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

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

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

Contact us

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

MLX90132

13.56MHz RFID / NFC Transceiver

Features and Benefits

- Conforms with ISO/IEC 18092 (NFC)
- Conforms with ISO/IEC 14443A and B,
- Conforms with ISO/IEC 15693
- Conforms with ISO/IEC 18000-3 mode 1
- High speed communication (up to 848kbit/s)
- Standard SPI/UART interfaces
- Built-in Field and TAG detectors

Application Examples

- NFC enabled car for access and start
- Combo NFC and Wireless Power Charging solutions
- NFC applications in Industrial area (e.g. White goods, security …)

Ordering Information

Part Code	Temperature Code	Package Code	Option Code	Packing Form Code
MLX90132	R (-40°C to 105°C)	LQ (Lead free QFN 5x5 32 leads)	AEA-000	RE
MLX90132	R (-40°C to 105°C)	LQ (Lead free QFN 5x5 32 leads)	AEA-000	TU
MLX90132	S (-20°C to 85°C)	LQ (Lead free QFN 5x5 32 leads)	AEA-000	RE
MLX90132	S (-20°C to 85°C)	LQ (Lead free QFN 5x5 32 leads)	AEA-000	TU

Functional Diagram

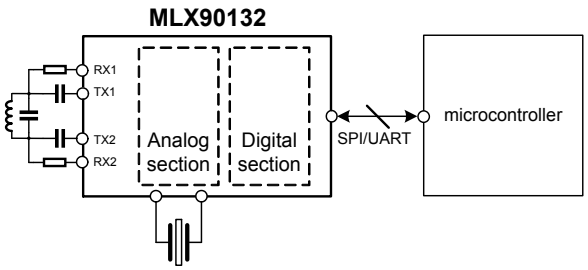


Figure 1: MLX90132 functional diagram

Description

The MLX90132 is a 13.56MHz, fully integrated, multi-protocol RFID/NFC transceiver IC. It has been designed to handle sub-carrier frequencies from 106 to 848 kHz and baud rates up to 848kbit/s.

The dual driver architecture of the MLX90132 requires minimal external support components and allows the transmitter to provide up to 300milliwatts RF power to an appropriate antenna load. This delivered power is suitable for most short to mid-range applications.

The MLX90132 embeds tag emulation functionality to support NFC Peer to Peer passive communication mode. Enhanced tag and field detection capabilities provide significant power consumption reduction in RFID reader configuration and in NFC mode.

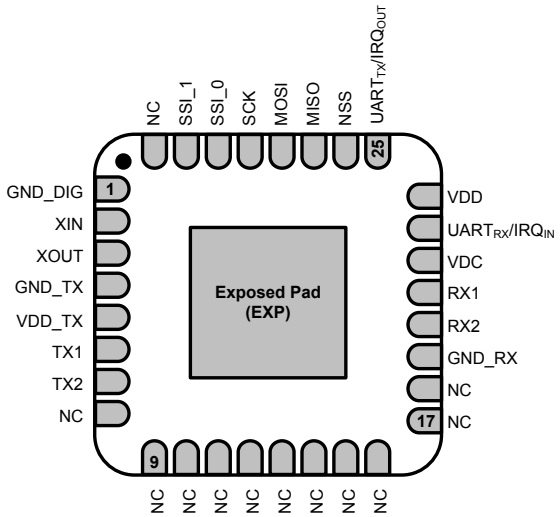
The digital section of the MLX90132 handles the low protocol layers from API to physical layer using advanced bit and frame encoding/decoding functions. It contains a digital demodulator based on sub-carrier detection and a programmable bit/symbol encoder/decoder. It also encodes and decodes the start and stop bits, parity bits, extra guard time (EGT), start and end of frame (SOF/EOF) and CRC.

Its 528 byte buffer handles an entire RFID frame. The SPI/UART communication ports guarantee easy interface with the majority of microcontrollers.

Table of Contents

1 Pin and signal descriptions

The device is packaged in a 32 pin lead free QFN package.



Pin	Symbol	Pin Type	Description
1	GND_dig	Supply	Ground (Digital)
2	XIN	Analog	Xtal oscillator input
3	XOUT	Analog	Xtal oscillator output
4	GND_TX	Supply	Ground (Drivers)
5	VDD_TX	Supply	Drivers Power Supply
6	TX1	Analog	Driver output_1
7	TX2	Analog	Driver output_2
19	GND_RX	Supply	Ground (analog)
20	RX2	Analog	Receiver input_2
21	RX1	Analog	Receiver input_1
22	VDC	Analog	Melexis Reserved
23	UART_RX / IRQ_in	Digital I	UART Receive pin/Interrupt input
24	VDD	Supply	Main Power Supply
25	UART_TX / IRQ_out	Digital O	UART Transmit pin/Interrupt output
26	NSS	Digital I	SPI Slave Select
27	MISO	Digital O	SPI data output
28	MOSI	Digital I	SPI data input
29	SCK	Digital I	SPI clock
30	SSI_0	Digital I	Select serial communication interface
31	SSI_1	Digital I	Must be set to GND
8-18, 32	NC		Not connected
EXP		Exposed Pad	Must be set to GND

Table 1: Pin definitions and descriptions

2 General Description

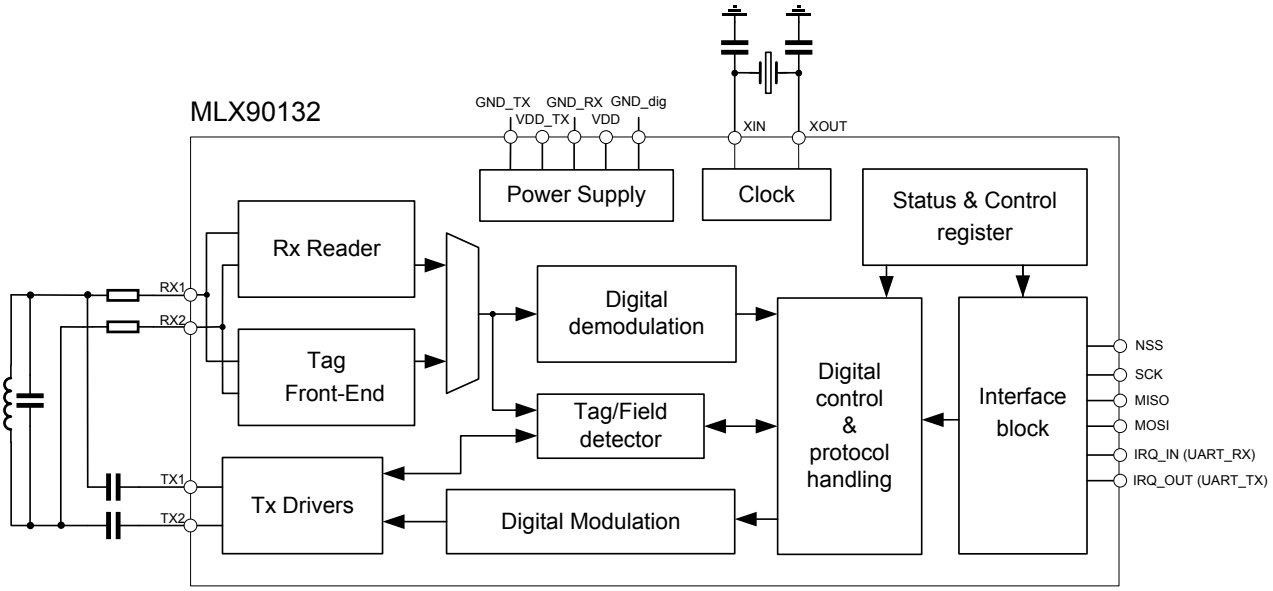


Figure 2: MLX90132 simplified block diagram

Power Supply

The MLX90132 is supplied with the 2 pins V_{DD} (supply of the digital and analog blocks) and V_{DD_TX} (direct supply of the TX Drivers), each requiring a nominal stable external power supply from 2.7 to 5.5 volt. Both pins V_{DD} and V_{DD_TX} are independent and could be connected together to the same power supply level or to different ones. The current drain depends on the antenna impedance and on the output matching network configuration.

Special attention should be paid to the filtering of V_{DD_TX}. Typically, a ferrite and a decoupling capacitor will be added close to the MLX90132 device.

TX Drivers

The transmission stage of the MLX90132 is composed of two differential outputs T_{X1} and T_{X2}, providing square waves with a frequency of f_{HFO} (typ. 13.56MHz), an amplitude of V_{DD_TX} and with a phase shift of 180 degrees. Each output is featuring an equivalent serial resistance R_{ON} which has to be taken into account when calculating the antenna matching network.

The transmission stage of the MLX90132 could be modulated using Amplitude Shift Keying (ASK) with a modulation index between 10% and 100%. The modulation index is automatically set with the selection of the protocol of communication, using the command Protocol select command (0x02). The modulation index could be fine adjusted by following the procedure described in the section Modifying internal settings for optimal performances.

In TAG emulation mode, the two outputs TX1 and TX2 are internally connected together, insuring a proper parallel resonance of the antenna. In this configuration, the two serial capacitors CS are put in parallel to the parallel capacitor CP. This operation is done automatically when selecting TAG emulation modes and should also be taken into account when defining an EMI filter for EMC considerations.

RX Reader

The reception stage of the MLX90132 is used in Reader mode to receive information from a transponder or an NFC/RFID device. This stage performs the analog demodulation using two internal diode detectors on RX1 and RX2.The information is then filtered with the appropriate bandwidth and finally digitized for further processing. The receiver inputs RX1 and RX2 are typically connected to the resonance point of the antenna, through two external attenuation resistors or capacitors to avoid saturation of the internal detector set to V_{RXMAX}. The complete receiver stage is automatically configured according to the protocol in use (Protocol select command (0x02)).

Tag Front-end

This block is enabled in Tag emulation mode and performs all operations related to Tag emulation functionality with low power consumption. The modulated information coming from an NFC/RFID device is demodulated through the two built-in detectors connected on RX1 and RX2, filtered with the appropriate bandwidth and finally digitized for further processing. The full settings of the Tag front-end stage are automatically set with the selected protocol using the Protocol select command (0x02). The load modulation used to send back the information in TAG emulation mode is also performed by the Tag front-end block. In this case, an internal resistor is connected between the two inputs RX1 and RX2, modifying the antenna load.

Digital control & protocol handling

This block is responsible for the control of the device, as well as the frame coding and decoding parts of the protocols supported by the MLX90132. The MLX90132 exchanges with the application microcontroller, pure payload information after adding/removing frame related information such as SOF, EOF, EGT … It can also be configured to calculate the CRC for each communication protocol.

Interface Block

The MLX90132 is addressed through SPI or UART (Reader mode only) interfaces with a specific and simple set of commands. The built-in 528 byte buffer allows minimum interaction with the application microcontroller. This reduces the burden of the microcontroller whose resources can be fully dedicated for the application.

Tag/Field Detector

This block manages the enhanced Tag and Field detection capabilities. It generates a detection signal that is available for the application microcontroller through the interrupt pin IRQ_OUT. It allows the use of the MLX90132 with low power consumption constraints.

Reference clock and internal oscillator

The built-in reference oscillator works with a reference crystal f_{XTAL} of 27.12MHz from which the internal nominal system clock frequency $\mathbf{f_{HFO}}$ of **13.56 MHz** is derived. An internal low frequency RC oscillator frequency $\mathbf{f_{LFO}}$ of **32 kHz** is used for low-power operating modes, for example to control the internal timings.

In TAG emulation mode the clock is recovered from the HF field, through the built-in Clock Recovery block. In case of field loss (e.g. during Reader modulation), an internal backup clock of ~10MHz is used instead.

Power management

The MLX90132 features 2 modes of operation (Active and Idle), subdivided in 6 different states of operation:

- **Hibernate**, the device typically consumes 1µA
- **Sleep**, the device typically consumes 20µA
- **TAG detection**, the device typically consumes 45µA.
- **TAG emulation**, the device typically consumes 2.5mA.
- **Ready** (RF field OFF), the device typically consumes 2.5mA.
- **Reader**, the consumption depends on the antenna load and on the operating conditions

3 Power Management and Operating modes

The MLX90132 features 2 main operating modes: Idle and Active, with 6 different states of operation, as described on the table below:

Mode	State	Description
Idle	Hibernate	Lowest power consumption, the MLX90132 wakes-up with low level pulse on IRQ_IN pin
	Sleep	Low Power consumption: Wake-up source to exit from this mode is configurable: - Timer - IRQ_in pin (low-level) - NSS pin (low-level) - Field detector
	Tag detection	Low power consumption: Tag detection feature, wake up source is configurable - Timer - IRQ_in pin (low level) - NSS pin (low level) - Tag detector (mandatory)
Active	Ready	High frequency oscillator (HFO) is running. In this mode the MLX90132 is in reader mode with its HF field turned OFF. The MLX90132 waits for a command from the external application, through the selected serial interface SPI or UART
	Reader	High frequency oscillator (HFO) is running. In this mode the MLX90132 is selected in reader mode with its HF field set ON. The MLX90132 is able to receive and execute commands through the selected serial interface SPI or UART and is able to communicate with transponders and NFC devices, according to the selected protocol. In Reader mode, the command “SendRecv” is used to send and receive information from an NFC/RFID transponder or devices
	TAG Emulation	High frequency oscillator (HFO) is running. In this mode the MLX90132 is selected in Tag emulation mode with its HF field set OFF. The MLX90132 is able to receive and execute commands through the serial interface SPI and is able to communicate with an NFC/RFID reader, according to the selected protocol. In TAG emulation mode, the commands “Listen” and “Send” will be used to respectively receive/send the information from/to an NFC/RFID reader. The information is returned to the NFC/RFID reader by using load modulation method

Table 2: MLX90132 Operating modes & States

Entering in Hibernate, Sleep and Tag detector states is done with the Idle command (0x07). As soon as one of these states is activated, an appropriate source signal is required to wake-up the device (see description above). The wake-up time from Sleep or Hibernate to Ready state is typically of 2ms. This time is mainly due to settling time of XTAL oscillator (HFO).

In Reader state, the MLX90132 is able to communicate with Transponder (TAG). In TAG emulation state, the MLX90132 is able to communicate with a reader by emulating a Transponder. Both states could be entered using the Protocol select command (0x02). In Ready state, the MLX90132 is fully enabled but waiting for the required command to enter either the Reader or the TAG Emulation state, without settling time penalty.

Please note the IDLE mode could be entered directly from Reader/Tag emulation state by sending the Idle command (0x07), no need to return to READY state to access the IDLE mode.

The command Protocol select command (0x02) with the option field OFF is used to return from Reader/Tag emulation state to Ready state.

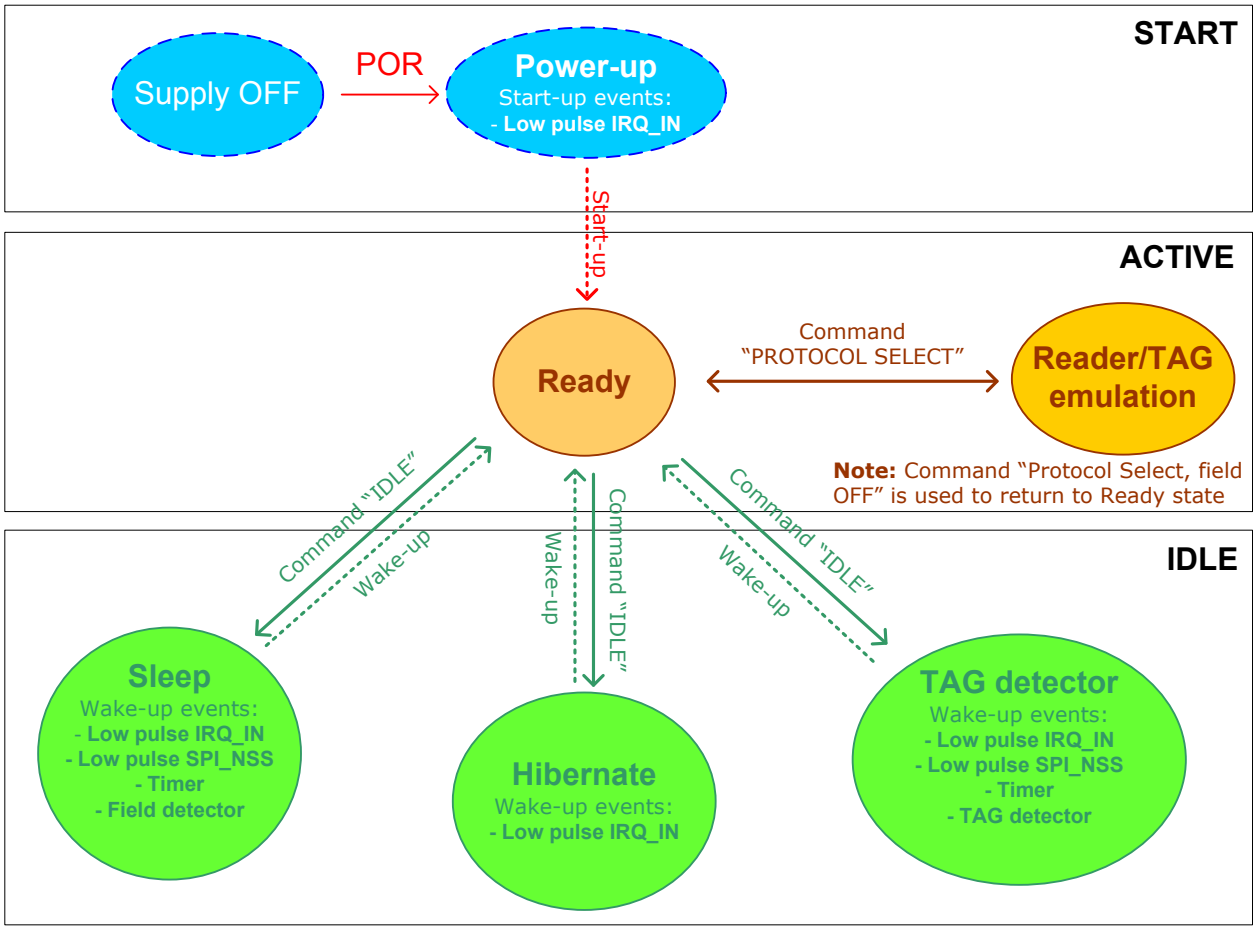


Figure 3: MLX90132 Power modes transitions

4 Start-up sequence

Once powered-up, the MLX90132 waits for a low pulse on the pin IRQ_IN (greater than 10µs) before automatically selecting the external interface (SPI or UART) and entering Ready state after a delay of approximately 2ms.

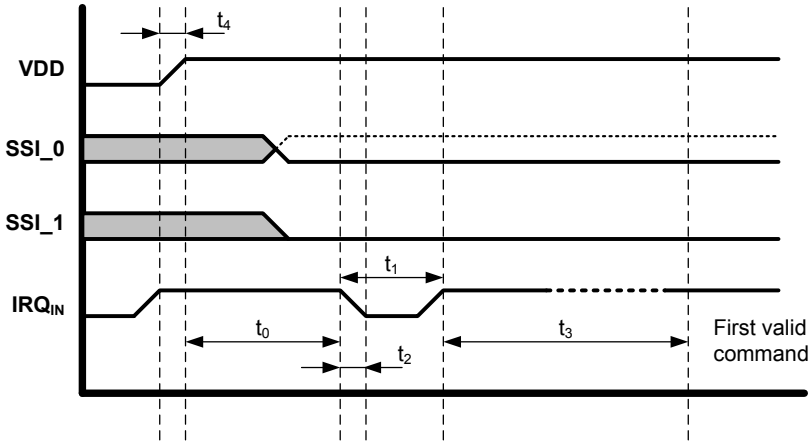


Figure 4: MLX90132 operating states transition

Figure 4 above shows the power-up sequence for a MLX90132 device where:

- t_0 is the initial wake-up delay[1)] — 100µs (minimum)
- t_1 is the minimum pulse width in IRQ_{IN} pin[1)] — 10µs (minimum)
- t_2 is the delay for the serial interface selection[1)] — 250ns (typical)
- t_3 is the delay before the MLX90132 could accept commands[1)] — 2ms (minimum)
- t_4 is the $_{VDD}$ ramp-up time[1)] — 10ms (maximum)

1) Value specified by design

The following configuration at power on reset (POR) is required to select the communication interface to be used.

Interface/Pin	SSI_1	SSI_0
SPI	0	1
UART	0	0

Table 3: Selection of the serial communication interface

Notes:

- *The Serial Interface is selected after the following falling edge of the pin IRQ_IN when leaving from POR or Hibernate states.*
- *When the MLX90132 leaves the IDLE state following a UART_RX/IRQ_{IN} low level pulse, this pulse is NOT interpreted as the UART start bit character.*

5 Communication Interface & protocol

Whatever the communication protocol selected (SPI or UART), the principle of communication is always the same: The application sends a command to the MLX90132 and waits for the appropriate answer. A simple and specific set of commands allows the configuration and control of the MLX90132.

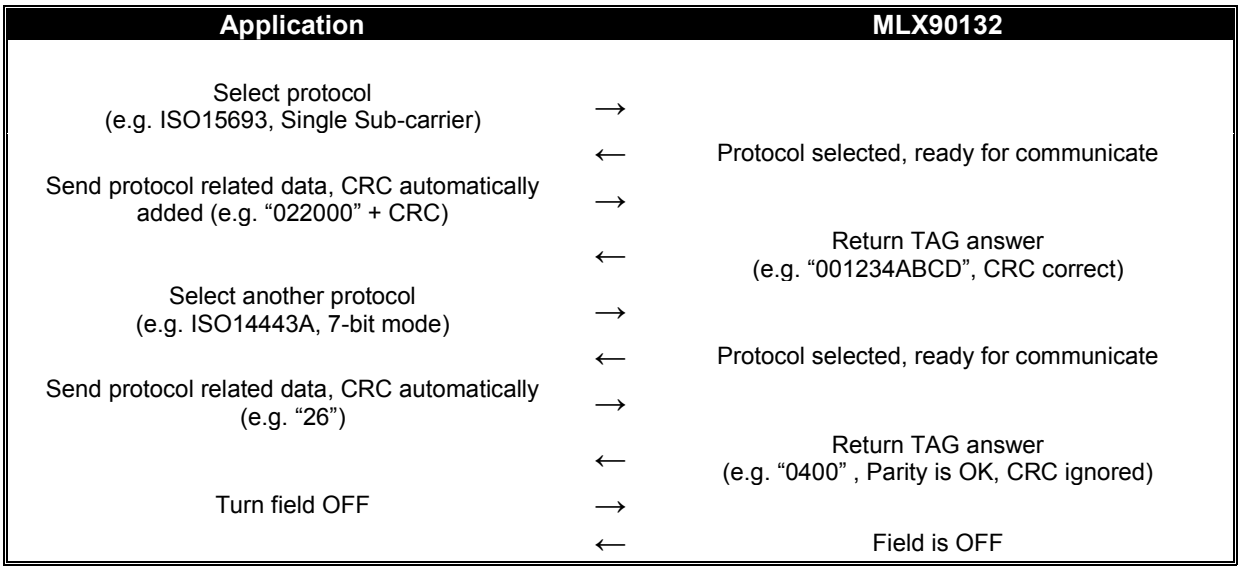


Figure 5: Example of communication with MLX90132

In order to start RFID communication, the application has to choose the protocol and specify some parameters, using the command Protocol select command (0x02). When the protocol is selected, the application sends data and parses response until the next protocol is selected or a specific parameter is changed.

5.1 UART

The default baud rate is 57.6kbps and the maximum allowed baud rate is 2 Mbps.

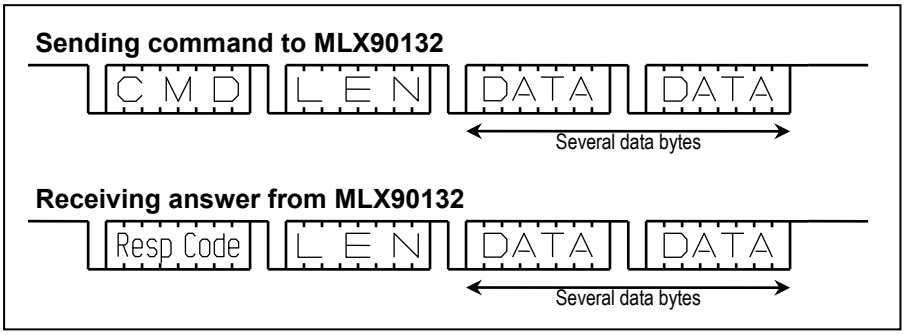


Figure 6: UART communication

Notes:

- Option “clock recovery” (“ClkRec” in Table 11) should not be used when UART interface is selected. Therefore the UART mode is not recommended for TAG emulation mode
- Length of data field can be zero, in this case no data is sent.

Warning: The UART communication is least significant bit (LSB) first.

5.2 SPI

5.2.1 Polling mode

In order to send commands and receive answers, the application software has to pass 3 stages:

1. Send the command to the MLX90132
2. Poll the MLX90132 until it is ready to transmit the response.
3. Read the response.

The application software should never read the MLX90132 without being sure that the MLX90132 is ready to send its response.

The maximum allowed communication speed is 2Mbps. Please note that the communication speed is limited to 1.5Mbps in case of TAG emulation mode with “clock recovery” option selected (“ClkRec” in Table 11, TAG/Card emulation mode).

A Control byte is used to specify the communication type and direction (see pictures below):
– 00: Send command to the MLX90132
– 11: Poll the MLX90132
– 10: Read data from the MLX90132
– 01: Reset the MLX90132

The SPI_NSS line is used to select a device on the common SPI bus; the SPI_NSS active level is LOW.

When the SPI_NSS line is inactive, all data sent by the application will be ignored and the SPI_MISO line will be set in high impedance state.

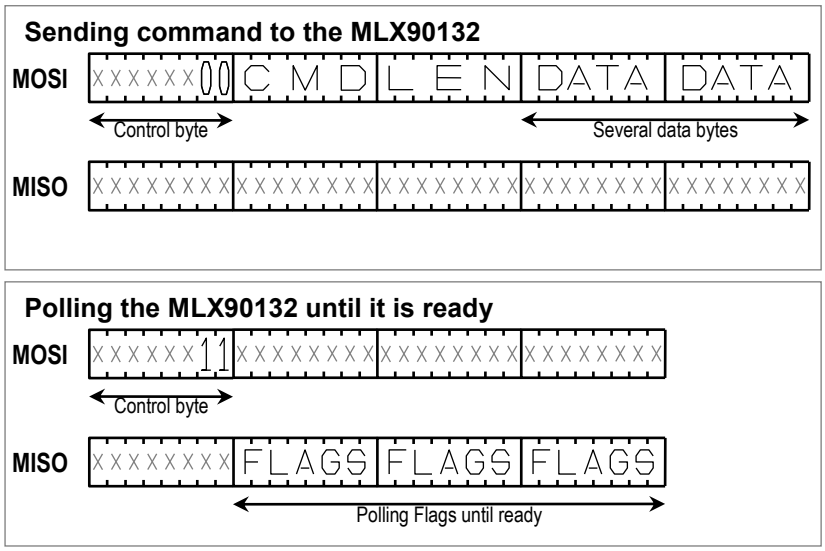


Figure 7: SPI communication, sending command & polling method

The following table shows the meaning of the flags returned by the MLX90132 device.

Bit	Description
[4:7]	RFU, will be set to “0000”
3	Data can be read from MLX90132 when set
2	Data can be sent to MLX90132 when set
[1:0]	MLX Reserved

Table 4: Interpretation of SPI flags

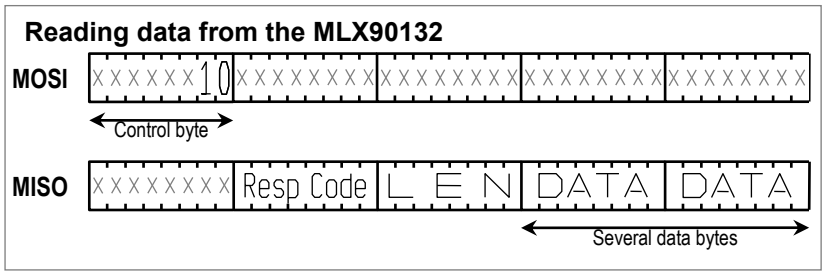


Figure 8: SPI communication, reading data from the MLX90132

Data must be sampled by the rising edge of the SPI_SCK signal.

'Sending', 'Polling' and 'Reading' commands must be separated by a high level of the SPI_NSS line.

For example, when the application needs to wait for data from the MLX90132, it sets to low the pin SPI_NSS and issues a 'Polling' command. By keeping the SPI_NSS "low", the application can continuously read the Flags waiting for the bit indicating that the MLX90132 is ready (the flags will be automatically updated, no need to send several polling commands).Then, the application has to set high the pin SPI_NSS to finish the polling sequence. The application puts low again the pin SPI_NSS to issue a 'Reading' command to read data. When all data is read, the application sets high the pin SPI_NSS to terminate the communication.

The MLX90132 can issue as many 'Polling' commands as necessary.

For example, the application sets low the pin SPI_NSS to issue a 'Polling' commands. If the MLX90132 is not ready, the application can put high the pin SPI_NSS and continue its operations. Then, as soon as the application is ready again, it sets low the pin SPI_NSS to issue a 'Polling' commands, to see if the MLX90132 is ready. These operations are not time critical which makes it easy to insert in the application flow.

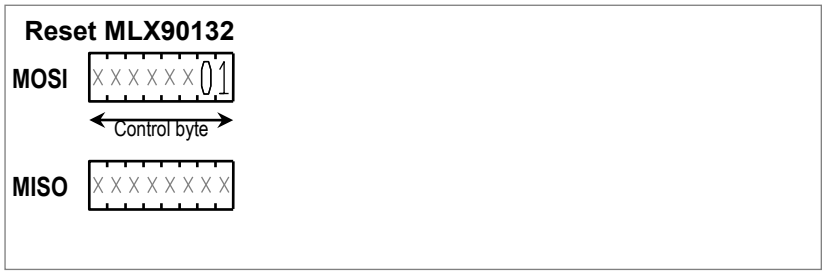


Figure 9: SPI communication reset the MLX90132

Control byte 0x01 resets the MLX90132 and places the device in Power-up state. A wake-up sequence is then necessary to start again the communication with the MLX90132.

Warning: The SPI communication is most significant bit (MSB) first.

5.2.2 IRQ mode

When the MLX90132 is configured to use the SPI serial interface, the pin IRQ_OUT is used to give additional information to the application. When the MLX90132 is ready to send back a reply it sends an Interrupt request by setting a low level on pin IRQ_OUT, which remains low until the application reads the data. The application can use the IRQ mode to skip the polling stage.

6 Commands

6.1 Command format

The structure of the command sent by the application is almost identical to the structure of the answer from the MLX90132, as shown below:

- **Command:** [CMD] + [LEN] + [DATA]
- **Answer:** [RESPCODE] + [LEN] + [DATA]

- [CMD] = Command (**1byte**)
- [LEN] = Length including only the field DATA, zero if no data sent (**1byte**)
- [RESPCODE] = Response code, depends on the command (**1byte**)
- [DATA] = Data information, depends on the command (**0 to 528bytes**)

6.2 List of commands

Code	Command	Description
0x01	IDN	Requests short information about device and its FW version
0x02	Protocol Select	Selects communication protocol and specifies some protocol-related parameters
0x03	Poll field	Returns the current value of the field detector flag (“FieldDet”)
0x04	SendRecv	Sends data using previously selected protocol and receives the response of the TAG.
0x05	Listen	Listens to the data using previously selected protocol.
0x06	Send	Sends data using previously selected protocol.
0x07	Idle	Switches device into Idle/Sleep/Hibernate mode and specifies which condition is used to exit from these modes
0x0A	BaudRate	Sets UART baud rate
0x0B	SubFreqRes	Gets the last value of sub-carrier frequency received during ISO/IEC18092 and NFC Tag Type 3 (Felica) communications
0x0D	AC-Filter	Activates/deactivates anti-collision filter
0x55	Echo	MLX90132 replies with an Echo of 0x55 to this command. **In this specific case, the command format is not respected as the data is only 0x55**
Other codes		MELEXIS reserved

Table 5: MLX90132 list of commands

6.3 IDN command (0x01)

The IDN command gives information about the MLX90132 and the internal firmware version

IDN**0x01**			
Direction	Data	Comment	Example
MCU – device	01	Command code	**0100**
	00	Length of data	
device - MCU	00	Result code	000F4E4643204653324A41535434002ACE: 4E4643204653324A4153543400= Device ID 2ACE= CRC of internal ROM
	<Len>	Length of data	
	<Device ID>	Data in ASCII format	
	<ROM CRC>	CRC calculated for ROM content	

Table 6: "IDN" command description

Note: *It takes about 6ms to calculate the CRC for the entire ROM. Application must allow sufficient time before waiting for an answer for this command.*

6.4 Protocol select command (0x02)

The "Protocol Select" command automatically configures the internal registers of the MLX90132 for the best communication performances. It also prepares the MLX90132 by automatically setting the HF field ON (except in TAG emulation state). The field will be automatically switched OFF either by sending a "Protocol select" command with "Field OFF", or when the MLX90132 returns to "Idle" mode using the "Idle" command or by selecting TAG emulation.

Protocol Select **0x02**			
Direction	Data	Comment	Example
MCU – device	02	Command code	Refer to examples in table: Table 8, below
	<Len>	Length of data	
	<Protocol>	Protocol codes (Reader) 00 = Field OFF 01 = ISO/IEC15693 02 = ISO/IEC14443-A / NFC-A 03 = ISO/IEC14443-B / NFC-B 04 = ISO/IEC18092 (212,424Kbps) / NFC-F	
		Protocol codes (TAG) 12 = ISO/IEC14443-A/ NFC-A 13 = ISO/IEC14443-B / NFC-B 14 = ISO/IEC18092 (212,424kbps)/ NFC-F	
	<Parameters>	Depends on protocol selected, refer to Table 8	
Device - MCU	00	Result code	**0000–**Protocol successfully selected
	00	Length of data	
Device - MCU	82	Error code	**8200-** Invalid command length
	00	Length of data	
Device - MCU	83	Error code	**8300 -** Invalid protocol
	00	Length of data	

Table 7: "Protocol select" command description

Parameter list for different protocols (**Reader**)					
Protocol (Reader)	Code	Parameters			Examples of commands
		Byte	Bit	Function	
Field OFF	00	0	7:0	RFU, set to '0'	**02020000**
ISO15693	01	0	7:6	RFU, set to '0'	**02020101** – Select ISO/IEC15693, SSC, 26kbps, modulation of 100%, CRC automatically added **02020107** – Select ISO/IEC15693, DSC, 26kbps, modulation 10%, CRC automatically added
			5:4	00 – 26kbps 01 – 52kbps 10 – 6kbps 11 – RFU	
			3	0 – Respect delay 312us 1 – Wait for SOF	
			2	0 - 100% modulation 1 – 10% modulation	
			1	0 – Single Sub-Carrier (SSC) 1 – Dual Sub-Carrier (DSC)	
			0	0 – No CRC added 1 – CRC auto. Added	
ISO14443A NFC-A	02	0	7:6	Transmission data rate 00 – 106kbps 01 – 212kbps 10 – 424kbps 11 – 847kbps	**02020200** – ISO/IEC14443A, 106kbps transmission & reception, Frame Delay Time (FDT) of 86/90µs
			5:4	Reception data rate 00 – 106Kbps 01 – 212Kbps 10 – 424Kbps 11 – 847Kbps	
			3:0	RFU, set to '0'	
		1	7:0	PP (max 14, i.e. 0x0E)	**Frame Delay Time (FDT) definition:** These 3 bytes are optional. When PP, MM and DD are not specified or set to 0x00, the default value corresponds to FDT of 86/90us, used during anti-collision process. Otherwise, the following formula applies: $FDT = \frac{2^{PP} \cdot (MM+1) \cdot (DD+128) \cdot 32}{13.56} [\mu s]$ If PP is defined, MM must be also set, but DD still remains optional
		2	7:0	MM (max 255, i.e. 0xFF)	
		3	7:0	DD (max 127, i.e. 0x7F)	
		4	7:0	N_{EMD}	Related to EMD algorithm, please refer to chapter Electromagnetic support (EMD)
		5	7:0	N_{EMDRES}	Related to EMD algorithm, please refer to chapter Electromagnetic support (EMD)

Table 8: Parameter values for "Protocol select" command (Reader)

Parameter list for different protocols (**Reader**)					
Protocol (Reader)	Code	Parameters			Examples of commands
		Byte	Bit	Function	
ISO14443B NFC-B	03	0	7:6	Transmission data rate 00 – 106kbps 01 – 212kbps 10 – 424kbps 11 – 847kbps	**02020301** – ISO/IEC14443B, 106kbps transmission & reception, Frame Waiting Time (FWT) of 302µs, CRC automatically added **020403010400** – ISO/IEC14443B, 106kbps transmission & reception, Frame Waiting Time (FWT) of 4.8ms, CRC automatically added
			5:4	Reception data rate 00 – 106kbps 01 – 212kbps 10 – 424kbps 11 – 847kbps	
			3:1	RFU, set to '0'	
			0	0 – No CRC added 1 – CRC auto. added	
		1	7:0	PP (max 14, i.e. 0x0E)	**Frame Waiting Time (FWT) definition:** These 2 bytes are optional. The default value corresponds to a FWT of 4949ms, answer to ATTRIB. $FWT = \frac{2^{PP} \cdot (MM+1) \cdot (DD+128) \cdot 32}{13.56} [\mu s]$ If PP is defined, MM must be also set, but DD still remains optional
		2	7:0	MM (max 255, i.e. 0xFF)	
		3	7:0	DD (max 127, i.e. 0x7F)	
		5:4	15:0	TTTT	Timing: TR0 = TTTT/13.56 us Coded with LSB first, default value 1023 = 0x3FF
		6	7:0	YY	Timing: Min_TR1 = 128 * YY / 13.56us. Default value: 0
		7	7:0	ZZ	Timing: Max_TR1 = 128 * ZZ / 13.56us. Default value:26 , i.e. 0x1A
		8	7:0	N_{EMD}	Related to EMD algorithm, please refer to chapter Electromagnetic support (EMD)
		9	7:0	N_{EMDRES}	Related to EMD algorithm, please refer to chapter Electromagnetic support (EMD)

Table 9: Parameter values for "Protocol select" command (Reader)

Parameter list for different protocols (**Reader**)					
Protocol (Reader)	Code	Parameters			Examples of commands
		Byte	Bit	Function	
ISO18092 (212,424Kb) NFC-F	04	0	7:6	Transmission data rate 00 – RFU 01 – 212kbps 10 – 424kbps 11 – RFU	**02020451** – ISO/IEC18092, 212kbps for transmission & reception, CRC automatically added Parameter 'Slot counter' is optional, the default value 00 (1 slot) will be used, if not present in the command. For command SDD (Single Device Detection), the bit 4 must be set to 0, In this case RWT is 2.4ms for the 1st slot and 1.2ms more for each following slot as specified in protocol ISO18092
			5:4	Reception data rate 00 – RFU 01 – 212Kbps 10 – 424Kbps 11 – RFU	
			3:1	RFU, set to '0'	
			0	0 – No CRC added 1 – CRC auto. added	
		1	7:5	RFU, set to '0'	
			4	0 - RWT = 2.4ms 1 – RWT is specified by PP:MM	
			3:0	Slot counter 0x0 – 1 slot 0x1 – 2 slots … 0xF – 16 slots	
		2	7:0	PP (max 14, i.e. 0x0E)	**Request Waiting Time (RWT) definition:** These 3 bytes are optional. The default value corresponds to a RWT of 302µs. $RWT = \frac{2^{PP} \cdot (MM+1) \cdot (DD+128) \cdot 32}{13.56} [\mu s]$ if PP is defined, then MM must be also defined while, DD remains optional
		3	7:0	MM (max 255, i.e. 0xFF)	
		4	7:0	DD (max 127, i.e. 0x7F)	

Table 10: Parameter values for "Protocol select" command (Reader)

Parameter list for different protocols (**TAG Emulation**)					
Protocol	Code	Parameters			Examples of commands Comments
		Byte	Bit	Function	
ISO14443A NFC-A	12	0	7:6	Transmission data rate 00 – 106kbps 01 – 212kbps 10..11 - RFU	**02021200** – TAG/Card emulation ISO/IEC14443A, 106kbps for transmission & reception, return error if no HF field detected, HFO used as master clock **0202120A** – TAG/Card emulation ISO/IEC14443A, 106kbps for transmission & reception, wait for HF field, CLKREC use as master clock
			5:4	Reception data rate 00 – 106kbps 01 – 212kbps 10..11 – RFU	
			3[1)]	0 = Return an error, if no field 1 = Wait for field	
			2	RFU, set to '0'	
			1	0 = HFO 1 = ClkRec	
			0	RFU, set to '0'	
ISO14443B NFC-B	13	0	7:6	Transmission data rate 00 – 106kbps 01 – 212kbps 10 – 424kbps 11 – 847kbps	**02021300** – TAG/Card emulation ISO/IEC14443B, 106kbps for transmission & reception, return error if no HF field detected, HFO use as master clock, CRC automatically added **0202130A** – TAG/Card emulation ISO/IEC14443B, 106kbps for transmission & reception, wait for HF field, CLKREC use as master clock, CRC automatically added
			5:4	Reception data rate 00 – 106kbps 01 – 212kbps 10 – 424kbps 11 – 847kbps	
			3[1)]	0 = Return an error, if no field 1 = Wait for field	
			2	RFU, set to '0'	
			1	0 = HFO 1 = ClkRec	
			0	0 – No CRC added 1 – CRC auto. added	
ISO18092 (212,424kb) NFC-F	14	0	7:4	RFU, set to '0'	**02021400** – TAG/Card emulation ISO/IEC18092, return error if no HF field detected, HFO use as master clock, CRC automatically added Note that it is not necessary to select a data-rate for ISO18092card mode, Data-rate will be automatically detected and adjusted during reception (application can read this information by sending "SubfreqRecv" command).
			3[1)]	0 = Return an error, if no field 1 = Wait for field	
			2	RFU, set to '0'	
			1	0 = HFO 1 = ClkRec	
			0	0 – No CRC added 1 – CRC auto. added	

Table 11: Parameter values for "Protocol select" command (TAG Emulation)

[1)] This option will be executed only after a "listen" command has been sent. Please refer to the chapter Listen command (0x05) for more information.

6.5 PollField command (0x03)

The “PollField” command is used to detect the presence of an HF field by monitoring the flag “FieldDet”. This command returns the current value of the flag “FieldDet”. The parameters <Presc> and <Timer> can also be used to define a time during which the MLX90132 continuously scans for the presence of HFfield. The answer to the “*PollField*” command is available with the flag <FieldDet> updated accordingly, after the scanning period is terminated.

PollField **0x03**			
Direction	Data	Comment	Example
MCU – device	03	Command code	**0300** – Check if Field is ON or OFF
	<Len>	Length of data	**0303010FFF**– Wait for field appearance during(16*256)/13.56=302µs
	<Flags>	Timer flag (Optional) 01 – Wait for field appearance 00 – Wait for field disappearance	
	<Presc>	Timer prescaler (Optional)	Parameters Flags, Presc and Timer are optional. They must be specified if application has to wait for field appearance or disappearance.
	<Timer>	Timer time-out (Optional)	The time to wait is: $\text{Time} = \frac{(\text{Presc}+1)\cdot(\text{Timer}+1)}{13.56}[\mu s]$
Device - MCU	00	Result code	**000101 – HF field is detected**
	01	Length of data	
	<FieldDet>	[7:1] – RFU [0] – 0 : No HF field detected 1 : HF field detected	

Table 12: “PollField” command

Note: *When the MLX90132 is selected in reader mode (protocol select command), the HF field will be automatically turned ON and the flag “FieldDet” will be set to ‘1’ (the MLX90132 detects its own field). Consequently, the PollField command should be used in Tag/Card Emulation state or in Reader state with the HF field set OFF.*

6.6 SendRecv command (0x04)

This command is used to send specific protocol data and receives corresponding answer. Before sending this command, the application must select a protocol using the Protocol select command. If the response of the Transponder was successfully received and decoded, the field <Data> will contain additional information which is protocol specific. This is explained in the Table 14 below.

SendRecv**0x04**			
Direction	Data	Comment	Example
MCU – device	04	Command code	**Depends on protocol previously selected!**
	<Len>	Length of data	
	<Data>	Data to be sent	**0403022012**– Command "Read single block 12" (ISO/IEC15693 protocol)
Device - MCU	<ResultCode>	Result code	**8008000000000077CF00 -** The response of the TAG is successfully decoded. This is an example of response from an ISO15693 TAG. The result code might contain additional information on the extended size of received data. Please refer to paragraph **Support of extended frames** below.
	<Len>	Length of data	
	<Data>	Data received. Interpretation depends on protocol	
Device - MCU	<ErrorCode>	Error code	Please refer to the error code table summary in the chapter List of Error codes
	<ErrorBufLen>	Length of Error Buffer stored during EMD algorithm	
	<ErrorBuf>	Error Buffer stored during EMD algorithm	

Table 13: "SendRecv" command description

Data format for transmission			
Protocol	Explanation	Response example	Comments
ISO15693	Send example Command code Length of entire data field Data	04 03 022000	If length of data is Zero, only EOF will be sent. This can be used for anti-collision procedure
ISO14443A NFC-A	Send example Command code Length of entire data field Data Transmission flags: 7 – 0 : ISO14443A 1: Topaz format (use EOF instead of P, use SOF at the beginning of each byte, make pause between bytes, assume 1st byte as 7-bit) 6 – SplitFrame if set 5 – append CRC if set 4 – Auto. add the parity bit in if set to '0'[1)] 3:0 – number of significant bits in last byte	04 07 9370800F8C8E 28	For bit oriented protocol, frames could be split by setting the bit **SplitFrame** to one. In this case, the MLX90132 will send the last byte of the command with none integer number of bits, according to the field **number of significant bits in last byte**. In reception, the MLX90132 expects to receive the complement (**8 – "number of significant bits in last byte"**). This option is used during anti-collision procedure.
ISO14443B NFC-B	Send example Command code Length of entire data field Data	04 03 050000	
ISO18092 (212,424Kb) NFC-F	Send example Command code Length of entire data field Data	04 05 00FFFF0000	

Table 14: Parameter values for "SendRecv" command

[1)]The process of automatically calculating and adding the parity bit by the MLX90132 can be disabled by setting the bit 4 of the flags to '1'. In this case, the application must add one byte to the data with the most significant bit corresponding to the parity bit. The other bits of these additional bytes are not considered and can be set to '0' or '1'. The datastream will then look like: <DataByte><Parity><DataByte><Parity>.

Interpretation of <Data> field for different protocols			
Protocol	Explanation	Response example	Comments
ISO15693	Response example | 80 | 08 | 0000000000 | 77CF | 00 Result code Length of entire data field Data received from TAG Original (received) value of CRC 7:2 – RFU 1 – CRC error if set 0 – Collision is detected if set		**000000000077CF**- this is a response on Read Single Block command for Iso15693 TAG. Other fields are added by the device
ISO14443A NFC-A	Response example | 80 | 09 | 80B30B8DB500 | 00 | 00 | 00 Result code Length of entire data field Data received from TAG 7 – Collision is detected 6 – RFU 5 – CRC error 4 – parity error 3:0 – shows how many significant bits are there in the first byte 7:0 – Index of the first byte where collision is detected 7:4 – RFU 3:0 – Index of the first bit where collision is detected		ISO/IEC14443A is bit oriented protocol, and non-integer amount of bytes can be received. **Number of significant bits in the 1st byte** is the same as indicated in Send command. To calculate a position of a collision, application has to take **index of byte** first. **Index of bit** indicates a position inside this byte. Note that both indices start from 0 and bit index can be 8, meaning that collision could also affect the parity bit. Note that collision information is only present when protocol ISO/IEC14443A with a data rate of 106kbps for transmission and reception is selected. When others protocols are selected, the two additional bytes are not transmitted.
ISO14443B NFC-B	Response example | 80 | 0F | 5092036A8D00000000007171 | 3411 | 00 Result code Length of entire data field Data received from TAG Original (received) value of CRC 7:2 – RFU 1 – CRC error if set 0 – RFU		
ISO18092 (212,424Kb) NFC-F	Response example | 80 | 12 | 01010105017B06941…93FF | 00 Result code Length of entire data field Data received from TAG 7:2 – RFU 1 – CRC error if set 0 – RFU		**801201010105017B06941004014B024F4 993FF00** – typical answer with no error detected

Table 15: "SendRecv" command, interpretation of <data> field for different protocol

6.6.1 Support of extended frames

In reader mode it is possible to receive up to 528 bytes of frame data. The extended size is included in the command code as follows:

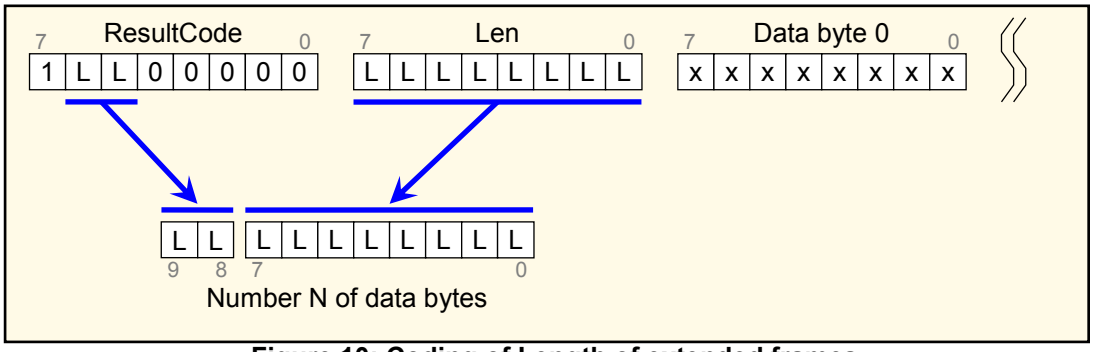


Figure 10: Coding of Length of extended frames

Consequently, the *ResultCode* returned depends on the length of the decoded frame received by the MLX90132.

Direction	Result Code	Length (LEN)	Effective length of received data	Comment
MCU - device	0x80	0x00 – 0xFF	0 – 255 bytes	
	0xA0		256 – 511 bytes	
	0xC0		512 – 528 bytes	
	0x90		0 – 255 bytes	In ISO/IEC14443A only in case of none integer number of bytes
	0xB0		256 – 511 bytes	
	0xD0		512 – 528 bytes	

Table 16: Coding of Length of extended frames

6.6.2 List of Error codes

The error code returned in the case of a“SendRecv” command includes the last error raised by the device in the field<ErrorCode>. But, it could also include a buffer of error if the EMD algorithm is enabled. This list of errors is stored into the dedicated buffer of maximum 8-bytes <ErrorBuf> with its length indicated in the error buffer length value <ErrorBufLen>. The list of error codes which could be returned after a “SendRecv” command is shown in the table below. The error codes marked with a * could be raised during the EMD process and stored in the Error buffer. For more information related to the EMD algorithm, please refer to the chapter Electromagnetic support (EMD) below.

Direction	Error Code	Definition
Device - MCU	0x61*	SOF error during the EMD process
	0x62*	CRC error during the EMD process
	0x63*	SOF error in ISO14443B occurs during high part (duration of 2 to 3 Elementary Unit Time, ETU)
	0x65*	SOF error in ISO14443B occurs during low part (duration of 10 to 11 Elementary Unit Time, ETU)
	0x66*	Extra Guard Time (EGT) error in ISO14443B
	0x67*	TR1 set by card too long in case of protocol ISO14443B
	0x68*	TR1 set by card too short in case of protocol ISO14443B
	0x86	Hardware Communication error
	0x82	Invalid command Length
	0x83	Invalid Protocol
	0x87	Frame waiting timeout (no valid reception) or no TAG
	0x88	Invalid SOF
	0x89	Receive buffer overflow (too many bytes received)
	0x8A	Protocol Framing error as follows: • ISO14443A & ISO18092 (106kbps) : Modified Miller, wrong symbol sequence • ISO14443B: Start/Stop bit polarity • ISO18092 (212,424kbps): SYNC ≠ 0xB24D
	0x8B	EGT time out (ISO14443B)
	0x8C	Invalid length received during ISO18092 (212,424kbps) communication (2 < Length < 255)
	0x8D	CRC error in case of protocolISO18092 (212,424kbps)
	0x8E	Reception lost without EOF received

Table 17: List of error codes for “SendRecv” command

6.7 Listen command (0x05)

This command is used with the MLX90132 in Tag emulation state to listen for the command from the reader. Before sending this command the application has to select a protocol using “Protocol Select” command with the related options.

Listen **0x05**			
Direction	Data	Comments	Example
MCU – device	05	Command code	**0500** – Listen for a request from reader
	00	Length of data	
Device - MCU	00	Result code	**0000-** No error. Confirmation that device now is in listening mode
	00	Length of data	
Device - MCU	82	Error code	**8200 -** Invalid command length
	00	Length of data	
Device - MCU	83	Error code	**8300 -** Invalid protocol or protocol is not supported.
	00	Length of data	
Device -MCU	85	Error code	**8500 -** Canceled by user using “Echo” command
	00	Length of data	
Device - MCU	8F	Error code	**8F00 -** No HF field detected, command cannot be executed
	00	Length of data	

Table 18: “Listen” command description

When the “listen” command is executed and the option “Waits for field” is activated, the MLX90132 waits for the HF field activation and corresponding request coming from an RFID reader.
If the option “Return an error if no field” is activated, the MLX90132 directly returns an error if no HF field is detected.

If the HF field is interrupted by the reader while the MLX90132 is waiting for the request, it will leave the listen command and return the error code 0x8F00.To wait for new request, the application must issue a new “listen” command.

The user can cancel the “listen” mode by issuing an “echo” command 0x55. When cancelled, the MLX90132 replies with a code 0x55 (as a sync reply) plus “Cancelled by user” message corresponding to 0x85, 0x00. To cancel the “listen” mode, the following procedure should be followed:

- Send the ECHO command 0x55 to cancel the “listen” mode
- Set the pin SPI_NSS to low, to read back the buffer content
- Read the sync reply 0x55
- By keeping SPI_NSS low, read the rest of the buffer (could be cancelled by user 0x8500 message or correct data information 0x80<LEN><DATA>)
- Set the pin SPI_NSS high to continue the operation

Possible return codes are listed in the table below.

Respond codes from the device in Listen mode			
Direction	Data	Comments	Example
Device - MCU	80	Result code	**800605000071FF00 -** The request from the Reader is decoded. This is an example of Request in Iso14443-B protocol
	<Len>	Length of data	
	<Data>	Data received. Interpretation depends on protocol	
Device - MCU	86	Error code	**8600 -** Communication error
	00	Length of data	
Device - MCU	87	Error code	**8700 -** Listening mode was cancelled by the application
	00	Length of data	
Device - MCU	88	Error code	**8800 -** Invalid SOF
	00	Length of data	
Device – MCU	89	Error code	**8900 -** Receive buffer overflow
	00	Length of data	
Device – MCU	8A	Error code	**8A00 -** Protocol Framing error: - ISO14443A & ISO18092 (106kbps): Mod. Miller, wrong symbol sequence - ISO14443B: Start/Stop bit polarity - ISO18092 (212,424kbps): SYNC ≠ 0xB24D
	00	Length of data	
Device – MCU	8B	Error code	**8B00 -** EGT time out (ISO14443B)
	00	Length of data	
Device - MCU	8E	Error code	**8E00 -** Reception lost without EOF received
	00	Length of data	

Table 19: "Listen" command, possible return codes

If the request from the Reader was successfully received and decoded, the MLX90132 will send data back to the application, as shown in the following table.

Data format sent to the application in 'Listen' mode			
Protocol	Explanation	Response example	Comments
ISO14443A NFC-A	Request example	80 0A 9370800F8C8E 8D 4E01 08	The anti-collision filter could be activated with the command "AcFilter". In this case, the complete anti-collision process is supported by the MLX90132 as soon as a "Listen" command is initiated. The information will be automatically sent by the MLX90132
	Result code		
	Length of entire data field		
	Data received from reader		
	Received value of BCC (if any)		
	Received value of CRC (if any)		
	7 – RFU 6 – RFU 5 – CRC error 4 – Parity error 3:0 – number of significant bits in last byte		
ISO14443B NFC-B	Request example	80 06 050000 71FF 00	
	Result code		
	Length of entire data field		
	Data received from Reader		
	Original (received) value of CRC		
	7:2 – RFU 1 – CRC error if set 0 – RFU		
ISO18092 (212, 424kbp) NFC-F	Request example	80 06 00FFFF0000 00	
	Result code		
	Length of entire data field		
	Data received from reader		
	7:2 – RFU 1 – CRC error if set 0 – RFU		

Table 20: Data format sent to the application in "Listen" mode

6.8 Send command (0x06)

This command is used with the MLX90132 in TAG emulation state, to send data back to the reader. This command sends specific protocol data without waiting for an answer.

Send **0x06**			
Direction	Data	Comments	Example
MCU – device	06	Command code	**Depends on protocol previously selected!**
	<Len>	Length of data	
	<Data>	Data to be sent	**040C50920E997500000000B37171** – Emulation of TAG response in ISO14443-B protocol
Device - MCU	00	Result code	**0000 -** Data was successfully sent
	00	Length of data	
Device - MCU	82	Error code	**8200 -** Invalid length
	00	Length of data	
Device - MCU	83	Error code	**8300 -** Invalid protocol previously selected by Select Protocol command
	00	Length of data	

Table 21: "Send" command description

Format of data to be sent using 'Send' command			
Protocol	Explanation	Response example	Comments
ISO14443A NFC-A	Send example Command code Length of entire data field Data 7:6 – RFU 5 – Append CRC 4 – RFU 3:0 – number of significant bits in first byte	06 03 0400 08	The anti-collision filter could be activated with the command "AcFilter". In this case, the complete anti-collision process is supported by the MLX90132 as soon as a "Listen" command is initiated. The information will be automatically sent by the MLX90132
ISO14443B NFC-B	Send example Command code Length of entire data field Data	06 04 01020304	
ISO18092 (212,424Kb) NFC-F	Send example Command code Length of entire data field Data Slot number (in which to reply)	06 05 01020304 00	The number of slot in which to reply is entered by the application in the field <Slot number>. In this case, the MLX90132 automatically manages the related timings defined by the protocol, to answer to the corresponding slot. This parameter is used for the Single Device Detection (SDD) process, For other commands, the field <Slot number> should simply be set to zero.

Table 22: Format of data to be sent using "Send" command