I_O	Output Current source or sink	DVDD = 5 V			10	mA

Table 21-8: Digital Output Pin Characteristics

Note: IRQ pin may also be configured as open collector. In that case the values for V_{OH} do not apply.

21.4.3 ANTENNA DRIVER OUTPUT PIN CHARACTERISTICS

The source conductance of the antenna driver pins TX1 and TX2 for driving the HIGH level can be configured via *GsCfgCW* in the *CwConductance Register*, while their source conductance for driving the LOW level is constant.

For the default configuration, the output characteristic is specified below:

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	Output Voltage HIGH	TVDD = 5.0 V, I_{OL} = 20 mA		4.97		V
		TVDD = 5.0 V, I_{OL} = 100 mA		4.85		V
V_{OL}	Output Voltage LOW	TVDD = 5.0 V, I_{OL} = 20 mA		30		mV
		TVDD = 5.0 V, I_{OL} = 100 mA		150		mV
I_{TX}	Transmitter Output Current	Continuous Wave			200	mA_{peak}

Table 21-9: Antenna Driver Output Pin Characteristics

21.5 AC Electrical Characteristics

21.5.1 AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' for time. The other characters indicate the name of a signal or the logic state of that signal (depending on position):

Designation:	Signal:	Designation:	Logic Level:
A	address	H	HIGH
D	data	L	LOW
W	NWR or nWait	Z	high impedance
R	NRD or R/NW or nWrite	X	any level or data
L	ALE or AS	V	any valid signal or data
C	NCS	N	NSS
S	NDS or nDStrb and nAStrb, SCK		

Example: t_{AVLL} = time for address valid to ALE low

21.5.2 AC OPERATING SPECIFICATION

21.5.2.1 Bus Timing for Separated Read/Write Strobe

SYMBOL	PARAMETER	MIN	MAX	UNIT
t_{LHLL}	ALE pulse width	20		ns
t_{AVLL}	Multiplexed Address Bus valid to ALE low (Address Set Up Time)	15		ns
t_{LLAX}	Multiplexed Address Bus valid after ALE low (Address Hold Time)	8		ns
t_{LLWL}	ALE low to NWR, NRD low	15		ns
t_{CLWL}	NCS low to NRD, NWR low	0		ns
t_{WHCH}	NRD, NWR high to NCS high	0		ns
t_{RLDV}	NRD low to DATA valid		65	ns
t_{RHDZ}	NRD high to DATA high impedance		20	ns
t_{WLDV}	NWR low to DATA valid		35	ns
t_{WHDX}	DATA hold after NWR high (Data Hold Time)	8		ns
t_{WLWH}	NRD, NWR pulse width	65		ns
t_{AVWL}	Separated Address Bus valid to NRD, NWR low (Set Up Time)	30		ns
t_{WHAX}	Separated Address Bus valid after NWR high (Hold Time)	8		ns
t_{WHWL}	period between sequenced read / write accesses	150		ns

Table 21-10: Timing Specification for Separated Read/Write Strobe

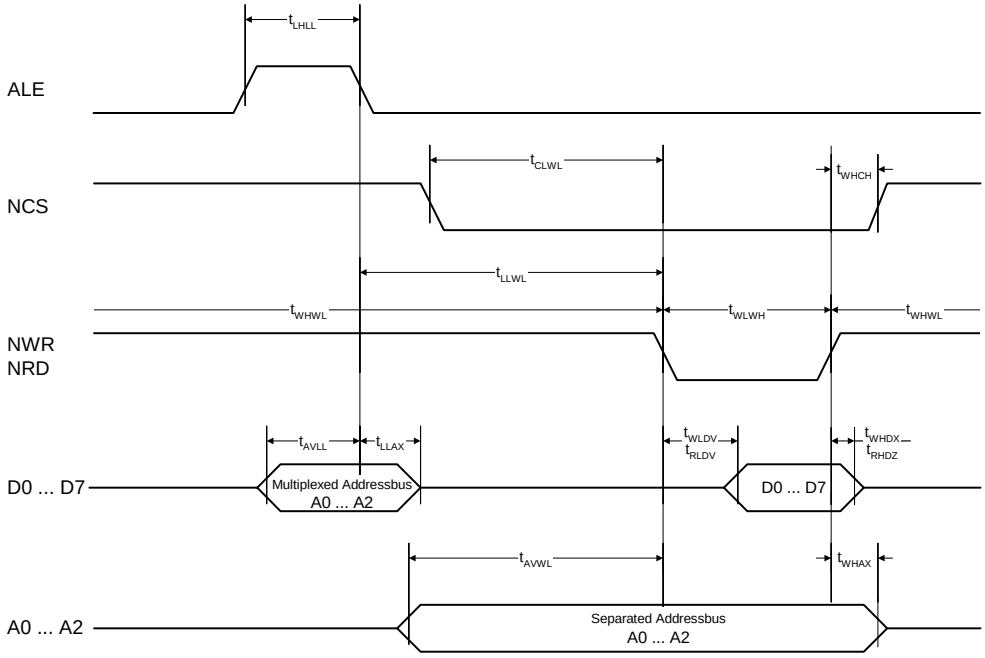


Figure 21-1: Timing Diagram for Separated Read/Write Strobe

Note: For separated address and data bus the signal ALE is not relevant and the multiplexed addresses on the data bus don't care.
For the multiplexed address and data bus the address lines A0 to A2 have to be connected as described in 4.3.

21.5.2.2 Bus Timing for Common Read/Write Strobe

SYMBOL	PARAMETER	MIN	MAX	UNIT
t_{LHLL}	AS pulse width	20		ns
t_{AVLL}	Multiplexed Address Bus valid to AS low (Address Set Up Time)	15		ns
t_{LLAX}	Multiplexed Address Bus valid after AS low (Address Hold Time)	8		ns
t_{LLSL}	AS low to NDS low	15		ns
t_{CLSL}	NCS low to NDS low	0		ns
t_{SHCH}	NDS high to NCS high	0		ns
$t_{SLDV,R}$	NDS low to DATA valid (for read cycle)		65	ns
t_{SHDZ}	NDS low to DATA high impedance (read cycle)		20	ns
$t_{SLDV,W}$	NDS low to DATA valid (for write cycle)		35	ns
t_{SHDX}	DATA hold after NDS high (write cycle, Hold Time)	8		ns
t_{SHRX}	R/NW hold after NDS high	8		ns
t_{SLSH}	NDS pulse width	65		ns
t_{AVSL}	Separated Address Bus valid to NDS low (Hold Time)	30		ns
t_{SHAX}	Separated Address Bus valid after NDS high (Set Up Time)	8		ns
t_{SHSL}	period between sequenced read/write accesses	150		ns
t_{RVSL}	R/NW valid to NDS low	8		ns

Table 21-11: Timing Specification for Common Read/Write Strobe

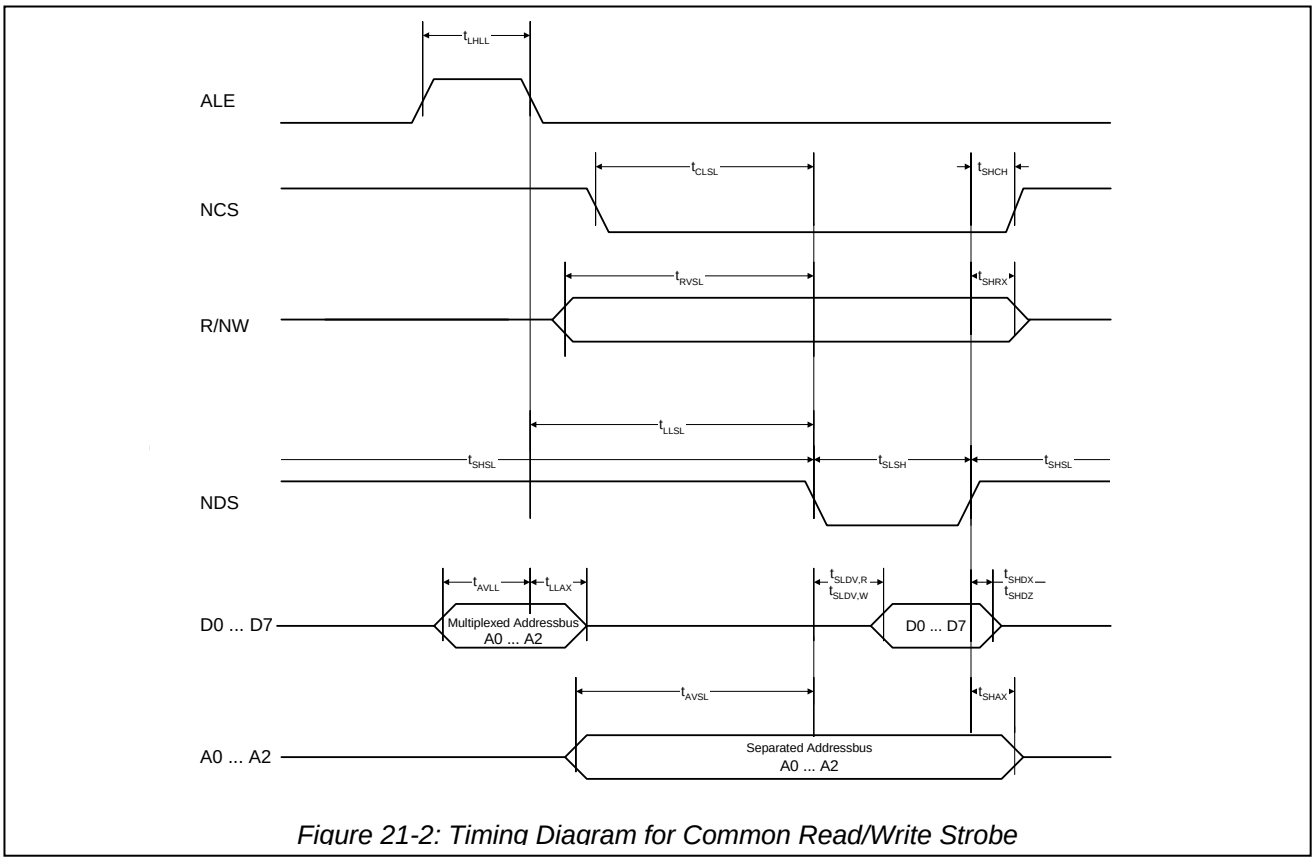


Figure 21-2: Timing Diagram for Common Read/Write Strobe

Note: For separated address and data bus the signal ALE is not relevant and the multiplexed addresses on the data bus don't care. For the multiplexed address and data bus the address lines A0 to A2 have to be connected as described in 4.3.

21.5.2.3 Bus Timing for EPP

SYMBOL	PARAMETER	MIN	MAX	UNIT
t_{LLLH}	nAStrb pulse width	20		ns
t_{AVLH}	Multiplexed Address Bus valid to nAStrb high (Set Up Time)	15		ns
t_{LHAX}	Multiplexed Address Bus valid after nAStrb high (Hold Time)	8		ns
t_{CLSL}	NCS low to nDStrb low	0		ns
t_{SHCH}	nDStrb high to NCS high	0		ns
$t_{SLDV,R}$	nDStrb low to DATA valid (read cycle)		65	ns
t_{SHDZ}	nDStrb low to DATA high impedance (read cycle)		20	ns
$t_{SLDV,W}$	nDStrb low to DATA valid (write cycle, Set up Time)		35	ns
t_{SHDX}	DATA hold after nDStrb high (write cycle, Hold Time)	8		ns
t_{SHRX}	nWrite hold after nDStrb high	8		ns
t_{SLSH}	nDStrb pulse width	65		ns
t_{RVSL}	nWrite valid to nDStrb low	8		ns
t_{SLWH}	nDStrb low to nWait high		75	ns
t_{SHWL}	nDStrb high to nWait low		75	ns

Table 21-12: Timing Specification for Common Read/Write Strobe

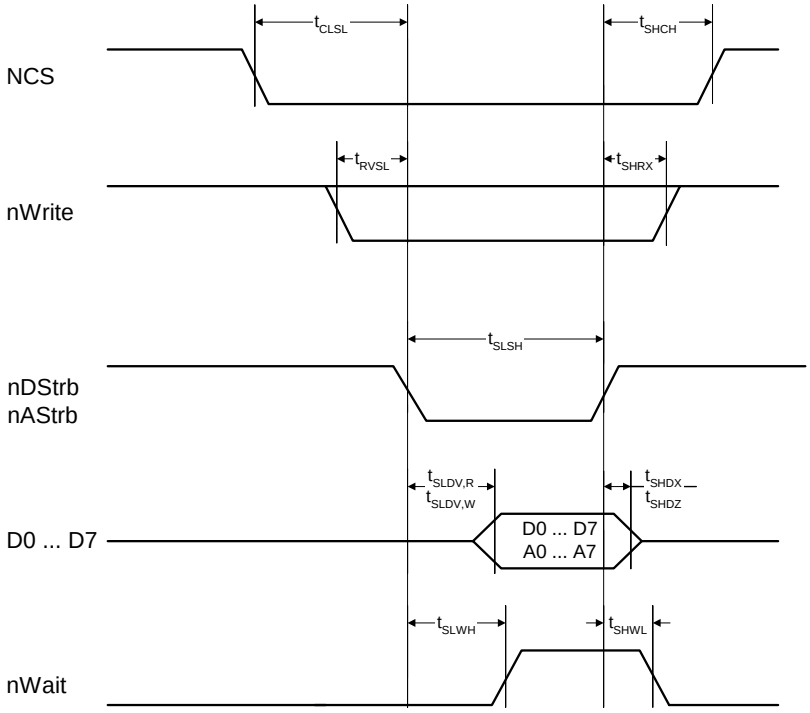


Figure 21-3: Timing Diagram for Common Read/Write Strobe

Remark: The figure does not distinguish between the Address Write Cycle and a Data Write Cycle. Take in account, that timings for the Address Write and Data Write Cycle are different. For the EPP-Mode the address lines A0 to A2 have to be connected as described in 4.3.

21.5.2.4 Timing for the SPI compatible interface

SYMBOL	PARAMETER	MIN	MAX	UNIT
t_{SCKL}	SCK low pulse width	100		ns
t_{SCKH}	SCK high pulse width	100		ns
t_{SHDX}	SCK high to data changes	20		ns
t_{DXSH}	data changes to SCK high	20		ns
t_{SLDX}	SCK low to data changes		15	ns
t_{SLNH}	SCK low to NSS high	20		ns

Table 21-13 Timing Specification for SPI

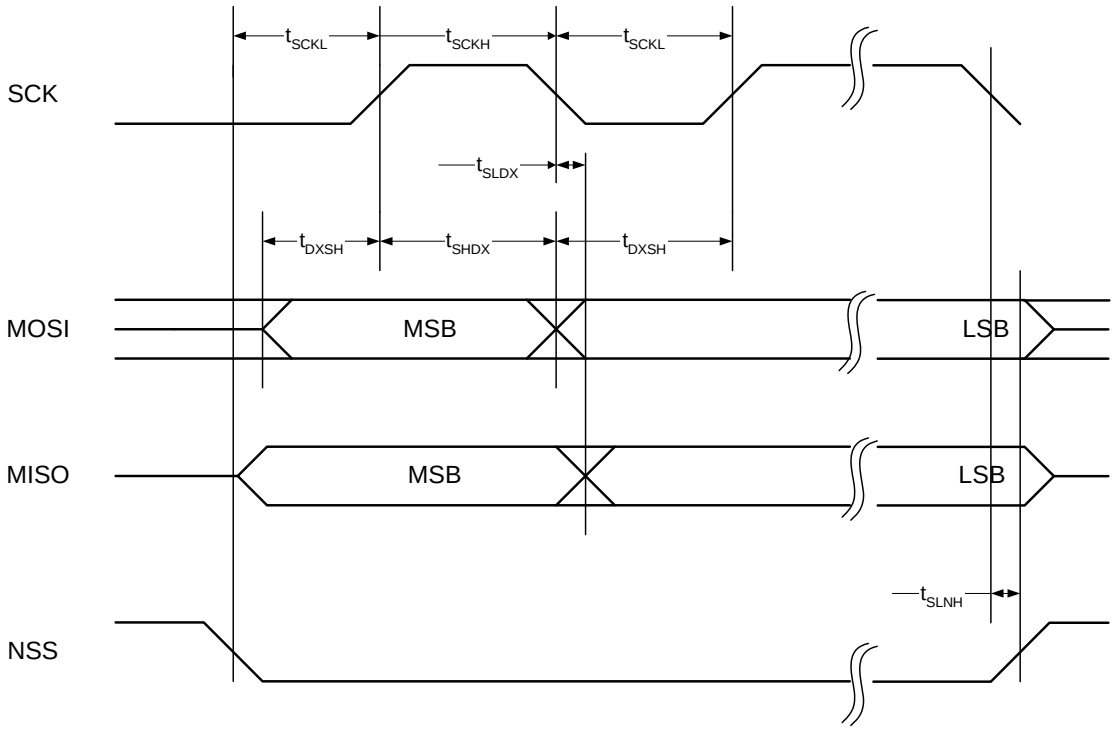


Figure 19 Timing Diagram for SPI

Note: To send more than bytes in one datastream the NSS signal has to low all the time.
To send more than one datastream NSS has to be set to HIGH level in between the datastreams.

21.5.3 CLOCK FREQUENCY

The clock input is pin 1, OSCIN.

PARAMETER	SYMBOL	MIN	TYP	MAX	UNIT
Clock Frequency (checked by the clock filter)	f_{OSCIN}		13.56		MHz
Duty Cycle of Clock Frequency	d_{FEC}	40	50	60	%
Jitter of Clock Edges	t_{jitter}			10	ps

The clock applied to the MF RC530 acts as time basis for the coder and decoder of the synchronous system. Therefore stability of clock frequency is an important factor for proper performance. To obtain highest performance, clock jitter shall be as small as possible. This is best achieved using the internal oscillator buffer with the recommended circuitry (see 12).

22 E²PROM CHARACTERISTICS

The E²PROM has a size of 32x16x8 = 4.096 bit.

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
$t_{EEEndurance}$	Data Endurance		100.000		erase/write cycles
$t_{EERetention}$	Data Retention	$T_{amb} \leq 55°C$	10		years
$t_{EEErase}$	Erase Time			2.9	ms
$t_{EEWrite}$	Write Time			2.9	ms

Table 22-1: E²PROM Characteristics

23 ESD SPECIFICATION

To ensure the usage of the MF RC 530 during production the ICs is specified as described in the following table.

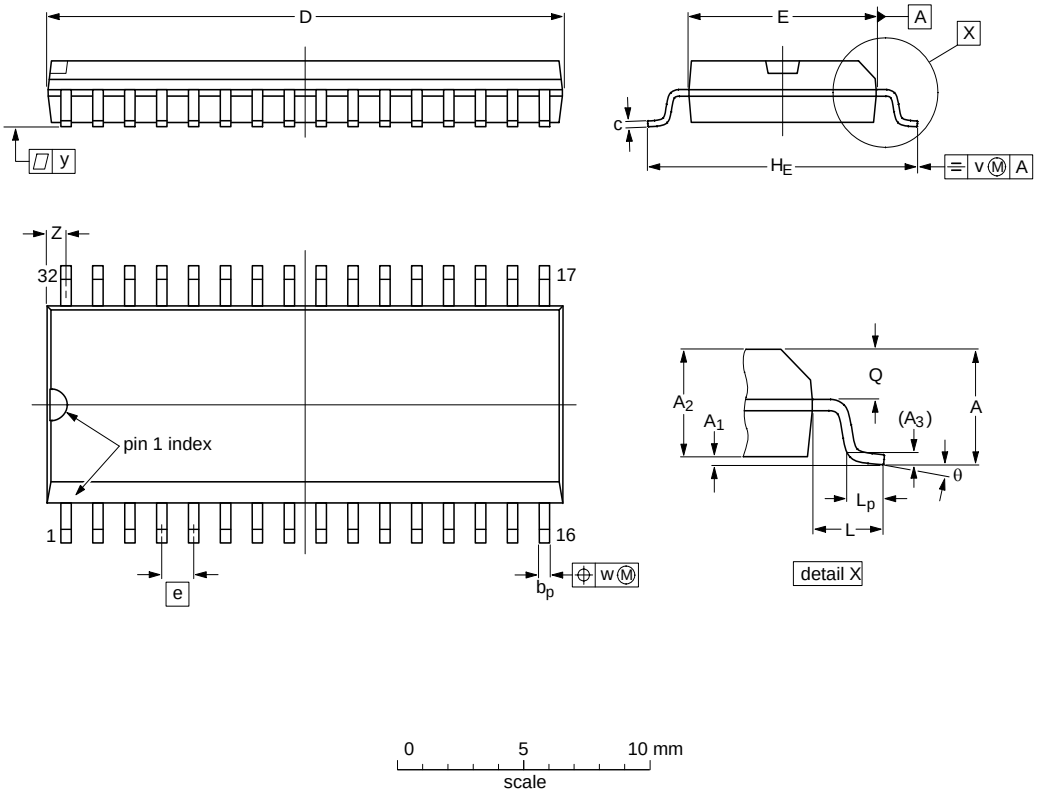
TEST	NAME	CONDITIONS	MAX
ESDH	ESD Susceptibility (Human body model)	1500 Ω, 100 pF	1 kV
ESDM	ESD Susceptibility (Machine model)	0.75 μH, 200 pF	100 V

Table 23-1. ESD Specification

24 PACKAGE OUTLINES

24.1 SO32

SO32: plastic small outline package; 32 leads; body width 7.5 mm **SOT287-1**



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	2.65	0.3 0.1	2.45 2.25	0.25	0.49 0.36	0.27 0.18	20.7 20.3	7.6 7.4	1.27	10.65 10.00	1.4	1.1 0.4	1.2 1.0	0.25	0.25	0.1	0.95 0.55	8^o 0^o
inches	0.10	0.012 0.004	0.096 0.086	0.01	0.02 0.01	0.011 0.007	0.81 0.80	0.30 0.29	0.050	0.419 0.394	0.055	0.043 0.016	0.047 0.039	0.01	0.01	0.004	0.037 0.022	

Note

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT287-1						~~95-01-25~~ 97-05-22

Figure 24-1: Outline and Dimension of MF RC530 in SO32

25 TERMS AND ABBREVIATIONS

Designation:	Description:
µ-Processor	Micro Processor
Crypto1	Crypto1 is the security algorithm of the MIFARE® Classic protocol.
E²PROM	Electrically Erasable Programmable Read Only Memory
EOF	End of Frame
FWT	Frame Waiting Time: maximum time delay between last bit transmitted by the reader and first bit received from the card's response.
Key	48 Bit Key for Crypto1 to authenticate a card sector.
MIFARE® Classic	A family of hard-wired logic contactless card ICs. The protocol of these cards is according to ISO 14443A part 2 and 3. On top they use a fixed set of commands including MIFARE® Classic security algorithm.
POR	Power On Reset: triggers a reset, caused by a rising edge on a supply pin.
ROM	Read Only Memory
SOF	Start of Frame

Definitions

Data sheet status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics section of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

26 LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so on their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

27 REVISION HISTORY

REVISION	DATE	CPCN	PAGE	DESCRIPTION
3.2	Dec 2005		88 89	Update Chapter 11.1 – include Note on reset phase duration Update Chapter 11.4 – detailed description initialization phase
3.1	May 2004			Update chapter “Current Consumption”
3.0	November 2002		-	Editorially revised version
2.0	December 2001		-	Editorially revised version
1.0	August2001		-	First published version

Table 27-1: Document Revision History

Philips Semiconductors - a worldwide company

Contact Information

For additional information please visit **http://www.semiconductors.philips.com**.Fax: **+31 40 27 24825**
For sales offices addresses send e-mail to: **sales.addresses@www.semiconductors.philips.com**.

SCA74

Let's make things better.

Philips Semiconductors

PHILIPS