1. SPI command overview

The NJJ29C0B is a joint LF driver and receiver IC with embedded µController. The NJJ29C0B is controlled by a host controller via SPI commands which are part of a pre-defined SPI command set, containing commands for device configuration, settings and data exchange. The host controller acts as SPI master, while the NJJ29C0B serves as SPI slave.

1.1 SPI frame structure

The SPI commands are transmitted in SPI frames. An SPI frame consists of up to 256 bytes. The default value of RX and TX is 0x00, corresponding to an SPI interface in idle mode.

The first data byte of an SPI message is interpreted as SPI message length (LEN). LEN is valid between 0x01 and 0xFF and specifies the number of data bytes following (in general max. 255 data byte within one SPI message, some commands support less data byte). The data byte can contain any content (including value 0x00).

Table 1. SPI frame example

LEN	Data
0x04	0xF0 0x00 0x10 0x34

Any leading byte with the value 0x00 is discarded. A change from leading 0x00 bytes to any other value is allowed any time.

In case the specified number of data bytes is not received by the NJJ29C0B within the expected time, RX is reset and the first byte different to 0x00 is interpreted as length of the next host command frame. If more data bytes than expected are received, these data bytes are ignored and the RX buffer containing these bytes is cleared.

When reading data from the NJJ29C0B, 0x00 bytes shall be used as TX data. Similar, 0x00 bytes are transmitted while receiving. Fig 1 shows SPI frames consisting of host frame and response frame. When neither RX nor TX action is intended, it is recommended to send no leading bytes with value 0x00 to reduce device load.

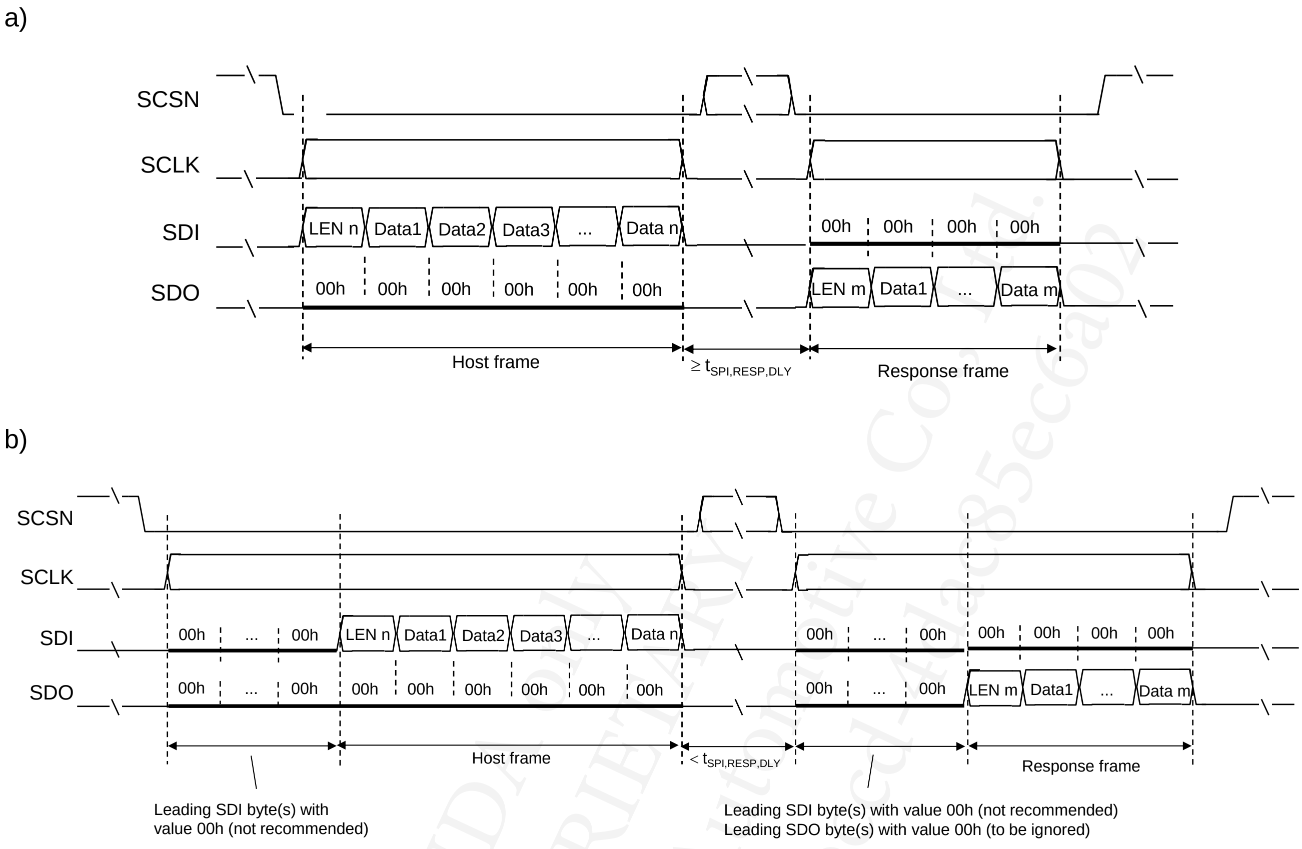

Fig 1. SPI frame (a) without data overhead (b) with leading 0x00 bytes (not recommended)

1.1.1 Host frame structure

The structure of a command frame from the host to the NJJ29C0B is shown in Table 2.

Table 2. SPI command structure from host to NJJ29C0B

LEN	CMD	PARAM	CRC8
1 byte	1 byte	0…253 bytes	1 byte

CMD holds the information which action should be taken. PARAM is optional and its size depends on CMD. PARAM can consist of 0 to 253 bytes.

Each SPI host frame is protected by a CRC8 at the frame end.

1.1.2 Response frame structure

Each command frame from the host is always acknowledged by a direct response frame from the NJJ29C0B.

The next host command frame shall not be sent before the direct response to the former command is received (e.g. by sending 0x00 bytes) or the respective maximum response delay time is elapsed.

The structure of a response frame from the NJJ29C0B to the host is shown in Table 3.

Table 3. SPI response structure from NJJ29C0B to host

LEN	CMD	STAT	PARAM	CRC8
1 byte	1 byte	1 byte	0…252 bytes	1 byte

CMD holds the echoed value from the host frame to the NJJ29C0B. The STAT byte contains the device status flags which signalize the application failures or special events that occurred during command execution. The PARAM coding depends on the CMD. PARAM can consist of 0 to 252 bytes.

Each SPI response frame is protected by a CRC8 at the frame end.

1.2 Bit and byte order

Each SPI byte is transmitted with the least significant bit first (LSBit first). For example, the byte 0xA2 (0b1010 0010) is transmitted as bit sequence via the SPI interface in the order 0b0100 0101.

Table 4. Example: LSBit and MSBit of 0xA2

MSBit							LSBit
1	0	1	0	0	0	1	0

If a value consists of 2 or more byte, it is transmitted with the least significant byte first (LSByte first). Each byte is transmitted with the least significant bit first (LSBit first). For example, the 16 bit value 0xC5A2 (0b1100 0101 1010 0010) is transmitted as bit sequence via the SPI interface in the order 0b0100 0101 1010 0011.

Table 5. Example: LSBit, MSBit, LSByte and MSByte of 0xC5A2

MSByte								LSByte							
MSBit							LSBit	MSBit							LSBit
1	1	0	0	0	1	0	1	1	0	1	0	0	0	1	0

For the sake of clarity, the position of low and high order bytes is specified unambiguously in each respective SPI command. Each low byte label ends with “_L” (e.g. “VALUE_L”), while each high byte label ends with “_H” (e.g. “VALUE_H”).

Table 6. Example SPI command sequence with 2 byte value

LEN	CMD	PARAM		CRC8
LEN	0xXX	VALUE_L	VALUE_H	CRC8

For values with 3 byte length, the medium byte is labeled with “_M” at the end (e.g. “VALUE_M”).

Table 7. Example SPI command sequence with 3 byte value

LEN	CMD	PARAM			CRC8
LEN	0xXX	VALUE_L	VALUE_M	VALUE_H	CRC8

Since not all values need all available bits given by the SPI byte structure, the bits are numbered from 0 (LSBit of LSByte) up to the MSBit of the MSByte in increasing order.

Table 8. Bit numbers at the example of value 0xC5A2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	1	0	1	1	0	1	0	0	0	1	0

If a value consist of several bits, this is noted as VALUE[m:n]. The following tables show examples for bit notations of values consisting of 8, 16 and 24 bit.

Table 9. VALUE8BIT (1 byte)

Byte symbol	Bit notation	Description
VALUE8BIT	VALUE8BIT[7:0]	8 bit value

Table 10. VALUE16BIT (2 byte)

Byte symbol	Bit notation	Description
VALUE16BIT	VALUE16BIT[15:0]	16 bit value
VALUE16BIT_H	VALUE16BIT[15:8]	16 bit value high byte
VALUE16BIT_L	VALUE16BIT[7:0]	16 bit value low byte

Table 11. VALUE24BIT (3 byte)

Byte symbol	Bit notation	Description
VALUE24BIT	VALUE24BIT[23:0]	24 bit value
VALUE24BIT_H	VALUE24BIT[23:16]	24 bit value high byte
VALUE24BIT_M	VALUE24BIT[15:8]	24 bit value medium byte
VALUE24BIT_L	VALUE24BIT[7:0]	24 bit value low byte

If a value consists of fewer bits than needed to complete a byte border, the remaining bits can be used otherwise. For example, a value consisting of 23 bits is noted as VALUE[22:0]. In this case, VALUE[23] representing the MSBit of the high byte can be used for other purposes.

1.3 SPI command set CMD

The CMD byte determines the device action to be taken. The SPI command set is summarized in Table 12.

Table 12. SPI command set[1][2]

Command	CMD coding	Description
General commands		
GET_VERSION	0xF0	Read device version identifiers
CONFIG_DEVICE	0x09	Configure device parameters
CONFIG_ADVANCED	0xD2	Configure advanced device parameters
Device reset		
SET_POR	0xF4	Perform a power on reset
GET_POR_STATUS	0x07	Read device power on reset status flags
CLEAR_POR_STATUS	0x08	Clear POR status flag which is set after device reset
Operating states		
START_SLEEP	0x50	Set the device in SLEEP state
START_SLEEP_FORCED	0x4F	Set the device in SLEEP state independent of the device state
Boost converter		
CONFIG_BOOST	0x45	Configure boost converter

Command	CMD coding	Description
LF antenna driver		
CONFIG_LF_DRIVER	0x40	Configure LF driver mode, baudrate, current, dithering, freq. shift
Parallel low current driver		
CONFIG_LC_DRIVER	0x44	Configure low current LF driver
Telegram sequencer		
SET_LF_DATA	0x41	Store data in RAM to be transmitted via the main drivers
SET_LC_DATA	0x47	Store data in RAM to be transmitted via the low current drivers
START_LF_TRANSMIT	0x42	Start LF transmission using pre-loaded configuration and data
START_LF_TRANSMIT_DATA	0x46	Start LF transmission using pre-loaded configuration and attached data
STOP_LF_TRANSMIT	0x43	Stop LF transmission
Immobilizer		
CONFIG_IMMO_DRIVER	0x60	Configure immobilizer antenna driver
CONFIG_IMMO_BPLM	0x61	Configure binary pulse length modulator timings
CONFIG_IMMO_RECEIVER	0x62	Configure immobilizer receiver
START_IMMO	0x63	Start immo driver and transmit constant carrier signal
STOP_IMMO	0x64	Stop constant carrier signal transmitted by immo driver
START_IMMO_TRANSCEIVE	0x65	Write to transponder and read response from transponder
START_IMMO_TRANSMIT	0x66	Write data to transponder
GET_IMMO_RESPONSE	0x67	Read data from receive data buffer
CLEAR_IMMO_STATUS	0x68	Clear immobilizer status flags
SET_IMMO_MASK	0x69	Mask immobilizer status flags to trigger INT pin
Antenna parameters		
MEAS_ANT_IMP	0x48	Measure antenna impedance values (variant 1)
MEAS_ANT_IMP_ADVANCED	0xD5	Measure antenna impedance values (variant 2)
SET_ANT_IMP	0x49	Set antenna impedance values
GET_ANT_IMP	0x4A	Read antenna impedance values
GET_ANT_IMP_EFFECTIVE	0xD4	Read the effective antenna impedance values
Device protection		
GET_PROT_STATUS	0x58	Read device protection status flags (malfunctions)
CLEAR_PROT_STATUS	0x59	Clear device protection status flags
SET_PROT_MASK	0x5A	Mask device protection flags to trigger INT pin
Device diagnostics		
START_DIAG	0x4C	Start diagnostics of boost converter, LF driver & antennas
GET_DIAG_STATUS	0x4D	Read device diagnostics status flags
CLEAR_DIAG_STATUS	0x4E	Clear device diagnostics status flags
SPI interface		
CONFIG_SPI	0xF1	Configure the SPI interface
ECHO_SPI	0x01	Check proper physical operation of the SPI link

Command	CMD coding	Description
Wake-up ports		
CONFIG_WUP	0x10	Configure wake-up ports
GET_WUP_STATUS	0x13	Read wake-up status flags
CLEAR_WUP_STATUS	0x14	Clear wake-up status flags
SET_WUP_MASK	0x15	Mask wake-up status flags to trigger INT pin
WUP event triggered polling		
CONFIG_WUP_POLLING	0x51	Configure driver sequence for wake-up via wake-up port
Timer triggered polling		
CONFIG_TIMER_POLLING	0x52	Configure polling timer and polling scheme
START_TIMER_POLLING	0x53	Sets the device in SLEEP state and starts timer triggered polling
Temperature indication		
CONFIG_TEMP	0x18	Configure threshold for temperature warning
GET_TEMP_STATUS	0x19	Read temperature status flag
CLEAR_TEMP_STATUS	0x1A	Clear temperature status flag
SET_TEMP_MASK	0x1B	Mask temperature status flag to trigger INT pin
Operation status flags		
GET_OP_STATUS	0x55	Read operation status flags
CLEAR_OP_STATUS	0x56	Clear operation status flags
SET_OP_MASK	0x57	Mask operation status flags to trigger INT pin
Program download		
DOWNLOAD_PROG	0x30	Stores program in RAM for execution
START_PROG	0x31	Executes program downloaded in RAM
GET_PROG_SIG	0x33	Read program signature

[1] The command set is subject to change

[2] Only the specified commands are allowed to be used, unless otherwise specified

In the detailed SPI command description, bits and bytes marked RFU are reserved for future use. For future compatibility, a write operation shall assign a ‘0’ or ‘0x00’ to an RFU bit or byte, respectively. Any read operation of an RFU bit or byte yields ‘0’ or ‘0x00’, respectively.

1.4 Command parameter PARAM

Most SPI commands contain parameters, which represent settings and/or data. These parameters are coded in the PARAM byte(s) included in the SPI frame.

PARAM:

PARAM contains one or more command parameters A, B, C, ….

Table 13. SPI command with parameters of different type

LEN	CMD	PARAM				CRC8
LEN	0xXX	A	B	C	…	CRC8

Parameters can be of the same type A (e.g. data byte), which are repeated A1, A2, …

Table 14. SPI command with parameters of same type

LEN	CMD	PARAM			CRC8
LEN	0xXX	A1	A2	…	CRC8

For the sake of a better lucidity, in this document a parameter which is repeated A1, A2, … is represented by its name followed by “i” in curly brackets {Ai}. The SPI message length LEN is used to determine the number of parameters in the respective command. The minimum number of parameters shall be one.

Table 15. SPI command with parameters of same type in simplified notation

LEN	CMD	PARAM	CRC8
LEN	0xXX	{Ai}	CRC8

Some commands allow to repeat parameter sequences A1, B1, C1, A2, B2, C2, ... containing parameters of different type. The parameters are repeated in the given order sequentially. Each parameter sequence represents a parameter set. The sequence order of the repeated parameters is unconstrained, hence the order A1, B1, C1, A2, B2, C2 is equivalent to the order A2, B2, C2, A1, B1, C1.

Table 16. SPI command with sequentially repeated parameters

LEN	CMD	PARAM										CRC8
LEN	0xXX	A1	B1	C1	A2	B2	C2	A3	B3	C3	…	CRC8

In the simplified notation this is shown by embedding the sequentially repeated parameters in curly brackets {Ai, Bi, Ci, …}. The SPI message length LEN is used to determine the number of parameter sequences (parameter sets) in the respective command. The number of sequences of the repeated parameters can be one.

Table 17. SPI command with sequentially repeated parameters in simplified notation

LEN	CMD	PARAM			CRC8
LEN	0xXX	{Ai	Bi	Ci}	CRC8

If the same parameters are transmitted more than once in the same command with different values, the last transmitted setting overrides any previous settings.

1.4.1 Driver identifier DRID

Some LF driver related SPI commands have a driver identifier DRID to select the requested driver, followed by parameters for this channel.

Table 18. SPI command structure for LF driver related commands

PARAM
DRID

DRID:

The driver (channel) ID identifies the LF channel for which the parameter set is assigned.

Table 19. DRID

Bit	Symbol	Access	Value	Description
7 to 3	RFU	W0		Reserved for future use
2 to 0	DRID	W		Driver identifier
			000	LF driver 1

Bit	Symbol	Access	Value	Description
			001	LF driver 2
			010	LF driver 3
			011	LF driver 4
			100	LF driver 5
			101	LF driver 6
			110	Reserved for future use
			111	Reserved for future use

1.4.2 Driver selection DRP

Some commands need the selection of more than one driver at the same time. These commands select the respective driver by setting the corresponding bit in the parameter DRP.

Table 20. Parameter for LF driver selection

PARAM	
DRP	RFU

DRP:

Selects the LF driver Tx.

Table 21. DRP

Bit	Symbol	Access	Value	Description
7 to 6	RFU	W0		Reserved for future use
5	DR6P	W		Select LF driver 6
			0	No selection
			1	Select
4	DR5P	W		Select LF driver 5
			0	No selection
			1	Select
3	DR4P	W		Select LF driver 4
			0	No selection
			1	Select
2	DR3P	W		Select LF driver 3
			0	No selection
			1	Select
1	DR2P	W		Select LF driver 2
			0	No selection
			1	Select
0	DR1P	W		Select LF driver 1
			0	No selection
			1	Select

1.4.3 Driver assignment order

In some cases the device has to assign by itself the respective dedicated drivers to parameter sequences transmitted via host command. In this case the drivers are assigned to the parameter sequences in ascending driver number, starting with the lowest number (Table 22).

Table 22. Driver assignment order

Number	Driver assignment
1	LF driver 1
2	LF driver 2
3	LF driver 3
4	LF driver 4
5	LF driver 5
6	LF driver 6

Example:

If in one command driver 1, 3 and 4 need to be assigned, the following order is applied:

- 1st: LF driver 1
- 2nd: LF driver 3
- 3rd: LF driver 4

1.4.4 Low current driver selection LCDRP

The commands to start LF transmission allow the activation of low current drivers in parallel to the main driver(s). Only low current drivers of channels not selected as main driver are allowed to be selected by setting the corresponding bits in the parameter LCDRP.

When activating the main drivers, the selected low current drivers start transmitting automatically.

Table 23. Parameter for low current driver selection

PARAM	
LCDRP	RFU

LCDRP:

LCDRP selects the low current LF driver Tx.

Table 24. LCDRP (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 6	RFU	W0		Reserved for future use
5	LCDR6P	W		Select LF driver 6
			0	No selection
			1	Select
4	LCDR5P	W		Select LF driver 5
			0	No selection
			1	Select

Bit	Symbol	Access	Value	Description
3	LCDR4P	W		Select LF driver 4
			0	No selection
			1	Select
2	LCDR3P	W		Select LF driver 3
			0	No selection
			1	Select
1	LCDR2P	W		Select LF driver 2
			0	No selection
			1	Select
0	LCDR1P	W		Select LF driver 1
			0	No selection
			1	Select

1.4.5 LF data bit order

The sequences to transmit or receive LF data are handled by splitting up the data into complete bytes and one last byte, if dependent on the LF protocol the bit number deviates from a multiple of 8 bits.

1.4.5.1 LF transmit

The following SPI commands specify data to be transmitted via the device LF interface

- SET_LF_DATA
- SET_LC_DATA
- START_LF_TRANSMIT_DATA
- START_IMMO_TRANSMIT
- START_IMMO_TRANSCEIVE (transmit part)

Data is transmitted byte oriented (each data value is represented by one byte) via the SPI interface starting with the first data byte down to the last data byte.

The last, possible incomplete byte holds the valid data MSB aligned (e.g. for 5 bits 0b10101xxx, last 3 bits xxx should be set to “0” and are ignored by the embedded system software). The data is sent with the MSB first.

Example:

A sequence of 18 bits (DATBLEN = 0x12) should be transmitted via the LF interface

0b1110 0010 1100 1100 10

In the example, first DATA1 is sent completely, and then DATA2 is transmitted. Due to the specified overall data length of 18 bit, finally the first 2 bits of DATA3 are MSB aligned and the remaining 6 lower bits of DATA3 are padded with zeros (Fig 2).

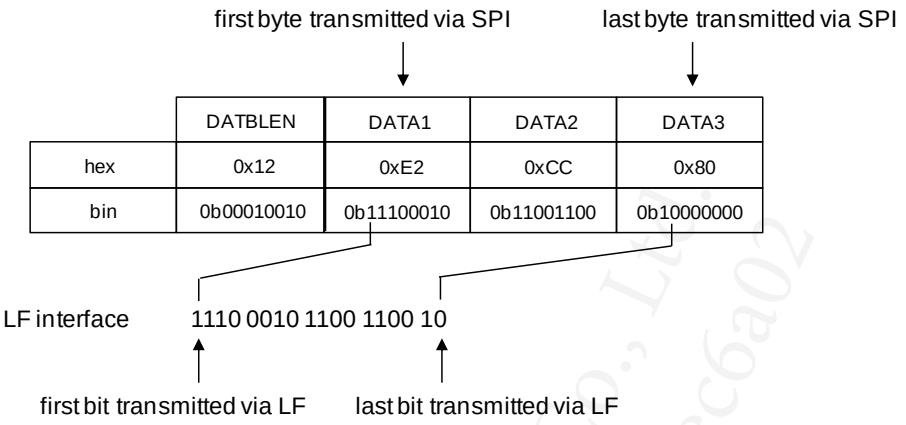

Fig 2. Example: Transmit 18 bits via the LF interface

1.4.5.2 LF receive

One command handles data received via the device LF interface

- GET_IMMO_RESPONSE

Data is received byte oriented (each data value is represented by one byte) via the SPI interface starting with the first data byte down to the last data byte.

The last, possible incomplete byte holds the valid data LSB aligned (e.g. for 5 bits 0bxxx10101, the higher first 3 bits are set to “0” and should be ignored by the host controller).

The receive timing is always a multiple of 8 bit. The embedded system software waits for the last complete byte received via the LF interface and replaces the non-used bits with zeros.

Example:

A sequence of 18 bits should be received via the LF interface

0b1110 0010 1100 1100 10

In the example, first DATA1 is received completely, and then DATA2 is received. Due to the specified overall data length of 18 bit, finally the first 2 bits of DATA3 are LSB aligned and the remaining 6 upper bits of DATA3 are padded with zeros (Fig 3).

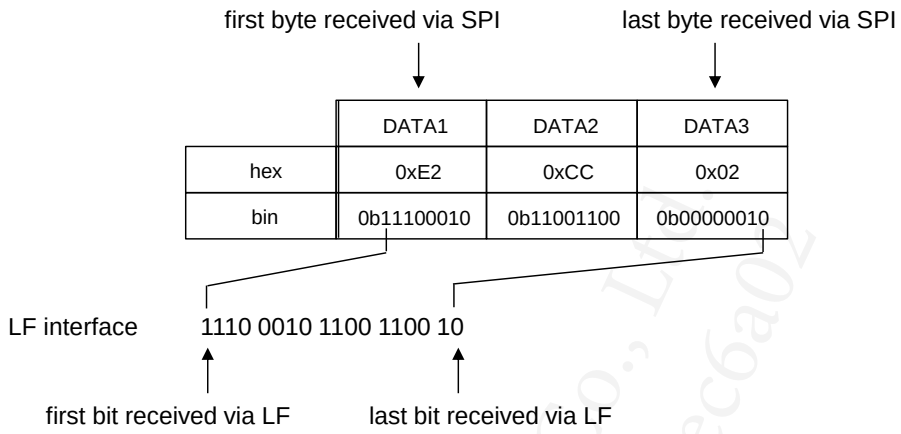

Fig 3. Example: Receive 18 bits via the LF interface

1.5 Device status flags STAT

Each command response contains the device status flags. The device status flags are coded in the STAT byte, including the status of the flag indicating a device reset.

STAT:

The device status flags can be split up in two groups

- Dynamic status flags STAT[7:4] and
- Static status flags STAT[3:0]

The dynamic status flags

- SF_CMD
- SF_LAST_OP
- SF_CRC and
- SF_PAR

represent the result of the last SPI command and change their values dynamically.

The static status flags

- SF_OP
- SF_WUP
- SF_PROT and
- SF_POR

represent group flags summarizing events in the corresponding group (operation status flag group, wake-up status flag group, protection status flag group and power-on-reset status flag group). Once set, these flags keep their values until the corresponding detailed status flags in the respective groups are cleared by the host controller.

Table 25. STAT (reset value 0x01)

Bit	Symbol	Access	Value	Description
7	SF_CMD	R		Command code interpretation status
			0	Valid command
			1	Command not known or (temporarily) not allowed
6	SF_LAST_OP	R		LF field activation status
			0	Activation successful
			1	Failure during activation
5	SF_CRC	R		CRC status
			0	No error
			1	Error
4	SF_PAR	R		Parameter interpretation status
			0	Feasible
			1	Failed
3	SF_OP	R		Operation status
			0	No request
			1	Request
2	SF_WUP	R		Wake-up status
			0	No request
			1	Request
1	SF_PROT	R		Protection status
			0	No request
			1	Request
0	SF_POR	R		Device reset status
			0	No reset
			1	Reset

SF_CMD:

SF_CMD is set if the command is not allowed (in this moment), because

- an unknown SPI command coding is used (enumerator failure)
- a SPI command is sent not allowed in the actual device operation state
- the NJJ29C0B processes a non-blocking command and the host controller interrogates a non-allowed SPI command before the NJJ29C0B sets the operation status flag

If an SPI response is provided with the status flag SF_CMD set, the intended function of the formerly sent command host frame was not executed.

SF_LAST_OP:

SF_LAST_OP is set if an error occurs during the following operations when activating the LF field

- Boost converter start-up
- LF driver supply start-up

- LF driver start-up
- Current control loop start-up
- PLL start-up
- Immo LDO diagnostic
- Immo ADC calibration (applies only for immo operation)

SF_LAST_OP thus indicates that the expected driver operation could not be performed.

For most SPI commands which can report SF_LAST_OP, SF_LAST_OP is not set in the direct response, but is reported with the response of the subsequent command. An exception is the command START_IMMO, which reports SF_LAST_OP in the direct response.

As besides SF_LAST_OP also SF_PROT is related to LF driver operations, the following conditions show which flag is set dependent on the failure condition.

SF_LAST_OP is set

- For commands START_LF_TRANSMIT, START_LF_TRANSMIT_DATA, START_TIMER_POLLING:
If during LF activation within $t_{LF,START}$ a STOP_LF_TRANSMIT command is received the SF_LAST_OP flag is reported in the response of the subsequent command.
- For commands MEAS_ANT_IMP and MEAS_ANT_IMP_ADVANCED:
If VBAT is larger than 18V an antenna impedance measurement is not performed and the SF_LAST_OP flag is reported in the response of the subsequent command.
- For command MEAS_ANT_IMP_ADVANCED:
If the impedance measurement was not successful for one or more antennas, e.g., due to an open antenna connection, the SF_LAST_OP flag is reported in the response of the subsequent command.
- For all commands activating the LF field:
If an error (and no protection) occurs during operations activating the LF field, SF_LAST_OP is reported in the response of the subsequent command (exception: START_IMMO reports SF_LAST_OP in the direct response).

SF_PROT is set

- For all commands activating the LF field:
When a protection occurs before the LF activation starts, SF_PROT and no SF_LAST_OP flag is reported in the response of the subsequent command.

Both SF_LAST_OP and SF_PROT are set

- For all commands activating the LF field:
When a device protection flag is raised after any hardware block is enabled both SF_LAST_OP and SF_PROT are reported in the response of the subsequent command (exception: START_IMMO reports SF_LAST_OP and SF_PROT in the direct response).

SF_CRC:

SF_CRC is set if the CRC8 attached to the SPI command has been wrong.

It is important to notice that if SF_CRC is set in the command response, the echoed command code cannot be treated as correct.

If an SPI response is provided with the status flag SF_CRC set, the intended function of the formerly sent command host frame was not executed.

SF_PAR:

SF_PAR is set if parameters transmitted via an SPI command cannot be interpreted correctly. This includes parameter values outside of their valid range and inconsistent lengths especially on cascased frames. SF_PAR is also set if RFU bits are set to 1.

Further, SF_PAR is set if the parameter set is not feasible, for example if a CLEAR_XX_STATUS command is sent with no status flag marked to be cleared.

If an SPI response is provided with the status flag SF_PAR set, the intended function of the corresponding command host frame was not executed.

SF_OP:

SF_OP is set if a non-blocking command has finished its operation and thus set its dedicated operation status flag.

If an SPI response is provided with the status flag SF_OP set, the command GET_OP_STATUS should be interrogated to check which operation has finished.

SF_WUP:

SF_WUP is set if a valid wake-up event has been detected at an enabled wake-up port.

If an SPI response is provided with the status flag SF_WUP set, the command GET_WUP_STATUS should be interrogated to check which wake-up port has triggered.

SF_PROT:

SF_PROT is set if a device protection event has been triggered. The device deactivates the boost converter and the classD* drivers (if activated) by hardware. Afterwards, it enters ERROR state.

If an SPI response is provided with the status flag SF_PROT set, the command GET_PROT_STATUS should be interrogated to check which protection event has triggered.

SF_POR:

SF_POR is set if a device power-on reset (POR) had occurred. After releasing the power on reset the SPI response is provided with the status flag SF_POR set.

If an SPI response is provided with the status flag SF_POR set, the command GET_POR_STATUS should be interrogated to check which event has triggered the power-on reset.

1.5.1 Multiple status flag handling

Dependent on the SPI command and on the parameters in some cases more than one device status flag might fit to be set. In these cases, the status flags are checked and set in the following order (Fig 4).

- SF_CRC is always checked first as in failure case the remaining content of the SPI command is not reliable

- SF_CMD is handled after SF_CRC and overrules SF_PAR, as parameter evaluation premises the precise knowledge of the SPI command
- SF_PAR is evaluated when no SF_CRC and SF_CMD issue are detected before

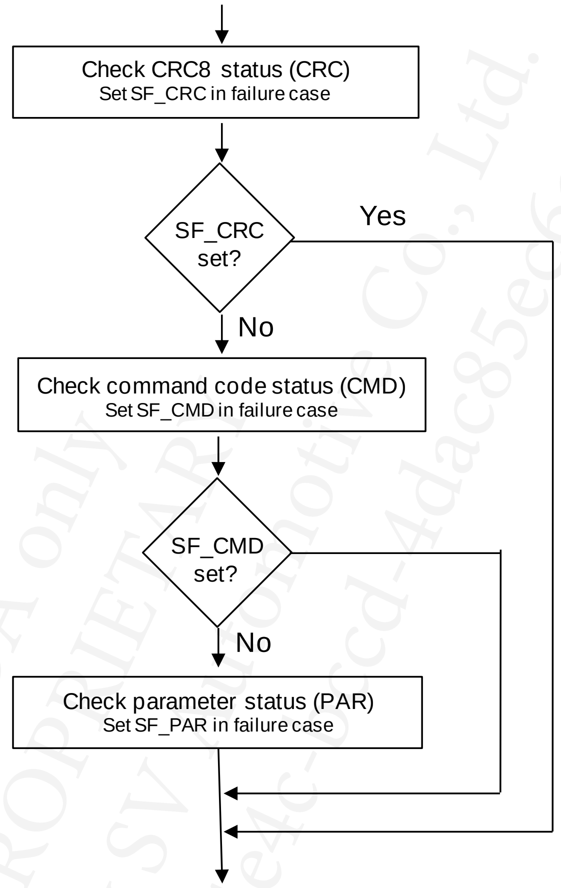

Fig 4. Multiple status flag handling

The parameter check is aborted if a failure is detected. For example, when the host controller interrogates an SPI command in a forbidden device operating state, SF_CMD is set, while the SPI command parameters PARAM are not checked.

1.5.2 Status flag events used by all commands

The following device status flags are used generically by all SPI commands if one of the following failures or events occurs.

Table 26. Status flag events used by all commands

Status byte	Status bit	Reported failure or event
STAT	SF_CMD	Command not allowed (in current device state)[1]
	SF_LAST_OP	Failure during LF power path activation (caused by command sent directly before)
		LF transmission aborted (caused by command STOP_LF_TRANSMIT sent directly before, when interogated after starting LF and before $t_{LF,START}$ has been elapsed)

Status byte	Status bit	Reported failure or event
	SF_CRC	SPI CRC error
	SF_PAR	Invalid SPI frame length
		Reserved for future use (RFU) bit or field values set to > 0[2]
	SF_OP	All LF data received (caused by non-blocking command sent before)
		All LF data transmitted (caused by non-blocking command sent before)
		Antenna impedance measurement finished (caused by non-blocking command sent before)
		Antenna diagnosis finished (caused by non-blocking command sent before)
	SF_WUP	WUP triggered
	SF_PROT	Protection triggered
	SF_POR	Device reset triggered

[1] Device state includes ERROR state that was entered due to protection event

[2] Includes “Reserved for future use “(RFU) bit or field values set by command parameters that are invalid or out of range

The assignment of the remaining device status flags events to the respective SPI commands is given in the detailed SPI command description.

1.6 Cyclic redundancy check CRC8

An 8 bit cyclic redundancy check value (CRC8) is attached to each SPI command. The CRC8 computation comprises all data bytes including LEN up to the last PARAM byte.

CRC8:

The CRC8 is computed according to the following algorithm.

CRC8 calculation

```
WORD wCRC;

wCRC = 0x00;
for( each data bit )
{
   wCRC = wCRC * 2;
   if(( wCRC >= 0x100 ) ^ ( data bit == 1 ))
   {
      wCRC = wCRC ^ 0x07;
      // CRC Polynome = (x^8 + x^2 + x + 1)
   }
   wCRC = wCRC & 0xFF;
}
```

Table 27 shows an example CRC8 calculation.

Table 27. Example for CRC8 calculation

Input (hex)	CRC8 (hex)
00 08 3F 00	4A

Input (hex)	CRC8 (hex)
FF 42 CA FE	C8
02 F0	76
0B F0 01 00 09 05 02 07 00 FF FF	EB

1.7 Command frame timing

1.7.1 Direct frame response

The direct response frame is sent with a delay time after receiving the CRC8 of the host frame. This response delay time $t_{RESP,DLY,CMD}$ depends on the SPI command and command parameters, as the correct command length, CRC8 and command parameters needs to be checked (Fig 5).

It is important to notice that any additional processing activity of the integrated µController like protection handling, active LF interface (e.g. LF channel change),WUP events may enlarge the response delay time.

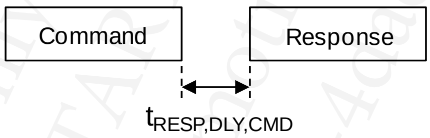

Fig 5. Command frame and response frame timing

1.7.2 Non-blocking commands

Some SPI commands initiate operations on the NJJ29C0B that need extended processing time and that are having operation status flags. These non-blocking commands are summarized in Table 28.

Table 28. Non-blocking commands with operation status flag

Non-blocking command	Operation status flag
START_LF_TRANSMIT	SF_TXREADY
START_LF_TRANSMIT_DATA	SF_TXREADY
START_IMMO_TRANSMIT	SF_TXREADY
START_IMMO_TRANSCEIVE	SF_RXREADY
MEAS_ANT_IMP	SF_IMPMEAS
MEAS_ANT_IMP_ADVANCED	SF_IMPMEAS
START_DIAG	SF_DIAG

After receiving one of these commands, the SPI response is prepared to be sent and the initiated NJJ29C0B operation (e.g. diagnostics) is started. The direct response indicating that the command is executed can be interrogated by the application after the command response delay time $t_{RESP,DLY,CMD}$. The initiated operation is ongoing after the direct SPI response has been sent (Fig 6).

The operation delay time $t_{OP,DLY,CMD}$ depends on the command and on the command parameters. The completion of the operation is signalized by a dedicated operation status flag, which can be checked by the host controller or which setting can be assigned to the device INT pin.

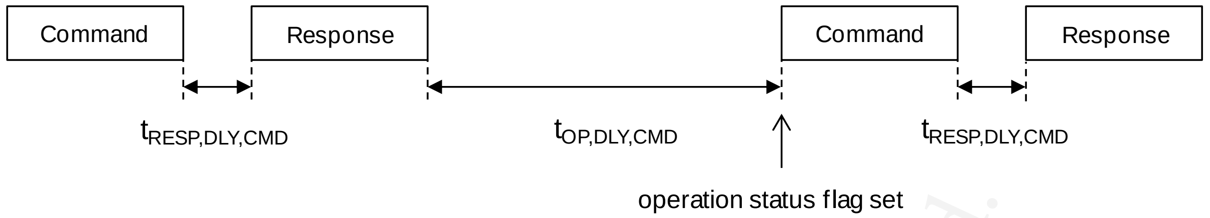

Fig 6. Command frame and response frame of non-blocking commands

After sending a non-blocking command and before the operation is finished, the following SPI commands are allowed to be interrogated

- GET_POR_STATUS
- GET_PROT_STATUS
- GET_DIAG_STATUS
- GET_WUP_STATUS
- GET_TEMP_STATUS
- GET_OP_STATUS
- START_SLEEP_FORCED
- SET_POR
- STOP_LF_TRANSMIT (only after sending commands START_LF_TRANSMIT or START_LF_TRANSMIT_DATA)
- STOP_IMMO (only after sending commands START_IMMO_TRANSMIT or START_IMMO_TRANSCEIVE)

Especially GET_OP_STATUS is intended to check when the initiated operation has been finished (if INT pin is not used for this purpose) (Fig 7).

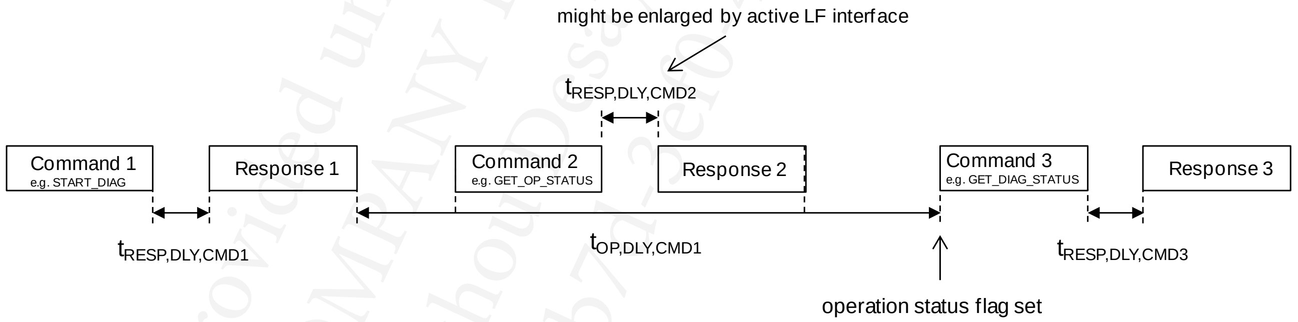

Fig 7. Host controller sends non-blocking command 1 and interrogates afterwards allowed SPI command 2

If the host controller interrogates a differing SPI command before the NJJ29C0B sets the operation status flag, this command is not executed and the SPI response is provided with the status flag SF_CMD set.

As the processing of SPI commands needs some processing time, in which the initiated NJJ29C0B operation is delayed, interrogating several SPI commands during running operation may influence the operation execution time.

After completion of the initiated operation, all SPI commands are allowed again to be sent.

1.8 Command restrictions

Some SPI commands are allowed under special device conditions only. Table 29 summarizes the SPI command restrictions dependent on the device operating state.

If an SPI command is interrogated that is not allowed in the respective device operating state, this command is not processed and the direct SPI response contains the device status flag SF_CMD set.

Table 29. SPI command restrictions

Command	IDLE state	PKE state	IMMO state	DIAG state	ERROR state	State during operation	State after completion[1]
General commands							
GET_VERSION	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
CONFIG_DEVICE	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
CONFIG_ADVANCED	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
Device reset							
SET_POR	Allowed	Allowed	Allowed	Allowed	Allowed	Cold start	IDLE
GET_POR_STATUS [2]	Allowed	Allowed	Allowed	Allowed	Allowed	No change	No change
CLEAR_POR_STATUS	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
Operating states							
START_SLEEP	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	SLEEP
START_SLEEP_FORCED [2]	Allowed	Allowed	Allowed	Allowed	Allowed	No change	SLEEP
Boost converter							
CONFIG_BOOST	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
LF antenna driver							
CONFIG_LF_DRIVER	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
Parallel low current driver							
CONFIG_LC_DRIVER	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
Telegram sequencer							
SET_LF_DATA	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
SET_LC_DATA	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
START_LF_TRANSMIT	Allowed	Not allowed	Not allowed	Not allowed	Not allowed	PKE	IDLE
START_LF_TRANSMIT_DATA	Allowed	Not allowed	Not allowed	Not allowed	Not allowed	PKE	IDLE
STOP_LF_TRANSMIT [2]	Not allowed	Allowed	Not allowed	Not allowed	Not allowed	No change	IDLE
Immobilizer							
CONFIG_IMMO_DRIVER	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
CONFIG_IMMO_BPLM	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change

Command	IDLE state	PKE state	IMMO state	DIAG state	ERROR state	State during operation	State after completion[1]
CONFIG_IMMO_RECEIVER	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
START_IMMO	Allowed	Not allowed	Not allowed	Not allowed	Not allowed	No change	IMMO
STOP_IMMO [2]	Not allowed	Not allowed	Allowed	Not allowed	Not allowed	No change	IDLE
START_IMMO_TRANSCEIVE	Not allowed	Not allowed	Allowed	Not allowed	Not allowed	No change	No change
START_IMMO_TRANSMIT	Not allowed	Not allowed	Allowed	Not allowed	Not allowed	No change	No change
GET_IMMO_RESPONSE	Allowed	Not allowed	Allowed	Not allowed	Allowed	No change	No change
CLEAR_IMMO_STATUS	Allowed	Not allowed	Allowed	Not allowed	Allowed	No change	No change
SET_IMMO_MASK	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
Antenna impedance							
MEAS_ANT_IMP [3]	Allowed	Not allowed	Not allowed	Not allowed	Not allowed	PKE	IDLE
MEAS_ANT_IMP_ADVANCED [3]	Allowed	Not allowed	Not allowed	Not allowed	Not allowed	PKE	IDLE
SET_ANT_IMP	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
GET_ANT_IMP	Allowed	Not allowed	Not vallowed	Not allowed	Allowed	No change	No change
GET_ANT_IMP_EFFECTIVE	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
Device protection							
GET_PROT_STATUS [2]	Allowed	Allowed	Allowed	Allowed	Allowed	No change	No change
CLEAR_PROT_STATUS	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	IDLE if all device protection flags are cleared. No change otherwise.
SET_PROT_MASK	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
Device diagnostics							
START_DIAG	Allowed	Not allowed	Not allowed	Not allowed	Not allowed	DIAG	IDLE
GET_DIAG_STATUS [2]	Allowed	Allowed	Allowed	Allowed	Allowed	No change	No change
CLEAR_DIAG_STATUS	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
SPI interface							
CONFIG_SPI	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
ECHO_SPI	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
Wake-up ports							
CONFIG_WUP	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
GET_WUP_STATUS [2]	Allowed	Allowed	Allowed	Allowed	Allowed	No change	No change
CLEAR_WUP_STATUS	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
SET_WUP_MASK	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
WUP event triggered polling							
CONFIG_WUP_POLLING	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
Timer triggered polling							
CONFIG_TIMER_POLLING	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change

Command	IDLE state	PKE state	IMMO state	DIAG state	ERROR state	State during operation	State after completion[1]
START_TIMER_POLLING	Allowed	Not allowed	Not allowed	Not allowed	Not allowed	No change	PKE state for PTIME = 0. Periodical repetition of POLLING/ IDLE/PKE state for PTIME > 0
Temperature indication							
CONFIG_TEMP	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
GET_TEMP_STATUS [2]	Allowed	Allowed	Allowed	Allowed	Allowed	No change	No change
CLEAR_TEMP_STATUS	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
SET_TEMP_MASK	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
Operation status flags							
GET_OP_STATUS [2]	Allowed	Allowed	Allowed	Allowed	Allowed	No change	No change
CLEAR_OP_STATUS	Allowed	Not allowed	Allowed	Not allowed	Allowed	No change	No change
SET_OP_MASK	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
Program download							
DOWNLOAD_PROG	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
START_PROG	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change
GET_PROG_SIG	Allowed	Not allowed	Not allowed	Not allowed	Allowed	No change	No change

[1] Each command can complete also in ERROR state if a protection event occurs during command execution

[2] SPI command allowed to be interrogated when non-blocking commands are operated

[3] During operation of either MEAS_ANT_IMP or MEAS_ANT_IMP_ADVANCED in PKE state the command STOP_LF_TRANSMIT is not allowed

1.9 Command parameter restrictions

All command parameters representing analog characteristics used in the following command descriptions (e.g. currents, voltages, impedances, inductances, quality factors, phases, times, frequencies, bit rates, temperatures and percentages) are symbolic typical values and may deviate from the exact values given in the parameter description.

2. General commands

2.1 GET_VERSION

GET_VERSION provides the version identifiers.

2.1.1 Command

Table 30. GET_VERSION command

LEN	CMD	CRC8
0x02	0xF0	CRC8

2.1.2 Response

Table 31. GET_VERSION response

LEN	CMD	STAT	PARAM								CRC8
0x0B	0xF0	STAT	PI_L	PI_H	HW_L	HW_H	ROM_L	ROM_H	RAM_L	RAM_H	CRC8

PI:

PI contains the product identifier.

Table 32. PI (reset value 0xXXXX)

Bit	Symbol	Access	Value	Description
15 to 0	PI[15:0]	R	0xXXXX	Product identifier

HW:

HW contains the hardware version identifier.

Table 33. HW (reset value 0xXXXX)

Bit	Symbol	Access	Value	Description
15 to 0	HW[15:0]	R	0xXXXX	Hardware version

ROM:

ROM contains the ROM version identifier.

Table 34. ROM (reset value 0xXXXX)

Bit	Symbol	Access	Value	Description
15 to 0	ROM[15:0]	R	0xXXXX	ROM version

RAM:

RAM contains the RAM version identifier. If no RAM download is performed, RAM version number 0xFFFF is returned.

Table 35. RAM (reset value 0xFFFF)

Bit	Symbol	Access	Value	Description
15 to 0	RAM[15:0]	R	0xXXXX	RAM version

2.1.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

2.2 CONFIG_DEVICE

CONFIG_DEVICE configures device parameters.

2.2.1 Command

Table 36. CONFIG_DEVICE command

LEN	CMD	PARAM	CRC8
0x03	0x09	DEVPAR	CRC8

DEVPAR:

Table 37. DEVPAR (default value 0x01)

Bit	Symbol	Access	Value	Description
7 to 2	RFU	W0		Reserved for future use
1	PREAMB	W		Send NXP preamble and code violation pattern automatically
			0	Disable
			1	Enable
0	VIO	W		VIO voltage level
			0	3.3 V
			1	5 V

PREAMB:

If PREAMB is enabled, every time a driver starts LF transmission automatically the NXP preamble and code violation (synchronization) pattern is sent before transmitting further data. The data sent after the preamble is attached without interruption of the data stream.

VIO:

If a level change is requested, the voltage is changed before the response is sent.

2.2.2 Response

Table 38.CONFIG_DEVICE response

LEN	CMD	STAT	CRC8
0x03	0x09	STAT	CRC8

2.2.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

2.3 CONFIG_ADVANCED

The initial settings of the device parameters are optimized to cover the needs of typical applications. For some device parameters it might be advantageous to adapt them to the application.

CONFIG_ADVANCED allows thus to overwrite the initially configured values of these device parameters with updated ones.

2.3.1 Command

Table 39. CONFIG_ADVANCED command

LEN	CMD	PARAM		CRC8
LEN	0xD2	{CAPARi	{VALUEij}}	CRC8

CAPARi, VALUEij:

CAPARi selects the CONFIG_ADVANCED parameter to be adapted with VALUEij.

Table 40. CONFIG_ADVANCED parameters

CAPARi	Function	VALUEij	Type	Description
0x04		DRVMINTON	VALUE8BIT	LF driver minimum on-time
0x12	CONFIG_CHIRP	CHIRPBLANK	VALUE8BIT	Number of chirp blanking samples
		CHIRPSTEPINC	VALUE8BIT	Number of frequency step increments
		CHIRPSTEPAFTPK	VALUE8BIT	Number of frequency steps after the peak
		CHIRPSTEPBCK	VALUE8BIT	Number of frequency steps jumped back
		CHIRPCYCLE	VALUE8BIT	Number of cycles per frequency

2.3.1.1 DRVMINTON

The antenna current is measured during the on-time of the classD* LF driver. Before sampling the mirrored signal in the current sense circuitry, the driver pulse needs to be settled. This settling time might be effected by an external capacitor connected at the TX pins to GND, dependent on the application.

The initial value for the LF driver minimum on-time is optimized to cover typical applications. In some applications it might be advantageous to update the default value for the LF driver minimum on-time. CONFIG_ADVANCED DRVMINTON allows thus overwriting the default setting for the LF driver minimum on-time.

Table 41. DRVMINTON (reset value 0x05)

Bit	Symbol	Access	Value	Description
7 to4	RFU	W0		Reserved for future use
3 to 0	DRV_MIN_TON[3:0]	W		LF driver minimum on-time
			0000	200 ns
			0001	250 ns
			…	…
			1110	900 ns
			1111	950 ns

DRV_MIN_TON:

The parameter DRV_MIN_TON adapts the time interval, in which a valid antenna driver current measurement will be performed (referenced to the middle of the active driver cycle). Regulated FB feedback operation is selected, if the active driver cycle is equal or larger than 2 x (DRV_MIN_TON + 100 ns).

2.3.1.2 CONFIG_CHIRP

A fast and accurate determination of the antenna resonance frequency is made with the help of a frequency chirp signal, consisting of a fast forward chirp and a slow reverse chirp.

The fast forward chirp takes every 6 cycles a sample value and increases the frequency by 2 frequency steps (stepsize $\Delta f = f_C / 256 = 488.28125$ Hz). Once a peak is detected, the fast forward chirp terminates.

Based on this first estimate of the resonance frequency, the slow reverse chirp starts near the termination frequency of the fast chirp. The slow reverse chirp applies every frequency for 8 or 9 cycles, before it samples. Afterwards, the frequency is slowly stepped back (stepsize $\Delta f = f_C / 256 = 488.28125$ Hz), until the sample maximum has been determined.

The chirp signal is optimized to cover typical applications. The optimization is done considering both operation delay time and result accuracy. In some applications, it might be advantageous to tailor the chirps with regard to the applied antennas. CONFIG_ADVANCED allows adapting the relevant chirp signal parameters.

CHIRPBLANK:

At the beginning of the chirp signal will be an overshoot, caused by the antenna driven with a frequency deviating from the resonance frequency. To avoid falsely detecting a peak based on this overshoot, by default the first sample values are discarded (blanked).

The initial number of blanked first samples is optimized to cover typical applications. In some applications, it might be advantageous to update the default value.

CONFIG_ADVANCED CHIRPBLANK allows thus overwriting the default setting for the number of blanked first samples.

Table 42. CHIRPBLANK (reset value 0x02)

Bit	Symbol	Access	Value	Description
7 to 3	RFU	W0		Reserved for future use
2 to 0	CHIRP_BLANK[2:0]	W		Number of chirp blanking samples
			0x00	Reserved for future use
			0x01	1
			0x02	2
			0x03	3
			0x04	4
			0x05	Reserved for future use
			…	
			0x07	Reserved for future use

CHIRPSTEPINC:

In order to speed up the antenna resonance frequency measurement, the chirp frequency is incremented by 2 frequency steps (stepsize $\Delta f = f_C / 256 = 488.28125$ Hz) every 6 cycles during the fast forward chirp.

The initial number of frequency step increments is optimized to cover typical applications. In some applications, it might be advantageous to update the default value.

CONFIG_ADVANCED CHIRPSTEPINC allows thus overwriting the default setting for the number of frequency step increments.

Please note that configuring the value 0x01 will increase the execution time of the command MEAS_ANT_IMP_ADVANCED.

Table 43. CHIRPSTEPINC (reset value 0x02)

Bit	Symbol	Access	Value	Description
7 to 4	RFU	W0		Reserved for future use
3 to 0	CHIRP_STEPINC[3:0]	W		Number of frequency step increments
			0x00	Reserved for future use
			0x01	1
			…	
			0x0A	10
			0x0B	Reserved for future use
			…	
			0x0F	Reserved for future use

CHIRPSTEPAFTPK:

The antenna resonance peak is detected if the chirp signals magnitude becomes maximum and if the signal values at the following frequency settings are smaller. The number of frequency steps for which the second condition is checked during the slow reverse chirp can be configured.

The initial number of frequency steps after the peak is optimized to cover typical applications. In some applications, it might be advantageous to update this number.

CONFIG_ADVANCED CHIRPSTEPAFTPK allows thus overwriting the default setting for the number of frequency steps after the peak.

Please note that configuring a value larger than 0x01 will increase the execution time of the command MEAS_ANT_IMP_ADVANCED.

Table 44. CHIRPSTEPAFTPK (reset value 0x01)

Bit	Symbol	Access	Value	Description
7 to 3	RFU	W0		Reserved for future use
2 to 0	CHIRP_STEPAFTPK[2:0]	W		Number of frequency steps after the peak
			0x00	Reserved for future use
			0x01	1
			…	
			0x05	5
			0x06	Reserved for future use
			0x07	Reserved for future use

CHIRPSTEPBCK:

The chirp signal remains at one frequency setting for a small number of cycles. Due to this, the maximum current will occur with a delay after the resonant frequency has been applied. At the time of the maximum current, the chirp signal has already moved on to higher frequencies.

In order to compensate this effect, the chirp frequency jumps back by 5 frequency steps (stepsize $\Delta f = f_C / 256 = 488.28125$ Hz) before the slow reverse chirp starts.

The initial number of frequency backsteps is optimized to cover typical applications. In some applications, it might be advantageous to update this number.

CONFIG_ADVANCED CHIRPSTEPBCK allows thus overwriting the default setting for the number of frequency backsteps.

Table 45. CHIRPSTEPBCK (reset value 0x05)

Bit	Symbol	Access	Value	Description
7 to 4	RFU	W0		Reserved for future use
3 to 0	CHIRP_STEPBCK[3:0]	W		Number of frequency steps jumped back
			0x00	Reserved for future use
			0x01	1
			…	
			0x05	5
			…	
			0x0A	10
			0x0B	Reserved for future use
			…	
			0x0F	Reserved for future use

CHIRPCYCLE:

After a peak has been detected with the fast forward chirp, the slow reverse chirp is started.

The initial number of cycles per frequency is optimized to cover typical applications. In some applications, it might be advantageous to update this number.

CONFIG_ADVANCED CHIRPCYCLE allows thus overwriting the default setting for the number of cycles per frequency.

Please note that the recommended value is 0x08 (differing to the reset value) and that a larger value will increase the execution time of the command MEAS_ANT_IMP_ADVANCED.

Table 46. CHIRPCYCLE (reset value 0x04)

Bit	Symbol	Access	Value	Description
7 to 5	RFU	W0		Reserved for future use
4 to 0	CHIRP_CYCLE[4:0] [1]	W		Number of cycles per frequency
			0x00	Reserved for future use
			0x01	5 or 6
			0x02	6 or 7
			0x03	7 or 8
			0x04	8 or 9
			…	
			0x1D	33 or 34
			0x1E	34 or 35

Bit	Symbol	Access	Value	Description
			0x1F	Reserved for future use

[1] The number of cycles per frequency may vary due to re-synchronization

2.3.2 Response

Table 47. CONFIG_ADVANCED response

LEN	CMD	STAT	CRC8
0x03	0xD2	STAT	CRC8

2.3.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 48. CONFIG_ADVANCED status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Number of chirp blanking samples 0 or larger than 4
		Number of frequency step increments 0 or larger than 10
		Number of frequency steps after the peak 0 or larger than 5
		Number of frequency steps jumped back 0 or larger than 10
		Number of cycles per frequency lower than “5 or 6” or larger than “34 or 35”

3. Device reset

3.1 SET_POR

SET_POR triggers a power on reset via SPI command. The device reset gets released after the response is sent. The SF_POR flag is set in the SPI response(s) after releasing the power on reset until clearing the flag by setting SC_RSTSW via the command CLEAR_POR_STATUS.

3.1.1 Command

Table 49. SET_POR command

LEN	CMD	CRC8
0x02	0xF4	CRC8

3.1.2 Response

Table 50.SET_POR response

LEN	CMD	STAT	CRC8
0x03	0xF4	STAT	CRC8

3.1.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

3.2 GET_POR_STATUS

GET_POR_STATUS reads the device power on reset flags. In case any of the flags is set, the INT pin is set to ‘high’.

3.2.1 Command

Table 51. GET_POR_STATUS command

LEN	CMD	CRC8
0x02	0x07	CRC8

The response contains the device reset flags. In case no power on reset has been detected, zero is returned in the corresponding parameter.

3.2.2 Response

Table 52. GET_POR_STATUS response

LEN	CMD	STAT	PARAM	CRC8
0x04	0x07	STAT	PORF	CRC8

PORF:

The reset flag register PORF signalizes the cause for the last power on reset event. Any write access is ignored.

Table 53. PORF (reset value 0x00)

Bit	Symbol	Access	Value	Description
7	RFU	R0		Reserved for future use
6	SF_CPUERR	R		Embedded µController access permission error
			0	Not asserted
			1	Asserted
5	SF_RAMERR	R		RAM parity error
			0	Not asserted
			1	Asserted
4	SF_WD	R		Watchdog Reset
			0	Not asserted
			1	Asserted
3	SF_CLKEXT	R		External Clock Error
			0	Not asserted
			1	Asserted
2	SF_RSTSW	R		Software Reset
			0	Not asserted
			1	Asserted
1	SF_RSTEXT	R		External Reset
			0	Not asserted
			1	Asserted
0	SF_LOWBAT	R		Low supply voltage
			0	Not asserted
			1	Asserted

3.2.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

3.3 CLEAR_POR_STATUS

CLEAR_POR_STATUS clears the device reset status flags, if marked.

3.3.1 Command

Table 54. CLEAR_POR_STATUS command

LEN	CMD	PARAM	CRC8
0x03	0x08	PORC	CRC8

PORC:

PORC specifies the device reset status flags to be cleared.

Table 55. PORC (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7	RFU	W0		Reserved for future use
6	SC_CPUERR	W		CPU Access Error
			0	No change
			1	Clear flag
5	SC_RAMERR	W		RAM parity error
			0	No change
			1	Clear flag
4	SC_WD	W		Watchdog Reset
			0	No change
			1	Clear flag
3	SC_CLKEXT	W		External Clock Error
			0	No change
			1	Clear flag
2	SC_RSTSW	W		Software Reset
			0	No change
			1	Clear flag
1	SC_RSTEXT	W		External Reset
			0	No change
			1	Clear flag
0	SC_LOWBAT	W		Low supply voltage
			0	No change
			1	Clear flag

3.3.2 Response

Table 56.CLEAR_POR_STATUS response

LEN	CMD	STAT	CRC8
0x03	0x08	STAT	CRC8

3.3.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 57. CLEAR_POR_STATUS status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	PORC set to 0

4. Operating states

4.1 START_SLEEP

START_SLEEP sets the device in SLEEP state. START_SLEEP is processed in IDLE and ERROR state only. SLEEP state is entered after sending the CRC8 in the command response.

4.1.1 Command

Table 58. START_SLEEP command

LEN	CMD	CRC8
0x02	0x50	CRC8

4.1.2 Response

Table 59. START_SLEEP response

LEN	CMD	STAT	CRC8
0x03	0x50	STAT	CRC8

4.1.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

4.2 START_SLEEP_FORCED

START_SLEEP_FORCED sets the device in SLEEP state similar to START_SLEEP, but differs in the following manner

- Processed in all device states
- SLEEP state is entered unconditionally
- Executed even if a non-blocking command is processed
- Wake-up events are ignored until SLEEP state is entered

Dependent on the device state, the boost converter and LF drivers are ramped-down after receiving the command.

SLEEP state is entered after sending the CRC8 in the command response.

4.2.1 Command

Table 60. START_SLEEP_FORCED command

LEN	CMD	CRC8
0x02	0x4F	CRC8

4.2.2 Response

Table 61. START_SLEEP_FORCED response

LEN	CMD	STAT	CRC8
0x03	0x4F	STAT	CRC8

4.2.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

5. Boost converter

5.1 CONFIG_BOOST

CONFIG_BOOST configures the initial boost converter coil current during voltage ramp-up and afterwards the maximum current in the boost converter coil (coil current limitation). Reducing the maximum current dependent on the applications need is advantageous for separating best fitting external components (coil, diode).

5.1.1 Command

Table 62. CONFIG_BOOST command

LEN	CMD	PARAM	CRC8
0x03	0x45	CURMAX	CRC8

CURMAX:

Table 63. CURMAX (reset value 0x0F)

Bit	Symbol	Access	Value	Description
7 to 4	CUR_INIT[3:0]	W		Initial boost converter current
			0000	5A
			0001	Reserved for future use
			0010	Reserved for future use
			0011	4 A
			0100	5 A
			…	…
			1110	15 A
			1111	16 A
3 to 0	CUR_MAX[3:0]	W		Maximum current in the boost converter coil
			0000	Reserved for future use
			0001	Reserved for future use
			0010	Reserved for future use
			0011	4 A
			0100	5 A
			…	…
			1110	15 A
			1111	16 A

CUR_INIT[3:0]:

CUR_INIT[3:0] determines the maximum initial boost converter coil current to charge the output capacitor and to reach the boost converter output voltage within the boost converter start-up time. After boost converter start-up, the boost converter coil current limitation is updated to the level defined by CUR_MAX[3:0].

CUR_MAX[3:0]:

CUR_MAX[3:0] limits the boost converter coil current after boost converter start-up. With the help of CUR_MAX external components (coil, diode) can be selected that fit best to the application.

5.1.2 Response

Table 64. CONFIG_BOOST response

LEN	CMD	STAT	CRC8
0x03	0x45	STAT	CRC8

5.1.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

6. LF antenna driver

6.1 CONFIG_LF_DRIVER

CONFIG_LF_DRIVER configures the LF divers. The configuration data is used when starting LF transmission.

The LF driver configuration and the low current driver settings are checked for consistency. In failure case an error code is returned without adopting the settings.

6.1.1 Command

Table 65. CONFIG_LF_DRIVER command

LEN	CMD	PARAM						CRC8
LEN	0x40	{DRIDi	DRPARi	CURSi	CURMi	0x40	DITHRi}	CRC8

DRIDi:

The driver (channel) ID DRIDi identifies the LF channel for which the parameter set is assigned.

DRPARi:

DRPARi configures the selected LF driver.

Table 66. DRPARi (reset value 0x14)

Bit	Symbol	Access	Value	Description
7 to 6	RFU	W0		Reserved for future use
5	PHINV	W		Invert driving phase
			0	Phase 0°
			1	Phase 180°
4	OPNLOOP	W		Open the antenna connection for fast antenna current decay
			0	Disable
			1	Enable
3 and 2	BDRATE[1:0]	W		LF driver baud rate
			00	2 kbit/s
			01	4 kbit/s
			10	8 kbit/s
			11	Reserved for future use
1	RFU	W0		Reserved for future use
0	MOD	W		Driver bridge operation mode
			0	Full bridge mode with mid level control
			1	Full bridge mode

PHINV:

Phase 0 means that driver P starts with driver ‘high’ phase.

Phase 180° means that driver N starts with driver ‘high’ phase.

OPNLOOP

Setting OPNLOOP accelerates the antenna current decay in the carrier off phase when sending modulated LF signals (but not at the end of an LF transmission sequence).

The acceleration of the antenna current ramp-down is realized by temporarily interrupting the antenna resonant circuit via the main driver switches.

BDRATE[1:0]:

If the LF driver baud rate is configured different for different drivers and these drivers are activated simultaneously, the baud rate value of the selected driver with the lowest number is used for all antennas.

MOD:

MOD determines the bridge operation mode of the LF drivers.

Consistency checks of the driver bridge settings are made when interrogating the commands

- START_LF_TRANSMIT
- START_LF_TRANSMIT_DATA
- MEAS_ANT_IMP
- START_DIAG
- CONFIG_WUP_POLLING
- CONFIG_TIMER_POLLING

CURSi:

CURSi is the current value to be adjusted for the selected driver in single channel operation.

It is important to notice that dependent on the antenna impedance the selected currents cannot always be reached in the application. The lower limits of the antenna current dependent on the antenna impedance magnitudes Z_{ANT} are specified in the product data sheet.

Table 67. CURSi (reset value 0x1F)

Bit	Symbol	Access	Value	Description
7 to 6	RFU	W0		Reserved for future use
5 to 0	CUR_S[5:0][1]	W		Driver current in single channel operation
			0x00	1 * 15.625 mA
			0x01	2 * 15.625 mA
			0x02	3 * 15.625 mA
			…	…
			0x3E	63 * 15.625 mA
			0x3F	64* 15.625 mA

[1] If a current value is selected that cannot be reached in the application due to the selected system parameters, the current control adjusts automatically to the nearest achievable current without further notice to the application

CURMi:

CURMi is the current value to be adjusted for the selected driver in multi channel operation.

It is important to notice that dependent on the antenna impedance the selected currents cannot always be reached in the application. The lower limits of the antenna current dependent on the antenna impedance magnitudes Z_{ANT} are specified in the product data sheet.

Table 68. CURMi (reset value 0x1F)

Bit	Symbol	Access	Value	Description
7 to 6	RFU	W0		Reserved for future use
5 to 0	CUR_M[5:0][1]	W		Driver current in multi channel operation
			0x00	1 * 15.625 mA
			0x01	2 * 15.625 mA
			0x02	3 * 15.625 mA
			…	…
			0x3E	63 * 15.625 mA
			0x3F	64* 15.625 mA

[1] If a current value is selected that cannot be reached in the application due to the selected system parameters, the current control adjusts automatically to the nearest achievable current without further notice to the application

DITHRi:

DITHR configures the dithering parameters for the selected driver.

Table 69. DITHRi (default value 0x02)

Bit	Symbol	Access	Value	Description
7 to 2	RFU	W0		Reserved for future use
1 to 0	DITHR[1:0]	W		Dithering range limit
			00	Dithering off
			01	Minimum range
			10	Medium range
			11	Maximum range

Example

In the following example, driver 1 and driver 2 are configured within one command.

Table 70. Configure driver 1 and driver 2 (full bridge mode) with CONFIG_LF_DRIVER command

LEN	CMD	PARAM												CRC8
0x0C	0x40	DRID1	DRPAR1	CURS1	CURM1	0x40	DITHR1	DRID2	DRPAR2	CURS2	CURM2	0x40	DITHR2	CRC8

6.1.2 Response

Table 71. CONFIG_LF_DRIVER response

LEN	CMD	STAT	CRC8
0x03	0x40	STAT	CRC8

6.1.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 72. CONFIG_LF_DRIVER status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Number of cascaded frames larger than number of drivers
		Duplicate DRID in set of cascaded frames

7. Parallel low current driver

7.1 CONFIG_LC_DRIVER

CONFIG_LC_DRIVER configures the low current LF drivers. The configuration data is used to adjust the currents transmitted via the low current LF drivers at the same time when transmitting data via the main LF drivers. The low current LF drivers are only active during data operation on the main LF drivers. During carrier off times and when sending constant carrier signals via the main LF drivers, no low current signals are sent.

The low current driver settings and the LF driver configuration are checked for consistency. In failure case an error code is returned without adopting the settings.

7.1.1 Command

Table 73. CONFIG_LC_DRIVER command

LEN	CMD	PARAM		CRC8
LEN	0x44	{DRIDi	LCCURi}	CRC8

DRIDi:

The driver (channel) ID identifies the LF channel for which the parameter set is assigned.

LCCURi:

LCCUR configures the low current signal of the selected channel DRID which is driven when other channels are driven with high power. The current is adjusted via the duty cycle of the low current driver (step size 2.5%).

Table 74. LCCURi (reset value 0x0F)

Bit	Symbol	Access	Value	Description
7 to 5	RFU	W0		Reserved for future use
4 to 0	LCDCY[4:0]	W		Duty cycle for low current driver
			0x00	10 %
			0x01	12.5 %
			…	…
			0x1E	85 %
			0x1F	87.5 %

7.1.2 Response

Table 75. CONFIG_LC_DRIVER response

LEN	CMD	STAT	CRC8
0x03	0x44	STAT	CRC8

7.1.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 76. CONFIG_LC_DRIVER status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Number of cascaded frames larger than number of drivers
		Duplicate DRID in set of cascaded frames

8. Telegram sequencer

8.1 SET_LF_DATA

SET_LF_DATA stores in the battery backed RAM the user defined LF telegram data to be transmitted when e,g. executing the command START_LF_TRANSMIT.
SET_LF_DATA specifies data for a particular segment of the LF telegram, identified by a segment number.

8.1.1 Command

Table 77. SET_LF_DATA command

LEN	CMD	PARAM				CRC8
LEN	0x41	DATAID	DATACFG	DATBLEN	{DATAi}	CRC8

DATAID:

DATAID is the number that identifies which segment is affected by this command. DATAID is limited to be between 0 and 59.

The data identifiers listed in Table 78 are pre-configured by default and can be used as specified without the need to be stored by the application before. The application can overwrite the pre-configured data with the help of the SET_LF_DATA command.

Table 78. Pre-configured DATAID after device reset (default values)

DATAID	DATACFG	DATABLEN	DATA1	DATA2	DATA3	Description
0x00	0x07	0x31	-	-	-	5 ms constant carrier
0x01	0x07	0x27	-	-	-	4 ms constant carrier
0x02	0x06	0x0E	-	-	-	1.5 ms carrier off
0x03	0x06	0x05	-	-	-	0.6 ms carrier off
0x04	0x06	0x03	-	-	-	0.4 ms carrier off
0x05	0x06	0x04	-	-	-	0.5 ms carrier off
0x35	0x00	0x08	0x00	-	-	Preamble
0x36	0x02	0x12	0xE2	0xCC	0x80	Code violation

DATACFG:

The data configuration applies to the segment identified by DATAID and specifies the coding for this data frame. This can be e.g. Manchester code or a special coding for sending a code violation. Additionally, driver off times and constant carrier signals can be specified.

Table 79. DATACFG (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7 to 3	RFU[4:0]	W0		Reserved for future use
2 to 0	DATA_CFG[2:0]	W		Data coding
			000	Manchester
			001	Bit wise (NRZ)
			010	Half bit wise, double transition speed (code violation)
			011	Reserved for future use

Bit	Symbol	Access	Value	Description
			100	Reserved for future use
			101	Reserved for future use
			110	Carrier off
			111	Constant carrier

DATBLEN:

For coded data frames (e.g. Manchester code), the data bit length specifies the number of bits to be sent. DATBLEN is limited to be between 0 and 120.

Table 80. DATBLEN for DATA_CFG = 000, 001 or 010 (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7 to 0	DUR[7:0]	W		Number of bits to be sent
			0x00	0 bit
			0x01	1 bit
			…	…
			0x77	119 bit
			0x78	120 bit
			0x79	Reserved for future use
			…	
			0xFF	Reserved for future use

For the special cases of a switched-off or constant carrier (DATA_CFG = 110 or 111), DATBLEN specifies the duration of the signal.

Table 81. DATBLEN for DATA_CFG = 110 or 111 (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7 to 0	DUR[7:0]	W		Duration of the carrier signal (steps a 100 μs)
			0x00	1 * 100 μs = 0.1 ms
			0x01	2 * 100 μs = 0.2 ms
			…	…
			0xFE	255 * 100 μs = 25.5 ms
			0xFF	Permanent off (DATA_CFG = 110) Permanent on (DATA_CFG = 111)

DUR[7:0]

The duration setting of the carrier signal is valid for LF transmission at one channel. In case of switching between different antennas within one LF transmission sequence, the LF transmission on the other antenna continues after the LF channel switching time ($t_{CH,CHG}$).

In case of setting “Permanent on” or “Permanent off”, the LF transmission has to be stopped via STOP_LF_TRANSMIT.

DATAi:

Each bit position within a DATAi byte represents one data bit. Each data bit is sent in the coding specified in DATACFG. The data bit handling takes place according to section 1.4.5.

8.1.2 Response

Table 82. SET_LF_DATA response

LEN	CMD	STAT	CRC8
0x03	0x41	STAT	CRC8

8.1.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 83. SET_LF_DATA status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	DATAi is included for a switched-off carrier or for a constant carrier (DATA_CFG = 110 or 111)

8.2 SET_LC_DATA

SET_LC_DATA specifies the data source for the data transmitted via the low current LF drivers at the same time when transmitting data via the main LF drivers. The low current LF drivers are only active during data operation on the main LF drivers. During carrier off times and when sending constant carrier signals via the main LF drivers, no low current signals are sent.

8.2.1 Command

Table 84. SET_LC_DATA command

LEN	CMD	PARAM			CRC8
LEN	0x47	LCPAR	LCDATBLEN	{LCDATAi}	CRC8

LCPAR:

The data to be sent via the low current drivers can be either the inverted version of the main data pattern or user defined data stored in the battery backed RAM.

If user defined data is selected, the data itself is specified in LCDATAi.

If inverted data is selected, parameters LCDATBLEN and LCDATAi can be omitted. If set albeit, the length consistency of LCDATBLEN with regard to LCDATAi is checked. If the check is successful, the parameters will be ignored, while the inverted data is selected. If the length consistency check fails, the status flag SF_PAR is set in the command response.

Table 85. LCPAR (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 1	RFU[6:0]	W0		Reserved for future use
0	LCSRC	W		Low current data source
			0	Inverted version of main data

Bit	Symbol	Access	Value	Description
			1	User defined data

LCDATBLEN:

LCDATBLEN specifies the length in bits of the pre-defined low current driver data LCDATAi to be sent. LCDATBLEN is limited to be between 0 and 120.

The number of physically transmitted LC data bits is defined by the number of data bits transmitted via the main LF driver DATBLEN.

If both LCDATBLEN and DATBLEN are a multiple of 8, for each byte of the main driver a LCDATAi byte is assigned and send. If LCDATBLEN is larger than DATBLEN, only the first bytes defined by LCDATAi are used. If LCDATBLEN is smaller than DATBLEN, after transmitting all pre-defined LC data bytes the transmission starts again at the first LC byte.

If LCDATBLEN is not a multiple of 8 bit, the missing bits of LCDATAi are padded with 0 and are assigned dependent on LCDATABLEN and DATBLEN (see Fig 8 and Fig 9).

Configuration of 18 LC data bit
SET_LC_DATA(LEN, 0x47, 1, 18, 0xE6, 0xC5, 0x02)

	LCDATA1	LCDATA2	LCDATA3
LC data	0xE6	0xC5	0x02
LC data	0b11100110	0b11000101	0b10

Send 32 bit after interogating
SET_LF_DATA(LEN, 0x41, 0x00, 0x01, 32, 0xFF, 0xFF, 0xFF, 0xFF, CRC8)

	DATA1	DATA2	DATA3	DATA4
Main LF driver	0b1111 1111	0b1111 1111	0b1111 1111	0b1111 1111
LC driver	0b1110 0110	0b1100 0101	0b1110 0110	0b1100 0101
	Start with LC DATA1	Send LC DATA2	Repeat LC DATA1	Repeat LC DATA2

Send 21 bit after interogating
SET_LF_DATA(LEN, 0x41, 0x00, 0x01, 21, 0xFF, 0xFF, 0xFF, 0xFF, CRC8)

	DATA1	DATA2	DATA3
Main LF driver	0b1111 1111	0b1111 1111	0b1111 1
LC driver	0b1110 0110	0b1100 0101	0b1110 0
	Start with LC DATA1	Send LC DATA2	Repeat LC DATA1, use 5 first bits

Send 19 bit after interogating
SET_LF_DATA(LEN, 0x41, 0x00, 0x00, 0x01,19, 0xFF, 0xFF, 0xFF, CRC8)

Main LF driver	0b1111 1111	0b1111 1111	0b111
LC driver	0b1110 0110	0b1100 0101	0b111
	Start with LC DATA1	Send LC DATA2	Repeat LC DATA1, use 3 first bits

Send 18 bit after interogating
SET_LF_DATA(LEN, 0x41, 0x00, 0x00, 0x01,18, 0xFF, 0xFF, 0xFF, CRC8)

Main LF driver	0b1111 1111	0b1111 1111	0b11
LC driver	0b1110 0110	0b1100 0101	0b10
	Start with LC DATA1	Send LC DATA2	Send LCDATA3

Send 17 bit after interogating
SET_LF_DATA(LEN, 0x41, 0x00, 0x00, 0x01,17, 0xFF, 0xFF, 0xFF, CRC8)

Main LF driver	0b1111 1111	0b1111 1111	0b1
LC driver	0b1110 0110	0b1100 0101	0b1
	Start with LC DATA1	Send LC DATA2	Send LCDATA3, use 1st bit

Send 12 bit after interogating
SET_LF_DATA(LEN, 0x41, 0x00, 0x00, 0x01, 12, 0xFF, 0xFF)

Main LF driver	0b1111 1111	0b1111
LC driver	0b1110 0110	0b1100
	Start with LC DATA1	Send LC DATA2, use 4 first bits

Fig 8. Examples for low current driver data sent dependent on main driver data length, LCDATBLEN = 18

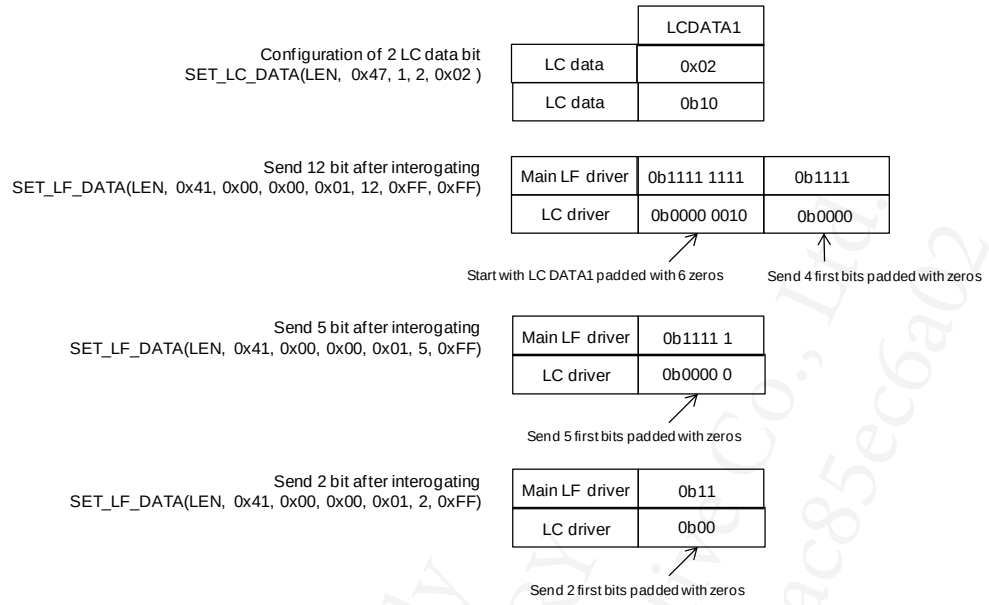

Fig 9. Examples for low current driver data sent dependent on main driver data length, LCDATBLEN = 2

LCDATAi:

Each bit position within an LCDATAi byte represents one data bit. Each data bit is sent via the low current driver in the coding (e.g. Manchester code or half-bit code) according to the configuration of the data bit sent in parallel via the main LF driver.

If the main driver either switches on constant carrier or switches off the carrier, no LCDATAi bit is sent.

Each time a main driver is activated, for every data ID transmitted by the main driver the low current data starts from the first bit. If more data bit are sent via the main driver than specified via LCDATBLEN, the LCDATAi bits will be repeated again from the beginning until the main driver ends data transmission.

8.2.2 Response

Table 86. SET_LC_DATA response

LEN	CMD	STAT	CRC8
0x03	0x47	STAT	CRC8

8.2.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 87. SET_LC_DATA status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Length consistency check of LCDATBLEN with regard to LCDATAi ifailed

8.3 START_LF_TRANSMIT

START_LF_TRANSMIT starts the LF transmission using the pre-loaded configuration and LF telegram data. The data transmission starts with a delay of $t_{LF,START}$ after the last byte (CRC8) of the incoming frame is received. The LF power path activation is started before sending the SPI response.

Before starting the driver, the following settings are checked for availability and consistency

- Clock settings
- Antenna impedance settings (boost drive)
- LF driver configuration
- Low current driver configuration

Additionally, all protection and diagnostics flags are controlled. If multiple antenna use is configured, the settings for the boost converter and duty cycle(s) are calculated.

After completion, the operation status flag SF_TXREADY is set. Further, the LF field is switched-off, except in case a permanent constant carrier has been configured (DATACFG[2:0] = 0b111, DUR[7:0] = 0xFF).

In case of a failure, an error code is returned and no transmission is performed. For example, when interrogating START_LF_TRANSMIT while V_{BAT} is below the undervoltage shutdown detection threshold, the status flag SF_CMD flag is set.

If the LF field is on already when receiving this command, the status flag SF_CMD is set in the command response.

8.3.1 Command

Table 88. START_LF_TRANSMIT command

LEN	CMD	PARAM						CRC8
LEN	0x42	{DRPi	RFU	LCDRPi	RFU	LENDATAIDi	{DATAIDk}}	CRC8

DRPi:

DRPi holds the information which drivers to be activated. In case only one driver is selected, the respective current setting for single channel is used.

Activating 2 or 3 drivers simultaneously, the current setting for multi channel operation is taken. The same LF telegram data is sent via the simultaneously driven channels.

Setting DRPi to 0, the status flag SF_PAR is set.

LCDRPi:

LCDRPi selects the low current LF drivers to be activated in parallel to the main drivers. The low current values used for the selected channels are specified before via the CONFIG_LC_DRIVER command.

LENDATAIDi:

LENDATAIDi specifies the number of identifiers k that follows in DATAIDik.

DATAIDik:

DATAIDik represents the identifiers of the data set to be sent.

If PREAMB is set (command CONFIG_DEVICE), the standard NXP preamble and code violation pattern is sent first, followed by the LF telegram data selected via DATAIDik. If PREAMB is not set, the data selected via DATAIDik is sent directly.

One DATAIDik is mandatory. If more than one DATAIDik is used, the information is queued and the data packets are sent in the selected order without interruption of the data stream.

The data to be transmitted is taken as stored with SET_LF_DATA and interpreted in the configured coding.

8.3.2 Response

Table 89. START_LF_TRANSMIT response

LEN	CMD	STAT	CRC8
0x03	0x42	STAT	CRC8

8.3.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 90. START_LF_TRANSMIT status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Selected DATAID not configured
		Number of selected main drivers 0 or larger than 3
		Same driver(s) used in normal operation and in low current operation

8.4 START_LF_TRANSMIT_DATA

START_LF_TRANSMIT_DATA starts the LF transmission. The data transmission starts with a delay of $t_{LF,START}$ after the last byte (CRC8) of the incoming frame is received. The LF power path activation is started after preparing the SPI response to be sent.

Before starting the driver, the following settings are checked for availability and consistency

- Clock settings
- Antenna impedance settings (boost drive)
- LF driver configuration
- Low current driver configuration

Additionally, all protection and diagnostics flags are controlled. If multiple antenna use is configured, the settings for the boost converter and duty cycle(s) are calculated.

After completion, the operation status flag SF_TXREADY is set and the LF field is switched-off.

In case of a failure, an error code is returned and no transmission is performed. For example, when interrogating START_LF_TRANSMIT_DATA while V_{BAT} is below the undervoltage shutdown detection threshold, the status flag SF_CMD flag is set.

If the LF field is on already when receiving this command, the status flag SF_CMD is set in the command response.

8.4.1 Command

Table 91. START_LF_TRANSMIT_DATA command

LEN	CMD	PARAM							CRC8
LEN	0x46	{DRPi	RFU	LCDRPi	RFU	DATBLEN_Li	DATBLEN_Hi	{DATAIik}}	CRC8

DRPi:

DRPi holds the information which drivers to be activated. In case only one driver is selected, the respective current setting for single channel is used.

Activating 2 or 3 drivers simultaneously, the current setting for multi channel operation is taken. The same LF telegram data is sent via the simultaneously driven channels.

Setting DRPi to 0 the status flag SF_PAR is set.

LCDRPi:

LCDRPi select the low current LF drivers to be activated in parallel to the main drivers. The low current values used for the selected channels are specified before via the CONFIG_LC_DRIVER command.

DATBLENi:

The data bit length specifies the length k of the data to be sent in bits. The maximum length is 511 bit.

Table 92. DATBLENi (reset value 0xXXXX)

Bit	Symbol	Access	Value	Description
15 to 9	RFU	W0		Reserved for future use
8 to 0	DATBLEN [8:0]	W		Data bit length
			0x000	Reserved for future use
			0x001	1 bit
			0x002	2 bit
			…	
			0x1FF	511 bit

DATAik:

DATAik represents the data set to be sent.

If PREAMB is set (command CONFIG_DEVICE), the standard NXP preamble and code violation pattern is sent first, followed by the attached LF telegram data. If PREAMB is not set, the attached LF telegram data is sent directly.

One DATAik byte is mandatory. Each bit position within a DATAik byte represents one data bit. Each data bit is sent in Manchester coding. The non-used bits at the end should be padded with 0 until the next byte border. They will be ignored during transmission.

If more than one DATAik byte is used, the information is queued and the data packets are sent in the selected order without interruption of the data stream.

8.4.2 Response

Table 93. START_LF_TRANSMIT_DATA response

LEN	CMD	STAT	CRC8
0x03	0x46	STAT	CRC8

8.4.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 94. START_LF_TRANSMIT_DATA status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Number of selected main drivers 0 or larger than 3
		Same driver(s) used in normal operation and in low current operation

8.5 STOP_LF_TRANSMIT

STOP_LF_TRANSMIT is valid in PKE state only. STOP_LF_TRANSMIT stops the LF transmission after the CRC8 of the incoming frame is received and validated. The SPI response is sent afterwards. All drivers are switched off. The command is intended to switch off a permanently activated constant carrier configured via SET_LF_DATA (DATA_CFG[2:0] = 0b111, DUR[7:0] = 0xFF).

8.5.1 Command

Table 95. STOP_LF_TRANSMIT command

LEN	CMD	CRC8
0x02	0x43	CRC8

8.5.2 Response

Table 96. STOP_LF_TRANSMIT response

LEN	CMD	STAT	CRC8
0x03	0x43	STAT	CRC8

8.5.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

9. Immobilizer

9.1 CONFIG_IMMO_DRIVER

CONFIG_IMMO_DRIVER configures the immobilizer antenna driver.

9.1.1 Command

Table 97. CONFIG_IMMO_DRIVER command

LEN	CMD	PARAM		CRC8
0x04	0x60	TXPAR	RXPAR	CRC8

Table 98. TXPAR (reset value 0x5F)

Bit	Symbol	Access	Value	Description
7	RFU	W0		Reserved for future use
6	OPNLOOP	W		Open the antenna connection for fast antenna current decay
			0	Disabled
			1	Enabled
5 to 0	TXCUR [5:0][1]	W		Current during immobilizer transmission
			0x00	1 * 15.625 mA
			0x01	2 * 15.625 mA
			…	…
			0x3E	63* 15.625 mA
			0x3F	64 * 15.625 mA

[1] If a current value is selected that cannot be reached in the application due to the selected system parameters, the current control adjusts automatically to the nearest achievable current without further notice to the application

OPNLOOP

The antenna current decay is accelerated by opening the driver switches at one side of the antenna.

TXCUR:

TXCUR defines the current during immobilizer transmission.

It is important to notice that dependent on the antenna impedance the selected currents cannot always be reached in the application. The lower limits of the antenna current dependent on the antenna impedance magnitudes Z_{ANT} are specified in the product data sheet.

Table 99. RXPAR (reset value 0x1F)

Bit	Symbol	Access	Value	Description
7 to 6	RFU	W0		Reserved for future use
5 to 0	RXCUR [5:0][1]	W		Current during immobilizer reception
			0x00	1 * 15.625 mA
			0x01	2 * 15.625 mA
			…	…
			0x3E	63* 15.625 mA
			0x3F	64 * 15.625 mA

[1] If a current value is selected that cannot be reached in the application due to the selected system parameters, the current control adjusts automatically to the nearest achievable current without further notice to the application

RXCUR:

RXCUR defines the current during immobilizer reception.

If RXCUR is unequal to TXCUR the current levels will be changed to the RXCUR settings after the stop condition of the BPLM modulator (Fig 10).

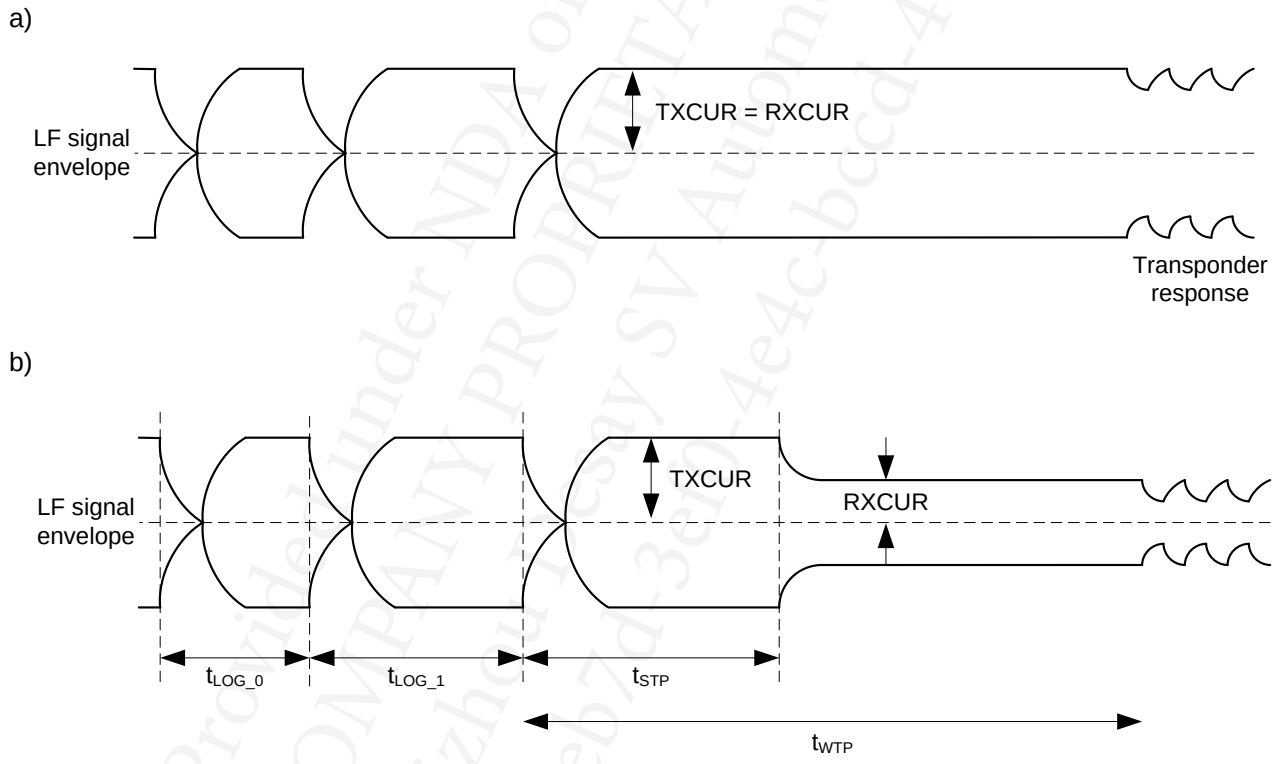

Fig 10. Transmit current TXCUR and receive current RXCUR with identical (a) and different settings (b)

It is important to notice that dependent on the antenna impedance the selected currents cannot always be reached in the application. The lower limits of the antenna current dependent on the antenna impedance magnitudes Z_{ANT} are specified in the product data sheet.

9.1.2 Response

Table 100. CONFIG_IMMO_DRIVER response

LEN	CMD	STAT	CRC8
0x03	0x60	STAT	CRC8

9.1.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

9.2 CONFIG_IMMO_BPLM

CONFIG_IMMO_BPLM configures the binary pulse length modulator timings. The settings need to be done in accordance with the transponder specification.

9.2.1 Command

Table 101. CONFIG_IMMO_BPLM command

LEN	CMD	PARAM		CRC8
0x04	0x61	TLOG	MPT	CRC8

TLOG:

Defines the BPLM pulse length for logic “0” and “1”

Table 102. TLOG (Reset value 0x01)

Bit	Symbol	Access	Value	Description
7 to 4	TLOG_1[3:0]	W		BPLM pulse time for logic 1
			0x0	28 T0
			0x1	29 T0
			0x2	30 T0
			0x3	31 T0
			0x4	32 T0
			0x5	33 T0
			0x6	34 T0
			0x7	35 T0
			0x8	36 T0
			0x9	37 T0
			0xA	38 T0
			0xB	39 T0
			0xC	40 T0
			0xD	41 T0
			0xE	42 T0
			0xF	43 T0
3 to 0	TLOG_0[3:0]	W		BPLM pulse time for logic 0
			0x0	19 T0
			0x1	20 T0
			0x2	21 T0
			0x3	22 T0
			0x4	23 T0
			0x5	24 T0
			0x6	25 T0
			0x7	26 T0
			0x8	27 T0
			0x9	28 T0
			0xA	29 T0
			0xB	30 T0
			0xC	31 T0
			0xD	32 T0
			0xE	33 T0
			0xF	34 T0

MPT:

Defines low duration of write pulse and stop pulse length

Table 103. MPT (Reset value 0x68)

Bit	Symbol	Access	Value	Description
7 to 4	TSTOP [3:0]	W		Duration of stop pulse
			0x0	Reserved for future use
			0x1	Reserved for future use
			0x2	Reserved for future use
			0x3	Reserved for future use
			0x4	36 T0
			0x5	37T0
			0x6	38 T0
			0x7	39 T0
			0x8	40 T0
			0x9	41 T0
			0xA	42 T0
			0xB	43 T0
			0xC	44 T0
			0xD	45 T0
			0xE	46 T0
			0xF	47 T0
3 to 0	TWRP [3:0]	W		Low duration of write pulse
			0x0	Reserved for future use
			0x1	Reserved for future use
			0x2	Reserved for future use
			0x3	Reserved for future use
			0x4	4 T0
			0x5	5 T0
			0x6	6 T0
			0x7	7 T0
			0x8	8 T0
			0x9	9 T0
			0xA	10 T0
			0xB	11 T0
			0xC	12 T0
			0xD	13 T0
			0xE	14 T0
			0xF	15 T0

Fig 11 illustrates the different protocol timings that are configurable via SPI commands.

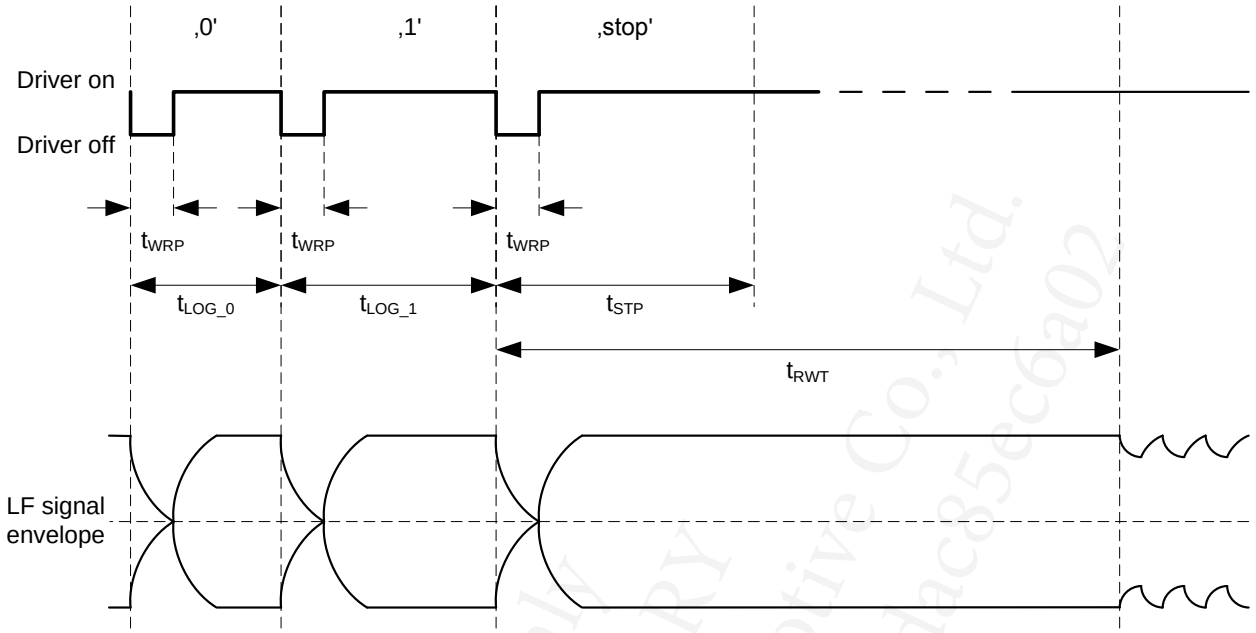

Fig 11. BPLM data coding

9.2.2 Response

Table 104. CONFIG_IMMO_BPLM response

LEN	CMD	STAT	CRC8
0x03	0x61	STAT	CRC8

9.2.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

9.3 CONFIG_IMMO_RECEIVER

CONFIG_IMMO_RECEIVER configures the immobilizer receiver. The settings need to be done in accordance with the transponder specification.

9.3.1 Command

Table 105. CONFIG_IMMO_RECEIVER command

LEN	CMD	PARAM		CRC8
0x04	0x62	RXCFG	TSYNC	CRC8

RXCFG:

RXCFG defines the equalizer pattern and the receiver wait time for a transponder response.

Table 106. RXCFG (reset value 0x01)

Bit	Symbol	Access	Value	Description
7	RFU	W0		Reserved for future use
6 to 2	TRWT[4:0]	W		Receiver wait time for transponder response
			0x00	1 * 202 T0 + TSYNC
			0x01	2 * 202 T0 + TSYNC
			…	…
			0x1F	32 * 202 T0 + TSYNC
1	RFU	W0		Reserved for future use
0	EQU	W		Equalizer pattern configuration
			0	EQ pattern (5x “1”)
			1	EQM pattern (6x “1” + 1x “0”)

TRWT:

TRWT is the time the receiver has to wait until the transponder sends back the response. During this time, the receiver is made sensitive for the incoming data.

TSYNC:

TSYNC is added to TRWT and intended for fine adjustment of TRWT.

Table 107. TSYNC (reset value 0x10)

Bit	Symbol	Access	Value	Description
7 to 5	RFU	W0		Reserved for future use
4 to 0	TSYNC[4:0]	W		Receiver wait time for transponder response
			0x00	-16 T0
			0x01	-15 T0
			…	…
			0x0F	-1 T0
			0x10	0 T0
			0x11	1 T0
			0x1F	15 T0

9.3.2 Response

Table 108. CONFIG_IMMO_RECEIVER response

LEN	CMD	STAT	CRC8
0x03	0x62	STAT	CRC8

9.3.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

9.4 START_IMMO

START_IMMO starts the immobilizer driver and receiver and transmits a constant carrier signal with the antenna current configured for immobilizer transmission. The transmission

starts with a delay of $t_{IMMO,START}$ after the CRC8 of the incoming frame is received and validated.

9.4.1 Command

Table 109. START_IMMO command

LEN	CMD	CRC8
0x02	0x63	CRC8

9.4.2 Response

Table 110. START_IMMO response

LEN	CMD	STAT	CRC8
0x03	0x63	STAT	CRC8

9.4.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 111. START_IMMO status flags

Status byte	Status bit	Event or failure condition
STAT	SF_LAST_OP	Failure during LF power path activation (caused by START_IMMO command itself)

9.5 STOP_IMMO

STOP_IMMO stops the constant carrier signal transmitted by the immobilizer driver. The transmission stops after the CRC8 of the incoming frame is received and validated. The immobilizer receiver is also stopped.

If the LF field is off already when receiving this command, the LF field is kept switched off.

9.5.1 Command

Table 112. STOP_IMMO command

LEN	CMD	CRC8
0x02	0x64	CRC8

9.5.2 Response

Table 113. STOP_IMMO response

LEN	CMD	STAT	CRC8
0x03	0x64	STAT	CRC8

9.5.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

Note: STOP_IMMO itself does not set any operation status flag OPF.

9.6 START_IMMO_TRANSMIT

START_IMMO_TRANSMIT writes data to the transponder. The BPLM modulator is started after the CRC8 of the incoming frame is received and validated. The BPLM is started before sending the SPI response.

After completion, the operation status flag SF_TXREADY is set.

This command can only be used after the START_IMMO command because this enables the constant carrier signal that is required for the communication. After completion of START_IMMO_TRANSMIT the constant carrier is not switched off.

If data should be received from the transponder, the command START_IMMO_TRANSCEIVE has to be used.

9.6.1 Command

Table 114. START_IMMO_TRANSMIT command

LEN	CMD	PARAM		CRC8
LEN	0x66	DATBLEN	{DATAi}	CRC8

DATBLEN:

The data bit length specifies the length of the data to be sent in bits.

Table 115. DATBLEN (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7 to 0	DATBLEN[7:0]	W		Number of bits to be sent
			0x00	Reserved for future use
			0x01	1 bit
			…	…
			0xFE	254 bit
			0xFF	255 bit

DATAi:

Each bit position within a DATAi byte represents one data bit. Each data bit is sent in BPLM coding. The bit handling takes place according to section 1.4.5.

9.6.2 Response

Table 116. START_IMMO_TRANSMIT response

LEN	CMD	STAT	CRC8
LEN	0x66	STAT	CRC8

9.6.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

9.7 START_IMMO_TRANSCEIVE

START_IMMO_TRANSCEIVE writes to the transponder and afterwards reads the response from the transponder within one command. The BPLM modulator is started after the last byte (CRC8) of the incoming frame is received and validated. The BPLM is started before sending the SPI response.

After the BPLM stop condition the antenna current is changed to the configured immobilizer reception current (if different to the immobilizer transmission current) and after the configured receiver wait time for transponder response (TRWT[4:0]) the receiver will be made sensitive for the incoming data. The command defines the number of data bits that will be received. If this bit number is unequal to a multiple of 8, always complete bytes are received. Missing bits are received as noise signal of the LF channel.

After data reception, the antenna current is changed back to the configured immobilizer transmission current (if different to the reception current) and after completion, the operation status flag SF_RXREADY is set.

This command can only be used after the START_IMMO command because this enables the constant carrier signal that is required for the communication. After completion of START_IMMO_TRANSCEIVE, the constant carrier is not switched off.

9.7.1 Command

Table 117. START_IMMO_TRANSCEIVE command

LEN	CMD	PARAM				CRC8
LEN	0x65	DATBLEN	{DATAi}	RXLEN_L	RXLEN_H	CRC8

DATBLEN:

The data bit length specifies the length of the data to be sent in bits.

Table 118. DATBLEN (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7 to 0	DATBLEN[7:0]	W		Number of bits to be sent
			0x00	Reserved for future usev
			0x01	1 bit
			…	…
			0xFE	254 bit
			0xFF	255 bit

DATAi:

Each bit position within a DATAi byte represents one data bit. Each data bit is sent in BPLM coding. The bit handling takes place according to section 1.4.5.

RXLEN:

RXLEN indicates the number of bits expected to be received.

Table 119. RXLEN (reset value 0xXXXX)

Bit	Symbol	Access	Value	Description
15 to 9	RFU	W0		Reserved for future use
8 to 0	RXLEN[8:0]	W		Number of received bits
			0x000	Reserved for future use
			0x001	1 bit
			0x002	2 bit
			…	
			0x1FF	511 bit

9.7.2 Response

Table 120. START_IMMO_TRANSCEIVE response

LEN	CMD	STAT	CRC8
0x03	0x65	STAT	CRC8

9.7.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

9.8 GET_IMMO_RESPONSE

GET_IMMO_RESPONSE reads the given number of bytes from the receive data buffer. This command will be used after each execution of the START_IMMO_TRANSCEIVE command.

9.8.1 Command

Table 121. GET_IMMO_RESPONSE command

LEN	CMD	CRC8
0x02	0x67	CRC8

9.8.2 Response

Table 122. GET_IMMO_RESPONSE response

LEN	CMD	STAT	PARAM		CRC8
LEN	0x67	STAT	IMMOF	{DATAi}	CRC8

IMMOF:

IMMOF specifies the immobilizer data transmission error flags.

Table 123. IMMOF (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 1	RFU	R0		Reserved for future use
0	SF_OVRDRV	R		Immo RX ADC overdrive status
			0	No malfunction
			1	Malfunction

SF_OVRDRV:

SF_OVRDRV is set if the IMMO receiver has detected a clipping of the received input signal. In this case the data received from the transponder has to be seen as corrupted.

DATAi:

Each DATAi byte contains the Manchester decoded response sent by the transponder. The bit handling takes place according to section 1.4.5 .

9.8.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 124. GET_IMMO_RESPONSE status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Data reception not successful

9.9 CLEAR_IMMO_STATUS

CLEAR_IMMO_STATUS clears the immobilizer data transmission error flags.

9.9.1 Command

Table 125. CLEAR_IMMO_STATUS command

LEN	CMD	PARAM	CRC8
0x03	0x68	IMMOC	CRC8

IMMOC:

IMMOC specifies the immobilizer data transmission error flags to be cleared.

Table 126. IMMOC (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7 to 1	RFU	W0		Reserved for future use
0	SC_OVRDRV	W		Immo RX ADC overdrive status
			0	No change
			1	Clear flag

9.9.2 Response

Table 127.CLEAR_IMMO_STATUS response

LEN	CMD	STAT	CRC8
0x03	0x68	STAT	CRC8

9.9.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 128. CLEAR_IMMO_STATUS status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	IMMOC set to 0

9.10 SET_IMMO_MASK

SET_IMMO_MASK enables the triggering of the INT pin if the immobilizer data transmission error flag is set.

9.10.1 Command

Table 129. SET_IMMO_MASK command

LEN	CMD	PARAM	CRC8
0x03	0x69	IMMOM	CRC8

IMMOM:

IMMOM provides the mask to enable the INT pin if the immobilizer data transmission error flag is set. The immobilizer data transmission error flag itself is not affected.

Table 130. IMMOM (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 1	RFU	W0		Reserved for future use
0	SM_OVRDRV	W		Set INT pin if immo RX ADC overdrive status is set
			0	Disable
			1	Enable

9.10.2 Response

Table 131.SET_IMMO_MASK response

LEN	CMD	STAT	CRC8
0x03	0x69	STAT	CRC8

9.10.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

10. Antenna parameter

10.1 MEAS_ANT_IMP

MEAS_ANT_IMP measures for each selected channel the antenna parameters impedance ZANT and phase shift PHA. The boost converter and the LF driver are activated for this.

The LF driver settings shall be in line with the application before starting an antenna impedance measurement.

The impedance measurement is performed with a fixed driver voltage V_{DR} = 20 V.

It starts with a driver duty cycle of 30%. If the first measured antenna driver current is less than 250 mA, the driver duty cycle is set to 80% (Fig 12).

Using the identified driver duty cycle the antenna driver current peak is determined. If the peak current is equal or larger 100 mA and the command completes successfully, the measured values are stored in the battery backed RAM. These values are automatically used for following boost converter settings. If the maximum measured antenna driver current is below 100 mA (e.g. if no antenna is connected), ZANT and PHA are set to their reset values.

After completion of MEAS_ANT_IMP, the operation status flag SF_IMPMEAS is set. The values can be read via the GET_ANT_IMP command.

If SET_ANT_IMP or MEAS_ANT_IMP_ADVANCED is interrogated afterwards, the values formerly measured via MEAS_ANT_IMP are overwritten.

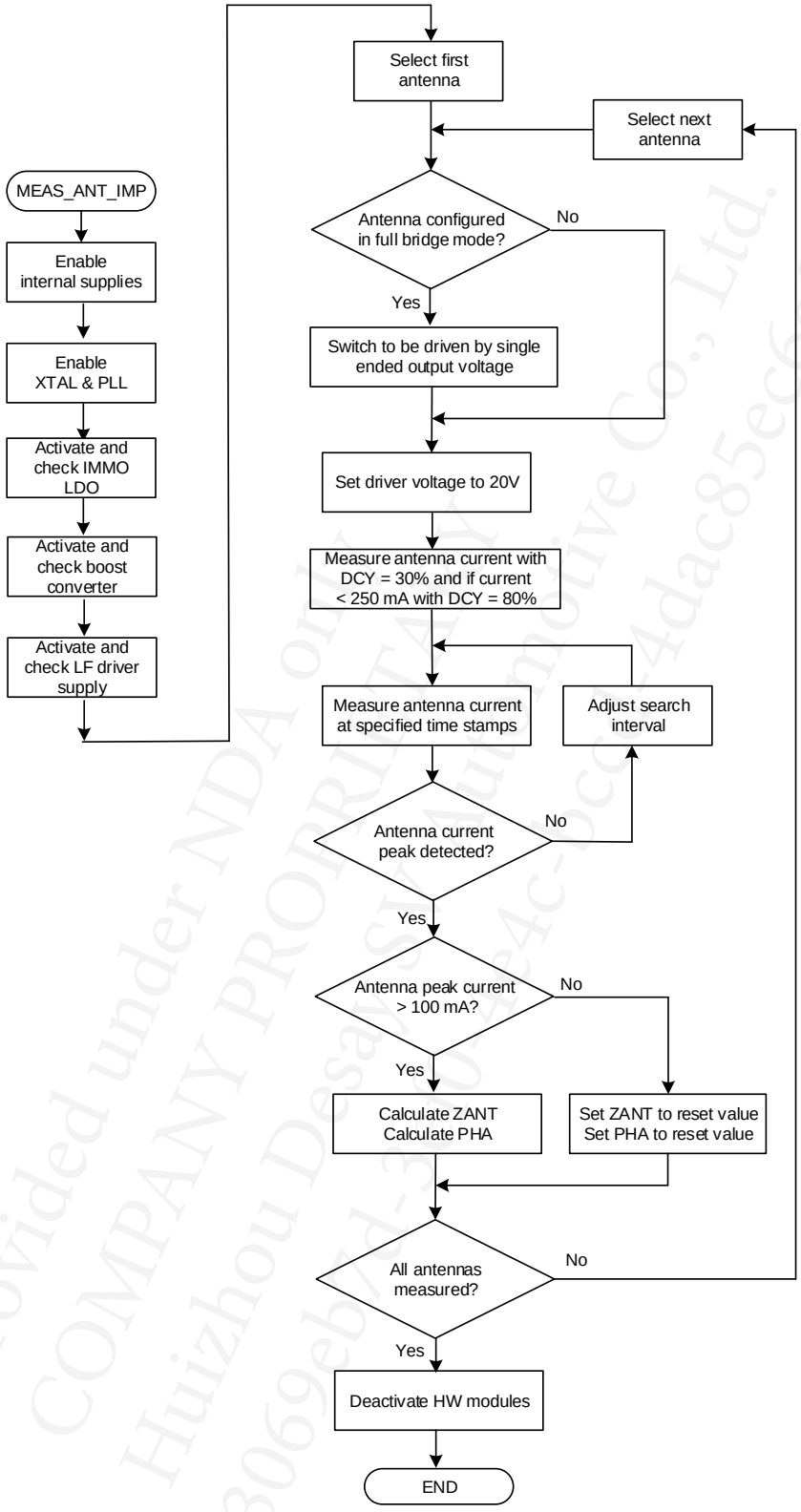

Fig 12. Measurement of antenna impedance and phase shift

10.1.1 Command

Table 132. MEAS_ANT_IMP command

LEN	CMD	PARAM		CRC8
0x04	0x48	DRP	RFU	CRC8

DRP:

DRP selects the LF driver for the antenna impedance measurement. Selecting more than one driver is possible.

10.1.2 Response

Table 133. MEAS_ANT_IMP response

LEN	CMD	STAT	CRC8
0x03	0x48	STAT	CRC8

10.1.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 134. MEAS_ANT_IMP status flags

Status byte	Status bit	Event or failure condition
STAT	SF_LAST_OP	Antenna impedance measurement not performed as V_{BAT} is larger than 18 V

10.2 MEAS_ANT_IMP_ADVANCED

MEAS_ANT_IMP_ADVANCED measures for each selected channel the antenna parameters impedance ZANT and phase shift PHA. The boost converter and the LF driver are activated for this.

The impedance measurement is performed with driver voltages V_{DR} of 20 V and 25 V and with driver duty cycles DCY_{DR} of 30 %, 50 %, 55 % and 75 %. The device operation frequency is varied arround the nominal operation frequency f_C with a resolution of f_C / 256 = 488.28125 Hz. The implemented algorithm starts at a frequency of 112.5 kHz and determines imdedances of antennas with a detuning of f_C - 25 % up to f_C + 8%.

After completion of MEAS_ANT_IMP_ADVANCED, the operation status flag SF_IMPMEAS is set (this flag is shared with MEAS_ANT_IMP). The values can be read via GET_ANT_IMP or GET_ANT_IMP_EFFECTIVE.

The LF driver settings (e.g. full bridge mode) shall be in line with the application before starting an antenna impedance measurement.

If SET_ANT_IMP or MEAS_ANT_IMP is interrogated afterwards, the values formerly measured via MEAS_ANT_IMP_ADVANCED are overwritten.

It is important to notice that the following two instructions shall be obeyed before interrogating MEAS_ANT_IMP_ADVANCED.

1) After each device cold start, CONFIG_ADVANCED with the following parameters has to be sent (CONFIG_CHIRP functionality):
    - CONFIG_ADVANCED (0x12, 0x02, 0x02, 0x01, 0x05, 0x08)

2) Each time before interrogating MEAS_ANT_IMP_ADVANCED, the device shall enter and leave SLEEP state, for example via the following SPI sequence:
   - Execute START_SLEEP (or alternatively START_SLEEP_FORCED)
   - Wake-up device (e.g. by SCSN transition)
   - Execute MEAS_ANT_IMP_ADVANCED

   Note: In single antenna operation, MEAS_ANT_IMP_ADVANCED can be executed without entering SLEEP state before.

10.2.1 Command

Table 135. MEAS_ANT_IMP_ADVANCED command

LEN	CMD	PARAM		CRC8
0x04	0xD5	DRP	DRN	CRC8

DRP, DRN:

DRP, DRN selects the LF driver for the antenna impedance measurement. Selecting more than one driver is possible.

10.2.2 Response

Table 136. MEAS_ANT_IMP_ADVANCED response

LEN	CMD	STAT	CRC8
0x03	0xD5	STAT	CRC8

10.2.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 137. MEAS_ANT_IMP_ADVANCED status flags

Status byte	Status bit	Event or failure condition
STAT	SF_LAST_OP	Antenna impedance measurement not performed as V_{BAT} is larger than 18 V
		Open antenna connections

10.3 SET_ANT_IMP

SET_ANT_IMP sets the antenna impedance values needed for the initial settings of the boost converter and of the class D* duty cycle. The antenna parameter set consists of inductivity L, quality factor Q and detuning DET.

If MEAS_ANT_IMP or MEAS_ANT_IMP_ADVANCED is interrogated afterwards, the settings formerly made via SET_ANT_IMP are overwritten.

10.3.1 Command

Table 138. SET_ANT_IMP command

LEN	CMD	PARAM				CRC8
LEN	0x49	{DRIDi	Li	Qi	DETi}	CRC8

DRIDi:

The driver (channel) ID identifies the LF channel for which the parameter set is assigned.

Li:

Li specifies the antenna inductivity.

Table 139. Li (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 0	L[7:0]	W		Inductance value in µH (steps a 10 µH)
			0x00	1 * 10 = 10 µH
			0x01	2 * 10 = 20 µH
			…	…
			0x8D	142 * 10 = 1420 µH
			0x8E	143 * 10 = 1430 µH
			0x8F	Reserved for future use
				…
			0xFF	Reserved for future use

Qi:

Qi specifies the antenna quality factor.

Table 140. Qi (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 5	RFU	W0		Reserved for future use
4 to 0	Q[4:0]	W		Quality factor
			0x00	1
			0x01	2
			…	…
			0x1E	31
			0x1F	32

DETi:

DETi specifies the antenna detuning.

Table 141. DETi (reset value 0x80)

Bit	Symbol	Access	Value	Description
7 to 0	DET[7:0]	W		Detuning of antenna ($\Delta f = f_C / 256 = 488.28125$ Hz)
			0x00	Reserved for future use
			…	…

Bit	Symbol	Access	Value	Description
			0x65	Reserved for future use
			0x66	f_C - 26 * Δf
			0x67	f_C - 25 * Δf
			…	…
			0x7F	f_C - 1 * Δf
			0x80	f_C + 0 * Δf
			0x81	f_C + 1 * Δf
			…	…
			0x99	f_C + 25 * Δf
			0x9A	f_C + 26 * Δf
			0x9B	Reserved for future use
				…
			0xFF	Reserved for future use

10.3.2 Response

The response is delayed by the calculation time to compute the antenna impedance Z_{ANT} and the antenna phase shift.

Table 142. SET_ANT_IMP response

LEN	CMD	STAT				CRC8
LEN	0x49	STAT	{DRIDi	ZANTi	PHAi}	CRC8

DRIDi:

The driver (channel) ID identifies the LF channel for which the parameter set is assigned.

ZANTi:

ZANTi specifies the antenna impedance. Values ≥ 64 Ω are handled internally with correct impedance, while return value is set to 0x7F.

Table 143. ZANTi reset value 0x00)

Bit	Symbol	Access	Value	Description
7	RFU	R0		Reserved for future use
6 to 0	ZANT[6:0]	R		Antenna impedance
			0x00	1 * 0.5 Ω = 0.5 Ω
			0x01	2* 0.5 Ω = 1.0 Ω
			0x02	3 * 0.5 Ω = 1.5 Ω
			…	
			0x7E	127* 0.5 Ω = 63.5 Ω
			0x7F	≥ 64.0 Ω

PHAi:

PHAi specifies the antenna phase shift.

Table 144. PHAi (reset value 0x80)

Bit	Symbol	Access	Value	Description
7 to 0	PHA[7:0]	R		Antenna phase shift
			0x00	-128°
			0x01	-127°
			…	
			0x7F	-1°
			0x80	0°
			0x81	+1°
			…	
			0xFE	+126°
			0xFF	+127°

10.3.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 145. SET_ANT_IMP status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Number of cascaded frames larger than number of drivers
		Duplicate DRID in set of cascaded frames

10.4 GET_ANT_IMP

GET_ANT_IMP reads the antenna impedance values.

10.4.1 Command

Table 146. GET_ANT_IMP command

LEN	CMD	PARAM		CRC8
0x04	0x4A	DRP	RFU	CRC8

DRP:

DRP selects the LF driver. Selecting more than one driver is possible.

10.4.2 Response

The command returns the requested actual antenna impedance values. If a requested impedance value has not been changed by settings or measurements, the default value is returned.

Table 147. GET_ANT_IMP response

LEN	CMD	STAT				CRC8
LEN	0x4A	STAT	{DRIDi	ZANTi	PHAi}	CRC8

DRIDi:

The driver (channel) ID identifies the LF channel for which the parameter set is retrieved.

ZANTi:

ZANTi specifies the antenna impedance. Values ≥ 64 Ω are handled internally with correct impedance, while return value is set to 0x7F.

If an open antenna i was detected, the values ZANTi = 0x00 and PHAi = 0x80 are returned.

Table 148. ZANTi (reset value 0x00)

Bit	Symbol	Access	Value	Description
7	RFU	R0		Reserved for future use
6 to 0	ZANT[6:0]	R		Antenna impedance
			0x00	1 * 0.5 Ω = 0.5 Ω
			0x01	2* 0.5 Ω = 1.0 Ω
			0x02	3 * 0.5 Ω = 1.5 Ω
			…	
			0x7E	127* 0.5 Ω = 63.5 Ω
			0x7F	≥ 64 Ω

PHAi:

PHAi specifies the antenna phase shift.

Table 149. PHAi (reset value 0x80)

Bit	Symbol	Access	Value	Description
7 to 0	PHA[7:0]	R		Antenna phase shift
			0x00	-128°
			0x01	-127°
			…	
			0x7F	-1°
			0x80	0°
			0x81	+1°
			…	
			0xFF	+127°

10.4.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

10.5 GET_ANT_IMP_EFFECTIVE

GET_ANT_IMP_EFFECTIVE reads the effective antenna impedance value (including the antenna impedance Z_{ANT} and the driver output resistance R_{DR}) and the effective antenna phase shift value.

10.5.1 Command

Table 150. GET_ANT_IMP_EFFECTIVE command

LEN	CMD	PARAM		CRC8
0x04	0xD4	DRP	DRN	CRC8

DRP, DRN:

DRP, DRN selects the LF driver. Selecting more than one driver is possible.

10.5.2 Response

The command returns the requested actual effective antenna impedance values. If a requested impedance value has not been changed by settings or measurements, the default value is returned.

Table 151. GET_ANT_IMP_EFFECTIVE response

LEN	CMD	PARAM					CRC8
LEN	0xD4	STAT	{DRIDi	ZANTEFFi	PHAEFFi_L	PHAEFFi_H}	CRC8

DRIDi:

The driver (channel) ID identifies the LF channel for which the parameter set is retrieved.

ZANTEFFi:

ZANTEFFi specifies the effective antenna impedance magnitude. Values ≥128 Ω are handled internally with correct impedance, while the return value is set to 0xFF.

If an open antenna i was detected (by interrogating START_DIAG before), the values ZANTEFFi = 0x0A and PHAEFFi = 0x400 are returned.

Table 152. ZANTEFFi (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7 to 0	ZANTEFF[7:0]	R		Effective antenna impedance
			0x00	1 * 0.5 Ω = 0.5 Ω
			0x01	2 * 0.5 Ω = 1.0 Ω
			0x02	3 * 0.5 Ω = 1.5 Ω
			…	
			0xFE	255 * 0.5 Ω = 127.5 Ω
			0xFF	≥128 Ω

PHAEFFi:

PHAEFFi specifies the effective antenna phase shift.

Table 153. PHAEFFi (reset value 0x0400)

Bit	Symbol	Access	Value	Description
15 to 11	RFU	W0		Reserved for future use
10 to 0	PHAEFF[10:0]	W		Effective antenna phase shift (related to center of rectangular driver voltage) $\Delta\varphi = 90°/1024 = 0.08789°$
			0x000	Reserved for future use
			0x001	Reserved for future use

Bit	Symbol	Access	Value	Description
			…	
			0x004	Reserved for future use
			0x005	- 1019 * Δφ
			0x006	- 1018 * Δφ
			…	
			0x3FE	- 2*Δφ
			0x3FF	- 1*Δφ
			0x400	0*Δφ (0°, no detuning)
			0x401	1*Δφ
			0x402	2*Δφ
			…	
			0x7FA	1018 *Δφ
			0x7FB	1019 *Δφ
			0x7FC	Reserved for future use
			…	
			0x7FF	Reserved for future use

10.5.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

11. Device protection

11.1 GET_PROT_STATUS

GET_PROT_STATUS reads the device protection flags.

11.1.1 Command

Table 154. GET_PROT_STATUS command

LEN	CMD	CRC8
0x02	0x58	CRC8

The response contains the device protection flags. In case no malfunction has been detected in the respective block, zero is returned in the corresponding parameter.

11.1.2 Response

Table 155. GET_PROT_STATUS response

LEN	CMD	STAT	PARAM			CRC8
0x06	0x58	STAT	PROTF	DRPF	RFU	CRC8

PROTF:

The protection flag register PROTF signalizes malfunctions pending that were generated by the corresponding block. Any write access is ignored.

Table 156. PROTF (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 5	RFU	R0		Reserved for future use
4	SF_DRSUP	R		Driver supply status
			0	No malfunction
			1	Malfunction
3	SF_BC	R		Boost converter status
			0	No malfunction
			1	Malfunction
2	SF_TEMPOV	R		Over temperature status
			0	No malfunction
			1	Malfunction
1	SF_BATUN	R		Battery undervoltage status
			0	No malfunction
			1	Malfunction
0	SF_BATOV	R		Battery overvoltage status
			0	No malfunction
			1	Malfunction

DRPF:

DRPF represents the LF driver protection status flag register.

Table 157. DRPF (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 and 6	RFU	R0		Reserved for future use
5	SF_DR6P	R		LF driver 6
			0	No malfunction
			1	Malfunction
4	SF_DR5P	R		LF driver 5
			0	No malfunction
			1	Malfunction
3	SF_DR4P	R		LF driver 4
			0	No malfunction
			1	Malfunction
2	SF_DR3P	R		LF driver 3
			0	No malfunction
			1	Malfunction
1	SF_DR2P	R		LF driver 2
			0	No malfunction
			1	Malfunction
0	SF_DR1P	R		LF driver 1
			0	No malfunction
			1	Malfunction

11.1.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

11.2 CLEAR_PROT_STATUS

CLEAR_PROT_STATUS clears the status flags which are marked.

11.2.1 Command

Table 158. CLEAR_PROT_STATUS command

LEN	CMD	PARAM			CRC8
0x05	0x59	PROTC	DRPC	RFU	CRC8

PROTC:

PROTC clears the status flags.

Table 159. PROTC (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7 to 5	RFU	W0		Reserved for future use
4	SC_DRSUP	W		Driver supply status
			0	No change
			1	Clear flag

Bit	Symbol	Access	Value	Description
3	SC_BC	W		Boost converter status
			0	No change
			1	Clear flag
2	SC_TEMPOV	W		Over temperature status
			0	No change
			1	Clear flag
1	SC_BATUN	W		Battery undervoltage status
			0	No change
			1	Clear flag
0	SC_BATOV	W		Battery overvoltage status
			0	No change
			1	Clear flag

DRPC:

DRPC specifies the drivers which flags to be cleared.

Table 160. DRPC (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7 and 6	RFU	W0		Reserved for future use
5	SC_DR6P	W		LF driver 6
			0	No change
			1	Clear flag
4	SC_DR5P	W		LF driver 5
			0	No change
			1	Clear flag
3	SC_DR4P	W		LF driver 4
			0	No change
			1	Clear flag
2	SC_DR3P	W		LF driver 3
			0	No change
			1	Clear flag
1	SC_DR2P	W		LF driver 2
			0	No change
			1	Clear flag
0	SC_DR1P	W		LF driver 1
			0	No change
			1	Clear flag

11.2.2 Response

Table 161.CLEAR_PROT_STATUS response

LEN	CMD	STAT	CRC8
0x03	0x59	STAT	CRC8

11.2.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 162. CLEAR_PROT_STATUS status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	All PROTC and DRPC set to 0

11.3 SET_PROT_MASK

SET_PROT_MASK enables the triggering of the INT pin if the device protection flag is set. In case any of the flags is set, the INT pin is set to ‘high’, if configured.

11.3.1 Command

Table 163. SET_PROT_MASK command

LEN	CMD	PARAM			CRC8
0x05	0x5A	PROTM	DRPM	RFU	CRC8

PROTM:

PROTM provides the mask to enable the INT pin if a device protection flag is set. The protection flags itself are not affected.

Table 164. PROTM (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 5	RFU	W0		Reserved for future use
4	SM_DRSUP	W		Set INT pin if driver supply status is set
			0	Disable
			1	Enable
3	SM_BC	W		Set INT pin if boost converter status is set
			0	Disable
			1	Enable
2	SM_TEMPOV	W		Set INT pin if over temperature status is set
			0	Disable
			1	Enable
1	SM_BATUN	W		Set INT pin if if battery undervoltage status is set
			0	Disable
			1	Enable
0	SM_BATOV	W		Set INT pin if battery overvoltage status is set
			0	Disable
			1	Enable

DRPM:

SM_DRiP masks the status flags of the LF drivers.

Table 165. DRPM (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 and 6	RFU	W0		Reserved for future use
5	SM_DR6P	W		Set INT pin if LF driver 6 status flag is set
			0	Disable
			1	Enable
4	SM_DR5P	W		Set INT pin if LF driver 5 status is set
			0	Disable
			1	Enable
3	SM_DR4P	W		Set INT pin if LF driver 4 status is set
			0	Disable
			1	Enable
2	SM_DR3P	W		Set INT pin if LF driver 3 status is set
			0	Disable
			1	Enable
1	SM_DR2P	W		Set INT pin if LF driver 2 status is set
			0	Disable
			1	Enable
0	SM_DR1P	W		Set INT pin if LF driver1 status is set
			0	Disable
			1	Enable

11.3.2 Response

Table 166.SET_PROT_MASK response

LEN	CMD	STAT	CRC8
0x03	0x5A	STAT	CRC8

11.3.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

12. Device diagnostics

12.1 START_DIAG

START_DIAG starts the diagnostics of the power stage, checking boost converter, LF driver and antennas.

The diagnostics sequence is shown in Fig 13. After completion, the operation status flag SF_DIAG is set.

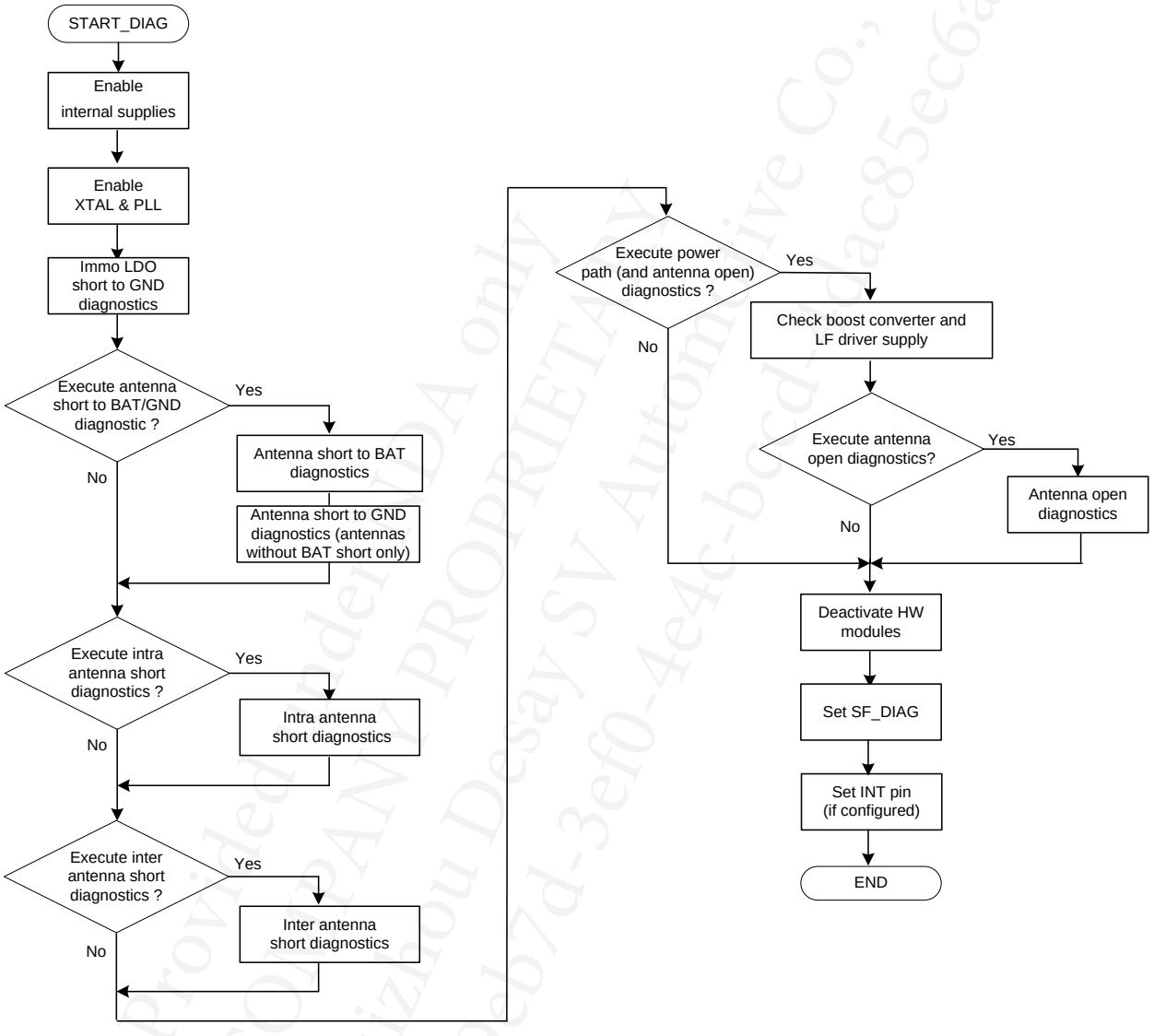

Fig 13. Diagnostics sequence

If a malfunction is detected, the diagnosis sequence is continued as far as possible in order to determine multiple malfunctions within one diagnosis sequence.

If the diagnosis status flags are not cleared before interrogating a START_DIAG command, the fail indications (‘1’ in status flags) from a previous diagnosis command are kept, while new detected fails are added.

If a diagnosis status flag is set for an LF antenna driver, no further diagnosis operations are performed at this driver until the diagnosis status flag is reset.

If an error occurs within one diagnosis step, all consecuting steps involving functionality of the former step will be omitted and the concerned flags keep their setting.

Example 1

If the check to “enable XTAL and PLL and internal supplies, check immo LDO output for DC short to GND” is returning a ‘1’ (malfunction), all consecutive checks will be omitted, hence those flags will not be modified.

Example 2

If a short of antenna 1 to GND was detected, all other following checks for antenna 1 will not be executed (potential risk of damange to the driver 1), hence the corresponding results of the checks for driver 1 will be not changed.

Example 3

All DC short, inter antenna and intra antenna checks and the check of the boost converter supply are PASS and reported as ‘0’ (no malfunction), the check of the LF driver supply is fail and reported back as ‘1’ (malfunction), none of the AC driver ‘open’ checks will be executed (potential risk of damage to the drivers), hence the corresponding result of the ‘open’ driver checks will be not changed.

12.1.1 Command

Table 167. START_DIAG command

LEN	CMD	PARAM			CRC8
0x05	0x4C	DRP	RFU	DIAGPAR	CRC8

DRP:

DRP specify the drivers for diagnostics.

DIAGPAR:

Table 168. DIAGPAR (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 6	SHTTIME	W		Time used to detect antenna shorts
			00	400 µs / 800 µs
			01	100 µs / 200 µs
			10	200 µs / 400 µs
			11	Reserved for future use
5	RFU	W0		Reserved for future use
4	OPN	W		Selects the AC antenna open diagnostics
			0	Disable
			1	Enable
3	POWER	W		Selects the power path diagnostics (boost converter and LF driver supply diagnostics)
			0	Disable
			1	Enable
2	SHTINTER	W		Selects the LF driver DC inter antenna short diagnostics
			0	Disable
			1	Enable
1	SHTINTRA	W		Selects the LF driver DC intra antenna short diagnostics
			0	Disable
			1	Enable
0	SHTBATGND	W		Selects the LF driver DC short to VBAT/GND diagnostics
			0	Disable
			1	Enable

SHTTIME:

SHTTIME determines the time the dedicated DC current sources are activated to detect shorts to GND or BAT and shorts between the antennas.

The overall diagnostics time depends on the SHTTIME setting. The first time is used for the short to GND test and for the short to BAT test each, while the second time is used for the intra/inter short test for each specified driver.

OPN:

Setting OPN selects the AC antenna open diagnostics. Please note that POWER has to be set at the same time.

POWER:

Starting the power path diagnosis requires that at least one valid driver is selected. If no driver is selected, SF_PAR is set in the command response.

SHTINTRA:

SHTINTRA enables detecting shorts between the LF driver output TxiP and TxiN of the same antenna.

SHTINTER:

SHTINTER enables detecting shorts between the LF driver outputs TxiP/N and TxiiP/N of different antennas.

12.1.2 Response

Table 169.START_DIAG response

LEN	CMD	STAT	CRC8
0x03	0x4C	STAT	CRC8

12.1.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 170. START_DIAG status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Command is used without selecting at least one valid driver
		OPN is set without setting POWER at the same time

12.2 GET_DIAG_STATUS

GET_DIAG_STATUS reads the diagnostics flags.

12.2.1 Command

Table 171. GET_DIAG_STATUS command

LEN	CMD	PARAM		CRC8
0x04	0x4D	DRP	RFU	CRC8

12.2.2 Response

Table 172. GET_DIAG_STATUS response

LEN	CMD	STAT	PARAM			CRC8
LEN	0x4D	STAT	SUPF	{DRIDi	DIAGFi}	CRC8

SUPF:

SUPF contains the diagnosis results for the boost converter and for the LF driver supply.

Table 173. SUPF (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 2	RFU	R0		Reserved for future use
1	SF_DIAGDRSUP	R		Diagnostic result of device supplies
			0	No malfunction
			1	Malfunction
0	SF_DIAGBC	R		Diagnostic result of boost converter
			0	No malfunction
			1	Malfunction

SF_DIAGDRSUP

SF_DIAGDRSUP is set if one of the following diagnosis steps fails

- Enable internal supplies
- Enable XTAL & PLL
- IMMO LDO short to GND diagnostics
- Check LF driver supply

There are three cases to be disguished:

- If a failure is detected either when enabling the internal supplies, XTAL & PLL or during the immo LDO short to GND diagnostics, the subsequent antenna and power path diagnostic (if selected) is omitted.
- If a failure is detected during the LF driver supply check, the results of the antenna short to BAT/GND diagnostics (if selected), intra antenna short diagnostics (if selected) and inter antenna short diagnostics (if selected) are still valid, while the power path diagnostic sequence (if selected) and the antenna open diagnostics (if selected) are omitted.
- If SF_DIAGDRSUP was set in a former diagnosis run and the application did not reset the flag, the complete antenna and power path diagnostic sequence (if selected) is omitted.

SF_DIAGBC

SF_DIAGBC is set if the check of the boost converter fails. In this case, the subsequent antenna open diagnostics (if selected) is omitted.

DRIDi, DIAGFi:

DIAGFi contain the diagnostic results for the respective LF driver DRIDi.

Table 174. DIAGFi (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 4	RFU	R0		Reserved for future use
3	SF_SHTANT	R		Tx intra or inter antenna-short
			0	No malfunction
			1	Malfunction
2	SF_OPN	R		Tx open antenna connection
			0	No malfunction
			1	Malfunction
1	SF_SHTBAT	R		Tx short to battery
			0	No malfunction
			1	Malfunction
0	SF_SHTGND	R		Tx short to GND
			0	No malfunction
			1	Malfunction

If any of the flags SF_SHTANTi, SF_OPNi, SF_SHTBATi or SF_SHTGNDi is set for driver i, either by a former or the actual diagnostics run, no further check for driver i will be performed until all error flags for this driver are cleared.

12.2.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

12.3 CLEAR_DIAG_STATUS

CLEAR_DIAG_STATUS clears the selected diagnosics status flags.

12.3.1 Command

Table 175. CLEAR_DIAG_STATUS command

LEN	CMD	PARAM			CRC8
LEN	0x4E	SUPC	{DRIDi	DIAGCi}	CRC8

SUPC:

SUPC clears the status flags of the boost converter and LF driver supply, dependent on the settings.

Table 176. SUPC (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7 to 2	RFU	W0		Reserved for future use
1	SC_DIAGDRSUP	W		ClassD* supply malfunction flag
			0	No change
			1	Clear flag
0	SC_DIAGBC	W		Boost converter malfunction flag
			0	No change
			1	Clear flag

DIAGCi:

DIAGCi clears the diagnostic flags for the respective LF driver DRIDi, dependent on the settings.

Table 177. DIAGCi (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 4	RFU	R0		Reserved for future use
3	SC_SHTANT	R		Tx intra or inter antenna-short
			0	No change
			1	Clear flag
2	SC_OPN	R		Tx open antenna connection flag
			0	No change
			1	Clear flag
1	SC_SHTBAT	R		Tx short to battery flag
			0	No change
			1	Clear flag
0	SC_SHTGND	R		Tx short to GND flag
			0	No change
			1	Clear flag

12.3.2 Response

Table 178.CLEAR_DIAG_STATUS response

LEN	CMD	STAT	CRC8
0x03	0x4E	STAT	CRC8

12.3.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 179. CLEAR_DIAG_STATUS status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	SUPC set to 0, if DRIDi and DIAGCi are omitted
		All SUPC and DIAGCi are set to 0
		Duplicate DRID in set of cascaded frames

13. SPI interface

13.1 CONFIG_SPI

CONFIG_SPI configures the SPI interface.

13.1.1 Command

Table 180. CONFIG_SPI command

LEN	CMD	PARAM		CRC8
0x04	0xF1	SPI_TIMEOUT	RFU	CRC8

SPI_TIMEOUT:

Holds the time used to detect an SPI receive error. The timeout counter gets active after receiving the length LEN and stops after receiving LEN bytes. When elapsed, the receive interface is reset without informing the host.

Table 181. SPI_TIMEOUT (reset value 0xFF)

Bit	Symbol	Access	Value	Description
7 to 0	SPI_TIMEOUT	W		Time to detect a receive error
			0x00	no timeout
			0x01	1 * 1 ms = 1 ms
			…	…
			0xFF	255 * 1 ms = 255 ms

13.1.2 Response

Table 182. CONFIG_SPI response

LEN	CMD	STAT	CRC8
0x03	0xF1	STAT	CRC8

13.1.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

13.2 ECHO_SPI

ECHO_SPI checks the proper operation of the SPI interface.

13.2.1 Command

Table 183. ECHO_SPI command

LEN	CMD	PARAM	CRC8
LEN	0x01	{DATAi}	CRC8

DATAi:

DATAi contains the data to be echoed from the device. The number of data bytes shall be larger than 0. It is calculated from the overall frame length.

LEN has to consider that the response containing the echoed data provides the STAT byte additionally, meaning the number of data bytes DATAi shall not exceed 252 bytes.

13.2.2 Response

Table 184. ECHO_SPI response

LEN	CMD	STAT	PARAM	CRC8
LEN	0x01	STAT	{DATAi}	CRC8

DATAi:

DATAi contains the echoed data bytes.

13.2.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 185. ECHO_SPI status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Number of data bytes to be echoed is 0 or > 252

14. Wake-up ports

14.1 CONFIG_WUP

CONFIG_WUP configures the wake-up ports. The port pin wake-up function becomes active during command execution and before the command response will be sent.

The port pin wake-up function can be switched off by setting WUPEN to 0x00.

14.1.1 Command

Table 186. CONFIG_WUP command

LEN	CMD	PARAM							CRC8
0x09	0x10	WUPEN	WUPEDG	WUPDEB	WUPVAL	WUPPRIO1	WUPPRIO2	WUPPRIO3	CRC8

WUPEN:

WUPEN enables or disables the port pin wake-up function.

Table 187. WUPEN (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 and 6	RFU	W0		Reserved for future use
5	WUP6EN	W		Port 6 wake-up
			0	Disabled
			1	Enabled
4	WUP5EN	W		Port 5 wake-up
			0	Disabled
			1	Enabled
3	WUP4EN	W		Port 4 wake-up
			0	Disabled
			1	Enabled
2	WUP3EN	W		Port 3 wake-up
			0	Disabled
			1	Enabled
1	WUP2EN	W		Port 2 wake-up
			0	Disabled
			1	Enabled
0	WUP1EN	W		Port 1 wake-up
			0	Disabled
			1	Enabled

WUPEDG:

WUPEDG specifies the signal edge on which the device wakes up.

Table 188. WUPEDG (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 and 6	RFU	W0		Reserved for future use
5	WUP6EDG	W		Port 6 wake-up edge selection
			0	Falling edge
			1	Rising edge
4	WUP5EDG	W		Port 5 wake-up edge selection
			0	Falling edge
			1	Rising edge
3	WUP4EDG	W		Port 4 wake-up edge selection
			0	Falling edge
			1	Rising edge
2	WUP3EDG	W		Port 3 wake-up edge selection
			0	Falling edge
			1	Rising edge
1	WUP2EDG	W		Port 2 wake-up edge selection
			0	Falling edge
			1	Rising edge
0	WUP1EDG	W		Port 1 wake-up edge selection
			0	Falling edge
			1	Rising edge

WUPDEB, WUPVAL:

WUPDEB and WUPVAL specify the time for additional debouncing of the input signal. In the following description, dependent on the wake-up edge configuration (falling/rising) inverted polarities have to be considered.

First, a valid wake-up event on the WUP input has to be detected after a falling / rising edge and after passing the wake-up input filter time $t_{WUP,WAKE}$. If the device is in SLEEP or POLLING state, the embedded µController wakes up within the time $t_{MRK3,WAKE}$ (Fig 14) and the device transfers to IDLE state. If the device is in IDLE state already, this state is kept (Fig 15).

Next, the debouncing timer is started. After the configured debouncing time t_{WUPDEB}, the device checks for low/high level.

If this check is successful, a validation timer is started. If the signal is always below/above the low/high level threshold until the time $t_{WUP,VAL}$ is reached, the wake-up signal is classified as valid and the SF_WUPi flag is set. Additionally, the INT pin is set, if configured.

If the signal gets high/low before the time $t_{WUP,VAL}$ is reached, the wake-up signal is classified to be invalid and the SF_WUPi flag is not set. Immediately afterwards the detection of a new WUP event is activated.

If the WUP event was triggered in SLEEP or POLLING state, in case of a valid WUP event the device is in IDLE state afterwards, while in case of an invalid WUP event the device re-enters SLEEP or POLLING state again, respectively.

If both values WUPDEB and WUPVAL are set to 0, the additional debouncing and validation of the the input signal is deactivated.

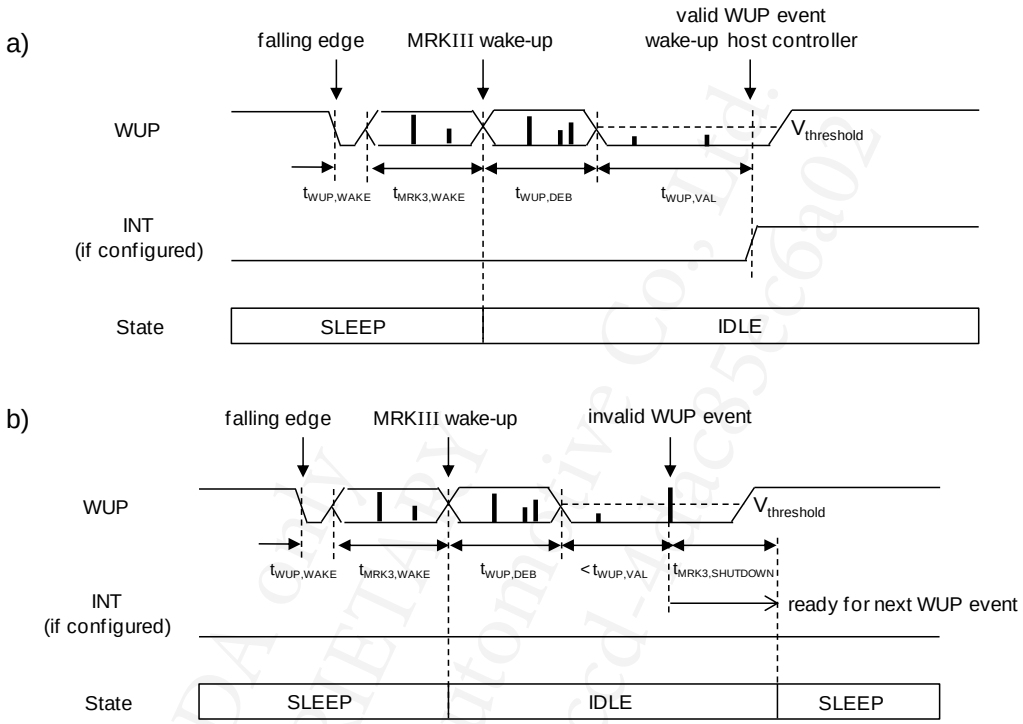

Fig 14. Wake-up signal debouncing starting from SLEEP state (wake-up on falling edge) for valid WUP (a) and for invalid WUP (b)

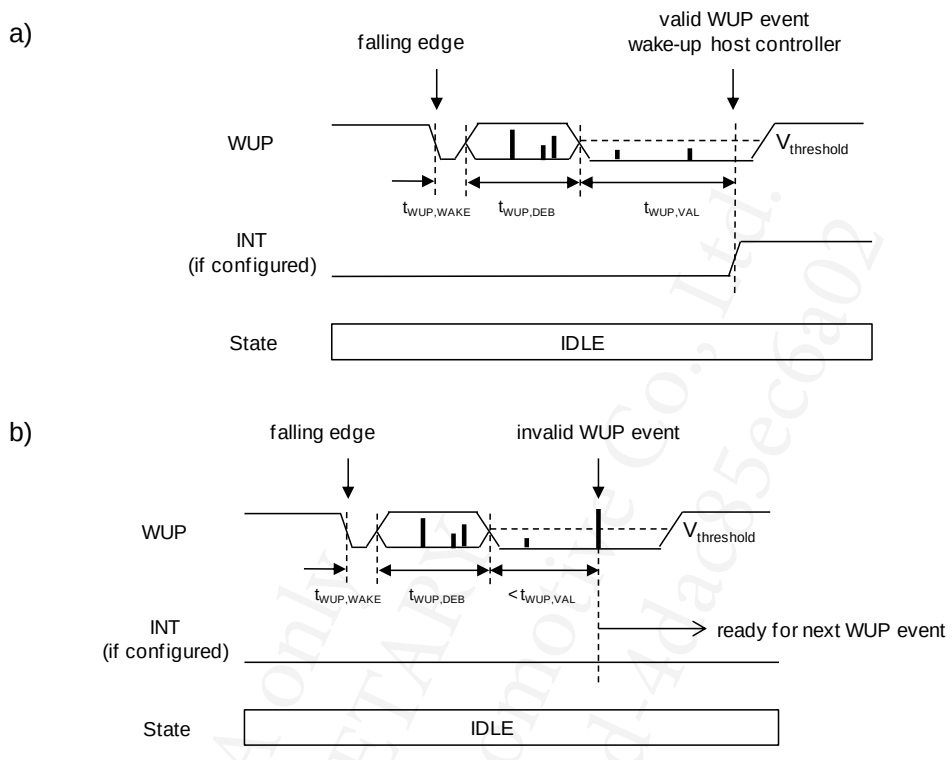

Fig 15. Wake-up signal debouncing in IDLE state (wake-up on falling edge) for valid WUP (a) and for invalid WUP (b)

If WUPDEB or WUPVAL is set to a value different to 0 and the device is not in SLEEP or POLLING state, external wake-up low pulses have to be at minimum 5 ms longer than the sum of the configured times for wake-up signal debouncing and validation due to possible delays caused by high priority firmware computing. Under these conditions WUP events will not be processed during execution of the commands MEAS_ANT_IMP or START_DIAG.

Table 189. WUPDEB (reset value 0x10)

Bit	Symbol	Access	Value	Description
7 and 6	RFU	W0		Reserved for future use
5 to 0	WUPDEB[5:0]	W		Time interval for WUP debouncing
			0x00	0 * 1 = 0 ms
			0x01	1 * 1 = 1 ms
			…	
			0x3F	63 * 1 = 63 ms

Table 190. WUPVAL (reset value 0x1E)

Bit	Symbol	Access	Value	Description
7 and 6	RFU	W0		Reserved for future use
5 to 0	WUPVAL[5:0]	W		Time interval for WUP valid check
			0x00	0 * 1 = 0 ms

Bit	Symbol	Access	Value	Description
			0x01	1 * 1 = 1 ms
			…	
			0x3F	63 * 1 = 63 ms

WUPPRIO:

WUPPRIO specifies the priorities for WUP handling.

If a WUP has been triggered and a second (or more) WUP is recognized before starting sending the LF signal of the first WUP, the WUP are served in the order of their priorities, starting with priority 7 in falling order downto priority 0.

If two (or more) WUP are configured to have the same priority, the WUP are served according to their number, starting with WUP1 having highest priority in rising order upto WUP6 having lowest priority.

Table 191. WUPPRIO1 (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 4	WUP6PRIO[3:0]	W		WUP6 priority
			0000	Priority 0 (lowest)
			0001	Priority 1
			…	
			0110	Priority 6
			0111	Priority 7 (highest)
			1000	Reserved for future use
			…	
			1111	Reserved for future use
3 to 0	WUP5PRIO[3:0]	W		WUP5 priority
			0000	Priority 0 (lowest)
			0001	Priority 1
			…	
			0110	Priority 6
			0111	Priority 7 (highest)
			1000	Reserved for future use
			…	
			1111	Reserved for future usev

Table 192. WUPPRIO2 (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 4	WUP4PRIO[3:0]	W		WUP4 priority
			0000	Priority 0 (lowest)
			0001	Priority 1
			…	
			0110	Priority 6
			0111	Priority 7 (highest)

Bit	Symbol	Access	Value	Description
			1000	Reserved for future use
			…	
			1111	Reserved for future use
3 to 0	WUP3PRIO[3:0]	W		WUP3 priority
			0000	Priority 0 (lowest)
			0001	Priority 1
			…	
			0110	Priority 6
			0111	Priority 7 (highest)
			1000	Reserved for future use
			…	
			1111	Reserved for future use

Table 193. WUPPRIO3 (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 4	WUP2PRIO[3:0]	W		WUP2 priority
			0000	Priority 0 (lowest)
			0001	Priority 1
			…	
			0110	Priority 6
			0111	Priority 7 (highest)
			1000	Reserved for future use
			…	
			1111	Reserved for future use
3 to 0	WUP1PRIO[3:0]	W		WUP1 priority
			0000	Priority 0 (lowest)
			0001	Priority 1
			…	
			0110	Priority 6
			0111	Priority 7 (highest)
			1000	Reserved for future use
			…	
			1111	Reserved for future use

14.1.2 Response

Table 194. CONFIG_WUP response

LEN	CMD	STAT	CRC8
0x03	0x10	STAT	CRC8

14.1.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

14.2 GET_WUP_STATUS

GET_WUP_STATUS reads the wake-up flags. In case a flag is set, the INT pin is set to 'high', if configured. As long as a WUP status flag is set, no new WUP event on that pin is recognized.

14.2.1 Command

Table 195. GET_WUP_STATUS command

LEN	CMD	CRC8
0x02	0x13	CRC8

The response contains the wake-up flags. In case no wake-up has been recognized in the respective block, zero is returned in the corresponding parameter.

14.2.2 Response

Table 196. GET_WUP_STATUS response

LEN	CMD	STAT	PARAM	CRC8
0x04	0x13	STAT	WUPF	CRC8

WUPF:

The wake-up status flag register WUPF signalizes wake-up events pending that were generated by the corresponding wake-up inputs (WUP1 to WUP6). Any write access is ignored.

Table 197. WUPF (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 and 6	RFU	R0		Reserved for future use
5	SF_WUP6	R		Port 6 wake-up status
			0	No wake-up
			1	Wake-up
4	SF_WUP5	R		Port 5 wake-up status
			0	No wake-up
			1	Wake-up
3	SF_WUP4	R		Port 4 wake-up status
			0	No wake-up
			1	Wake-up
2	SF_WUP3	R		Port 3 wake-up status
			0	No wake-up
			1	Wake-up
1	SF_WUP2	R		Port 2 wake-up status
			0	No wake-up
			1	Wake-up
0	SF_WUP1	R		Port 1 wake-up status
			0	No wake-up
			1	Wake-up

14.2.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

14.3 CLEAR_WUP_STATUS

CLEAR_WUP_STATUS clears the wake-up status flags which are marked.

14.3.1 Command

Table 198. CLEAR_WUP_STATUS command

LEN	CMD	PARAM	CRC8
0x03	0x14	WUPC	CRC8

WUPC:

Specifies the wake-up flag to be cleared.

Table 199. WUPC (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7 to 6	RFU	W0		Reserved for future use
5	SC_WUP6	W		Port 6 wake-up flag
			0	No change
			1	Clear flag
4	SC_WUP5	W		Port 5 wake-up flag
			0	No change
			1	Clear flag
3	SC_WUP4	W		Port 4 wake-up flag
			0	No change
			1	Clear flag
2	SC_WUP3	W		Port 3 wake-up flag
			0	No change
			1	Clear flag
1	SC_WUP2	W		Port 2 wake-up flag
			0	No change
			1	Clear flag
0	SC_WUP1	W		Port 1 wake-up flag
			0	No change
			1	Clear flag

14.3.2 Response

Table 200.CLEAR_WUP_STATUS response

LEN	CMD	STAT	CRC8
0x03	0x14	STAT	CRC8

14.3.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 201. CLEAR_WUP_STATUS status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	WUPC set to 0

14.4 SET_WUP_MASK

SET_WUP_MASK enables the triggering of the INT pin if a wake-up event is set.

14.4.1 Command

Table 202. SET_WUP_MASK command

LEN	CMD	PARAM	CRC8
0x03	0x15	WUPM	CRC8

WUPM:

WUPM provides the mask to enable the INT pin if a wake-up event occurs. The wake-up status flags itself are not affected.

Table 203. WUPM (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 and 6	RFU	W0		Reserved for future use
5	SM_WUP6	W		Set INT pin if port 6 wake-up status is set
			0	Disable
			1	Enable
4	SM_WUP5	W		Set INT pin if port 5 wake-up status is set
			0	Disable
			1	Enable
3	SM_WUP4	W		Set INT pin if port 4 wake-up status is set
			0	Disable
			1	Enable
2	SM_WUP3	W		Set INT pin if port 3 wake-up status is set
			0	Disable
			1	Enable
1	SM_WUP2	W		Set INT pin if port 2 wake-up status is set
			0	Disable
			1	Enable
0	SM_WUP1	W		Set INT pin if port 1 wake-up status is set
			0	Disable
			1	Enable

14.4.2 Response

Table 204.SET_WUP_MASK response

LEN	CMD	STAT	CRC8
0x03	0x15	STAT	CRC8

14.4.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

15. WUP event triggered polling

15.1 CONFIG_WUP_POLLING

CONFIG_WUP_POLLING configures the device for handling a port wake-up event on a port pin (WUP1-WUP6) and to trigger LF transmission without interaction of the host controller.

Simultaneous operation of up to 3 LF drivers is supported. Waiting times can be configured between driver activation. Dependent on the number of parameters, the parameter LEN has to be adopted accordingly.

The command CONFIG_WUP_POLLING should be used after configuring the drivers and the data due to consistency checks.

The WUP polling starts after the command response has been sent. The WUP polling can be switched off via the CONFIG_WUP command by setting WUPEN to 0x00.

If STOP_LF_TRANSMIT is called during WUP event triggered polling, all pending and ongoing frames are stopped. The WUP debouncing and validation is continued and may trigger a new LF frame immediately afterwards.

If there are any WUP events pending and not cleared when the WUP polling is configured, these WUP events are cleared.

In case of a valid WUP event after the WUP polling was enabled, the configured LF transmission will be initiated. Afterwards, the WUP event has to be cleared before the next LF transmission can be initiated.

15.1.1 Command

The command contains blocks of parameter bytes, each block containing the identifier (DRPi) of the drivers to be activated after recognition of the WUP event. A pause time PTIMEi allows activating different drivers after each other with a delay in between.

It is important to notice that the maximal SPI frame length of CONFIG_WUP_POLLING is limited to 68 bytes (SPI message length LEN is 67).

Table 205. CONFIG_WUP_POLLING command

LEN	CMD	PARAM										CRC8
LEN	0x51	WUPIO	{DRPi	RFU	LCDRPi	RFU	PTIME_Li	PTIME_Hi	LENDATAIDi	{DATAIDik}}		CRC8

The LF telegram to be sent can have flexible length. The LF telegram data is split up in data segments which are configured before via the command SET_LF_DATA. Each segment can be selected via the corresponding identifier DATAIDik (Fig 16).

LENDATAIDi is used to determine the number of data segments that are used to compose one LF telegram i.

If PREAMB is set (command CONFIG_DEVICE), each configured sequence i starts with the standard NXP preamble and code violation pattern, followed by the data configured in the specified data segment(s) DATAIDik. If PREAMB is not set, for each configured sequence i the data configured in the specified data segment(s) DATAIDik is sent directly.

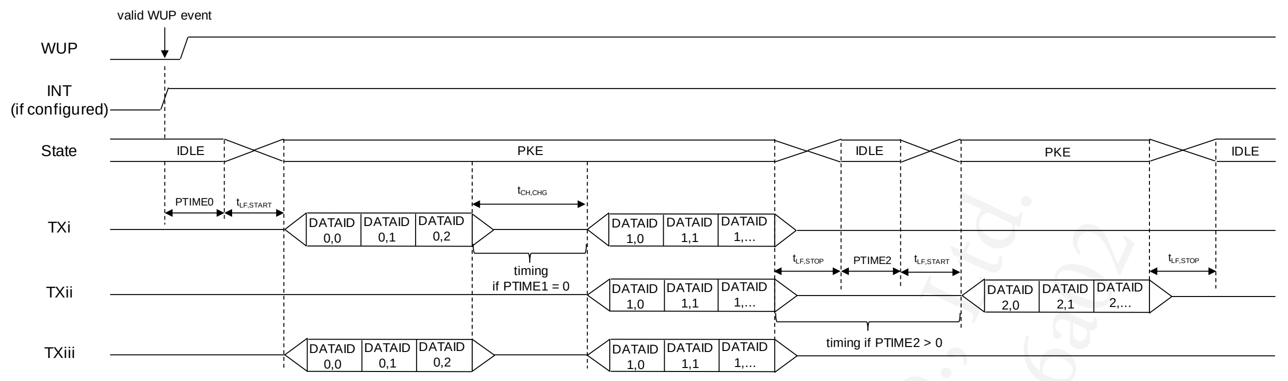

Fig 16. WUP event triggered polling (example with 3 LF telegrams)

WUPIO:

WUPIO selects the wake-up ports to trigger LF transmission.

Table 206. WUPIO (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 3	RFU	W0		Reserved for future use
2 to 0	WUPIO[2:0]	W		WUP to be configured
			0x00	WUP port 1
			0x01	WUP port 2
			0x02	WUP port 3
			0x03	WUP port 4
			0x04	WUP port 5
			0x05	WUP port 6
			0x06	Reserved for future use
			0x07	Reserved for future use

Note: If a channel is configured once for WUP polling, LF transmission is triggered afterwards according to the actual settings until a POR occurs.

DRPi:

DRPi selects the LF driver. Selecting more than one driver is possible.

LCDRPi:

LCDRPi selects the low current LF drivers to be activated in parallel to the main drivers. The low current values used for the selected channels are specified before via the CONFIG_LC_DRIVER command.

PTIMEi:

PTIMEi specifies a pause time between activating drivers. Please note that the device does not enter SLEEP state within the pause.

Table 207. PTIMEi (reset value 0x0000)

Bit	Symbol	Access	Value	Description
15 to 0	PTIME[15:0]	W		Pause time between activating drivers (steps a 1 ms)
			0x0000	0 * 1 = 0 ms
			0x0001	Reserved for future use
			0x0002	2 * 1 ms = 2 ms
			0x0003	3 * 1 ms = 3 ms
			…	
			0xFFFF	65535 * 1 ms = 65535 ms

PTIMEi is derived from a continuously running timer providing a 1 ms tick. As the timer is not synchronized to PTIMEi, setting a value PTIMEi results in an effective pause that is between (PTIMEi – 1) ms and PTIMEi ms.

Example

Setting PTIMEi to 0x0006 (6 ms), the resulting effective pause is between 5 ms and 6 ms.

It is important to notice that dependent on the value of PTIMEi (larger or equal 0) the following times have to be added to get the complete pause between the activated drivers

- PTIMEi = 0: $t_{CH, CHG}$
- PTIMEi > 0: $t_{LF,START}$, $t_{LF,STOP}$

LENDATAIDi, DATAIDik:

The LF telegram to be sent can have flexible length, thus the number of data segments building the telegram i has to be specified via LENDATAIDi. The LF telegram data itself is stored in data segments configured before via the command SET_LF_DATA and selected via DATAIDik.

15.1.2 Response

Table 208. CONFIG_WUP_POLLING response

LEN	CMD	STAT	CRC8
0x03	0x51	STAT	CRC8

15.1.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 209. CONFIG_WUP_POLLING status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Number of selected main drivers 0 or larger than 3
		Selected DATAID not configured
		LENDATAIDi set to 0
		Same driver(s) used in normal operation and in low current operation

16. Timer triggered polling

16.1 CONFIG_TIMER_POLLING

CONFIG_TIMER_POLLING configures the timer for sending LF telegram data autonomously and based on a predefined schedule without interaction of the host controller. The command allows configuring different drivers to become active after expiration of the respective polling time.

Simultaneous operation of up to 3 LF drivers is supported. Dependent on the polling scheme, the parameter LEN has to be adopted accordingly.

The command CONFIG_TIMER_POLLING shall be used only after configuring the drivers and the data due to consistency checks.

16.1.1 Command

The command contains blocks of parameter bytes. Each block contains the identifier (DRPi) of the drivers to be activated after expiration of the polling time and a parameter PTIMEi to specify the polling time.

It is important to notice that the maximal SPI frame length of CONFIG_TIMER_POLLING is limited to 131 byte (SPI message length LEN is 130).

Table 210. CONFIG_TIMER_POLLING command

LEN	CMD	PARAM								CRC8
LEN	0x52	{DRPi	RFU	LCDRPi	RFU	PTIME_Li	PTIME_Hi	LENDATAIDi	{DATAIDik}}	CRC8

The LF telegram to be sent can have flexible length. The LF telegram data is split up in data segments which are configured before via the command SET_LF_DATA. Each segment can be selected via the corresponding identifier DATAIDik (Fig 17).

LENDATAIDi is used to determine the number of data segments that are used to compose one telegram i.

If PREAMB is set (command CONFIG_DEVICE), each configured sequence i starts with the standard NXP preamble and code violation pattern, followed by the data configured in the specified data segment(s) DATAIDik. If PREAMB is not set, for each configured sequence i the data configured in the specified data segment(s) DATAIDik is sent directly.

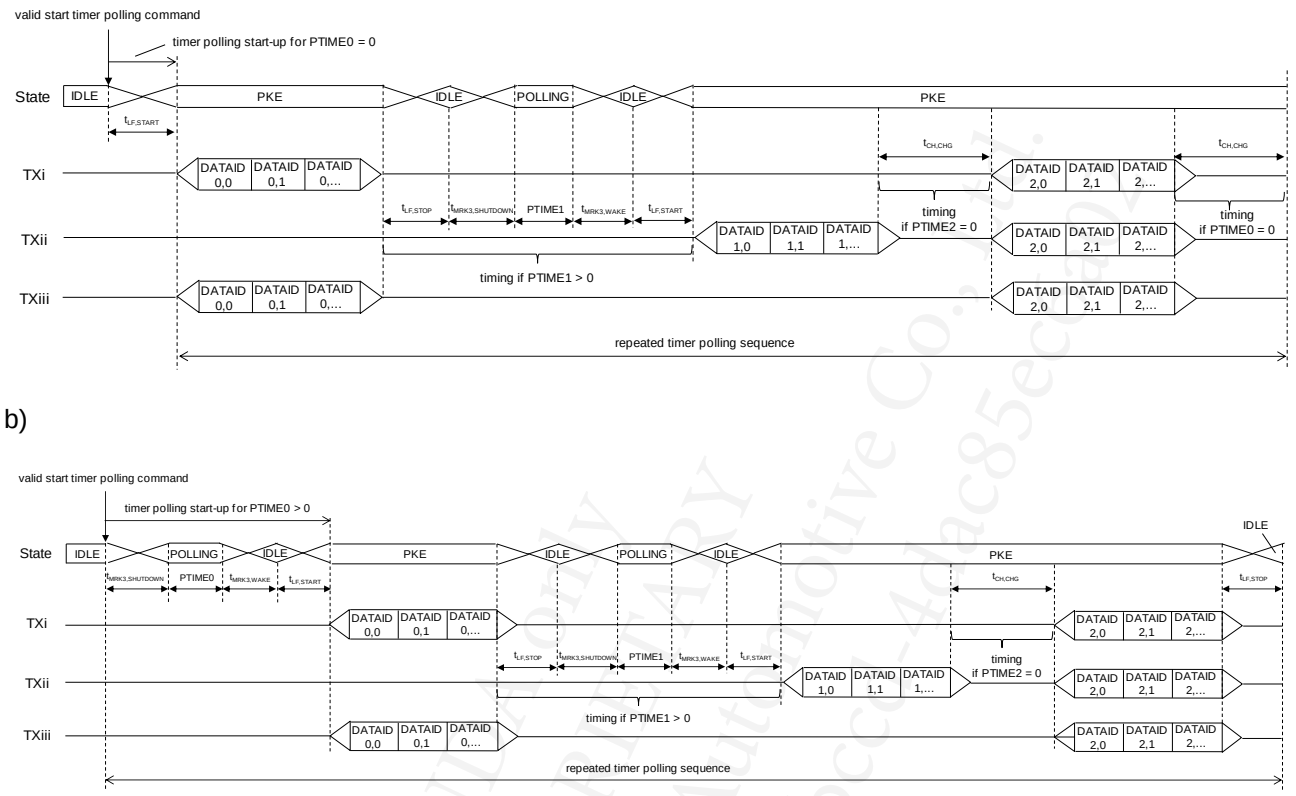

Fig 17. Timer triggered polling with PTIME0 = 0 (a) and PTIME0 > 0 (b) (example with 3 LF telegrams)

DRPi:

DRPi selects the LF driver. Selecting more than one driver is possible.

LCDRPi:

LCDRPi selects the low current LF drivers to be activated in parallel to the main drivers. The low current values used for the selected channels are specified before via the CONFIG_LC_DRIVER command.

PTIMEi:

PTIMEi represents the time interval between activating the configured drivers.

Please note that the timer activated in POLLING state is clocked by the low power RC oscillator with limited accuracy ($f_{OSC,RC,POL}$).

Table 211. PTIMEi (reset value 0x0000)

Bit	Symbol	Access	Value	Description
15 to 0	PTIME[15:0]	W		Time interval for polling
			0x0000	0 * 1 = 0 ms
			0x0001	Reserved for future use
			0x0002	2 * 1 ms = 2 ms

Bit	Symbol	Access	Value	Description
			0x0003	3 * 1 ms = 3 ms
			…	
			0xFFFF	65535 * 1 = 65535 ms

PTIMEi is derived from a continuously running timer providing a 1 ms tick. As the timer is not synchronized to PTIMEi, setting a value PTIMEi results in an effective pause that is between (PTIMEi – 1) ms and PTIMEi ms.

Example

Setting PTIMEi to 0x0006 (6 ms), the resulting effective pause is between 5 ms and 6 ms.

It is important to notice that dependent on the value of PTIMEi (larger or equal 0) the following times have to be added to get the complete pause between the activated drivers

- PTIMEi = 0: $t_{CH, CHG}$
- PTIMEi > 0: $t_{MRK3,WAKE}$, $t_{MRK3, SHUTDOWN}$, $t_{LF,START}$, $t_{LF,STOP}$

LENDATAIDi, DATAIDik:

The LF telegram to be sent can have flexible length, thus the number of data segments building the telegram i has to be specified via LENDATAik. The LF telegram data itself is stored in data segments configured before via the command SET_LF_DATA and selected via DATAIDik.

16.1.2 Response

Table 212. CONFIG_TIMER_POLLING response

LEN	CMD	STAT	CRC8
0x03	0x52	STAT	CRC8

16.1.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 213. CONFIG_TIMER_POLLING status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Number of selected main drivers 0 or larger than 3
		Selected DATAID not configured
		LENDATAIDi set to 0
		Same driver(s) used in normal operation and in low current operation

16.2 START_TIMER_POLLING

START_TIMER_POLLING sets the device in POLLING state and activates the polling timer according to the configured parameters. POLLING state is entered and the polling timer is started immediately after sending the CRC8 in the command response.

The POLLING state can be left via a wake-up event.

16.2.1 Command

Table 214. START_TIMER_POLLING command

LEN	CMD	CRC8
0x02	0x53	CRC8

16.2.2 Response

Table 215. START_TIMER_POLLING response

LEN	CMD	STAT	CRC8
0x03	0x53	STAT	CRC8

16.2.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 216. START_TIMER_POLLING status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Timer polling not configured

17. Temperature indication

17.1 CONFIG_TEMP

CONFIG_TEMP configures the threshold for temperature indication/warning.

17.1.1 Command

Table 217. CONFIG_TEMP command

LEN	CMD	PARAM	CRC8
0x03	0x18	TPAR	CRC8

TPAR:

TPAR specifies the temperature threshold for temperature indication/warning and enables the function on demand.

When the temperature threshold is reached, a value of 10°C (5°C for 85°C) below the configured threshold is used to realize a cool down hysteresis.

When cooled down, the configured temperature threshold is activated again to be sensitive for the next temperature indication/warning event.

Table 218. TPAR (reset value 0x0X)

Bit	Symbol	Access	Value	Description
7 to 5	RFU	W0		Reserved for future use
4	TWARN_EN	W		Temperature indication/warning enable
			0	Disable
			1	Enable
3 to 0	TTHR[3:0]	W		Threshold for temperature indication/warning
			0x0	85°C
			0x1	90°C
			…	
			0xE	155°C
			0xF	Reserved for future use

17.1.2 Response

Table 219. CONFIG_TEMP response

LEN	CMD	STAT	CRC8
0x03	0x18	STAT	CRC8

17.1.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

17.2 GET_TEMP_STATUS

GET_TEMP_STATUS reads the temperature warning flag.

17.2.1 Command

Table 220. GET_TEMP_STATUS command

LEN	CMD	CRC8
0x02	0x19	CRC8

17.2.2 Response

Table 221. GET_TEMP_STATUS response

LEN	CMD	STAT	PARAM	CRC8
0x04	0x19	STAT	TEMPF	CRC8

TEMPF:

TEMPF contains the temperature warning flag.

Table 222. TEMPF (reset value 0x00)

Bit	Symbol	Access	Value	Description
1 to 7	RFU	R0		Reserved for future use
0	SF_TWARN	R		Temperature warning flag
			0	Temperature threshold not reached
			1	Temperature threshold reached

17.2.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

17.3 CLEAR_TEMP_STATUS

CLEAR_TEMP_STATUS clears the temperature warning flag.

17.3.1 Command

Table 223. CLEAR_TEMP_STATUS command

LEN	CMD	PARAM	CRC8
0x03	0x1A	TEMPC	CRC8

TEMPC:

TEMPC clears the temperature warning flag dependent on the setting.

Table 224. TEMPC (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7 to 1	RFU	W0		Reserved for future use
0	SC_TWARN	W		Temperature warning flag
			0	No change
			1	Clear flag

17.3.2 Response

Table 225.CLEAR_TEMP_STATUS response

LEN	CMD	STAT	CRC8
0x03	0x1A	STAT	CRC8

17.3.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 226. CLEAR_TEMP_STATUS status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	TEMPC set to 0

17.4 SET_TEMP_MASK

SET_TEMP_MASK enables the triggering of the INT pin if the temperature warning flag is set.

17.4.1 Command

Table 227. SET_TEMP_MASK command

LEN	CMD	PARAM	CRC8
0x03	0x1B	TEMPM	CRC8

TEMPM:

TEMPM provides the mask to enable the INT pin if the temperature warning flag is set. The temperature warning flag itself is not affected.

Table 228. TEMPM (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 1	RFU	W0		Reserved for future use
0	SM_TWARN	W		Set INT pin if temperature warning flag is set
			0	Disable
			1	Enable

17.4.2 Response

Table 229.SET_TEMP_MASK response

LEN	CMD	STAT	CRC8
0x03	0x1B	STAT	CRC8

17.4.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

18. Operation status flags

18.1 GET_OP_STATUS

GET_OP_STATUS reads the operation status flags which are set by the device when a dedicated operation has been finished.

If a protection event occurs before the operation is finished, only the protection flag is set and the operation status flag is not set.

18.1.1 Command

Table 230. GET_OP_STATUS command

LEN	CMD	CRC8
0x02	0x55	CRC8

18.1.2 Response

Table 231.GET_OP_STATUS response

LEN	CMD	STAT	PARAM	CRC8
0x04	0x55	STAT	OPF	CRC8

OPF:

Table 232. OPF (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 4	RFU	R0		Reserved for future use
3	SF_RXREADY	R		Immo receiver ready
			0	Inactive or busy (no new data received)
			1	Ready (new data received)
2	SF_TXREADY	R		LF transmission ready
			0	Inactive or busy (LF transmission ongoing)
			1	Ready (LF transmission finished)
1	SF_IMPMEAS	R		Impedance measurement ready
			0	Inactive or busy
			1	Ready
0	SF_DIAG	R		Diagnostics ready
			0	Inactive or busy
			1	Ready

SF_RXREADY:

SF_RXREADY is set after finishing an immobilizer operation initiated before via the SPI command START_IMMO_TRANSCEIVE.

Please note that newly received data is not ready to be read from the receive buffer before SF_RXREADY is set. The reason is that after finishing the reception via the LF interface internal data processing time is needed.

SF_TXREADY:

SF_TXREADY is set after finishing an LF transmission initiated before via the SPI commands

- START_LF_TRANSMIT
- START_LF_TRANSMIT_DATA
- START_IMMO_TRANSMIT

SF_TXREADY is not set after executing a START_IMMO_TRANSCEIVE command nor after a WUP event triggered polling operation or after a timer triggered polling operation.

SF_IMPMEAS:

SF_IMPMEAS is set after finishing an antenna impedance measurement initiated before via the SPI command MEAS_ANT_IMP or MEAS_ANT_IMP_ADVANCED.

SF_DIAG:

SF_DIAG is set after finishing a device diagnostics initiated before via the SPI command START_DIAG.

18.1.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

18.2 CLEAR_OP_STATUS

CLEAR_OP_STATUS clears the status flags of the marked operations.

18.2.1 Command

Table 233. CLEAR_OP_STATUS command

LEN	CMD	PARAM	CRC8
0x03	0x56	OPC	CRC8

OPC:

Table 234. OPC (reset value 0xXX)

Bit	Symbol	Access	Value	Description
7 to 4	RFU	W0		Reserved for future use
3	SC_RXREADY	W		Immo receiver ready
			0	No change
			1	Clear flag
2	SC_TXREADY	W		LF transmission ready
			0	No change
			1	Clear flag

Bit	Symbol	Access	Value	Description
1	SC_IMPMEAS	W		Impedance measurement ready
			0	No change
			1	Clear flag
0	SC_DIAG	W		Diagnostics ready
			0	No change
			1	Clear flag

18.2.2 Response

Table 235.CLEAR_OP_STATUS response

LEN	CMD	STAT	CRC8
0x03	0x56	STAT	CRC8

18.2.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 236. CLEAR_OP_STATUS status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	OPC set to 0

18.3 SET_OP_MASK

SET_OP_MASK enables the triggering of the INT pin if an operation status flag is set to ready.

18.3.1 Command

Table 237. SET_OP_MASK command

LEN	CMD	PARAM	CRC8
0x03	0x57	OPM	CRC8

OPM:

OPM provides the masks to enable the INT pin if an operation status flag is set to ready. The operation status flags itself are not affected.

Table 238. OPM (reset value 0x00)

Bit	Symbol	Access	Value	Description
7 to 4	RFU	W0		Reserved for future use
3	SM_RXREADY	W		Set INT pin if the immo receiver ready flag is set
			0	Disable
			1	Enable
2	SM_TXREADY	W		Set INT pin if the LF transmission ready flag is set
			0	Disable
			1	Enable

Bit	Symbol	Access	Value	Description
1	SM_IMPMEAS	W		Set INT pin if the impedance measurement ready flag is set
			0	Disable
			1	Enable
0	SM_DIAG	W		Set INT pin if diagnostics ready flag is set
			0	Disable
			1	Enable

18.3.2 Response

Table 239.SET_OP_MASK response

LEN	CMD	STAT	CRC8
0x03	0x57	STAT	CRC8

18.3.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

19. Program download

19.1 DOWNLOAD_PROG

DOWNLOAD_PROG writes program code into the embedded application code memory (RAM) to be executed afterwards by the command EXECUTE_PROG. The program code size to be transmitted with 1 command is limited to 250 bytes, larger programs have to be downloaded in multiple steps.

The content can be checked using the command GET_PROG_SIG. No read access is granted to the code area.

Before starting a download, a power on reset (POR) shall be asserted to clear the RAM history and to ensure a safe setup.

19.1.1 Command

Table 240. DOWNLOAD_PROG command

LEN	CMD	PARAM			CRC8
LEN	0x30	ADDR_L	ADDR_H	{PROGi}	CRC8

ADDR:

ADDR specifies the address of the first byte where the data is stored. Downloading user defined program code in several steps, the address has to be adopted accordingly for each program segment.

With increasing index i the address increases by one byte and all PROGi entries are byte related (LSByte of word first). The valid address range is from 0x8000 to 0xA3FF.

Table 241. ADDR (reset value 0xXXXX)

Bit	Symbol	Access	Value	Description
15 to 0	ADDR[15:0]	W	0xXXXX	Byte address where the data is stored

PROGi:

The user defined program code to be stored. LEN shall be even such that the number of the {PROGi} fields is also even as the code area is used by 16-bit words only. Any program code segment can be written independently and no checks are done by the device for completeness and correctness of the code itself, e.g. for overriddes or gaps between address segments. The program code consistency could be checked with the signature.

19.1.2 Response

Table 242. DOWNLOAD_PROG response

LEN	CMD	STAT	CRC8
0x03	0x30	STAT	CRC8

19.1.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 243. DOWNLOAD_PROG status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	Invalid address used

19.2 START_PROG

START_PROG executes the downloaded embedded user defined application code memory (RAM) content. This function branches always to the start address of the program downloaded into the code RAM (via the command DOWNLOAD_PROG).

The response is sent before the execution is started. As soon as the last byte is sent to the host, the code gets executed. Please note that no check for a correct RAM download is performed.

It is important to notice that if there is no download available the RAM is uninitialized. In this case interrogating START_PROG by the host controller resets the device (memory management power on reset POR).

19.2.1 Command

Table 244. START_PROG command

LEN	CMD	CRC8
0x02	0x31	CRC8

19.2.2 Response

Table 245. EXECUTE_PROG response

LEN	CMD	STAT	CRC8
0x03	0x31	STAT	CRC8

19.2.3 Status flags

In failure case the generically used device status flags are set dependent on the event or failure condition.

19.3 GET_PROG_SIG

GET_PROG_SIG reads the signature of the downloaded program.

Signature calculation

```
// Note: data points to an even number (NumBytes) of bytes
static void _CalculateSignature( uint8_t * data, uint32_t
NumBytes )
{
uint16_t * wdata = (uint16_t *) data;
uint32_t NumWords = NumBytes / 2;
uint32_t hash = 0;
uint32_t i;
// CRC-24 algo works on 16-bit words.
for ( i = 0; i < NumWords; i++ ) {
   UpdateHashCRC24( *wdata++, &hash );
}
printf("The calculated signature: %.6X\n", hash );
```

```
}

// Simplified portable version of hash algorithm.
// output_data contains the last CRC-24 value.
static void _UpdateHashCRC24( uint16_t input_data, uint32_t *
pu32_hash )
{
uint32_t u32_reg; // only lower 24 bits are used
uint8_t carry;
u32_reg = *pu32_hash;
carry = (uint8_t) u32_reg & 0x01U;
u32_reg >>= 1;
if ( carry )
{
  u32_reg ^= 0x00D80000U;
}
u32_reg ^= ( (uint32_t)input_data << 8 );
*pu32_hash = u32_reg;
}
```

It is important to notice that if there is no download available the RAM is uninitialized. In this case interrogating GET_PROG_SIG by the host controller resets the device (memory management power on reset POR).

19.3.1 Command

Table 246. GET_PROG_SIG command

LEN	CMD	PARAM		CRC8
0x04	0x33	SIZE_L	SIZE_H	CRC8

SIZE:

SIZE specifies the number of bytes for which the signature of the downloaded program is calculated. The signature calculation starts always at the start address of the program downloaded into the code RAM (via the command DOWNLOAD_PROG). As the program code shall have an even length, also SIZE is restricted to even values.

19.3.2 Response

Table 247. GET_PROG_SIG response

LEN	CMD	STAT	PARAM			CRC8
0x06	0x33	STAT	SIGN_L	SIGN_M	SIGN_H	CRC8

SIGN:

The program signature contains a 24 bit value representing the downloaded program code.

Table 248. SIGN (reset value 0x 00 00 00)

Bit	Symbol	Access	Value	Description
23 to 0	SIGN[23:0]	R		Program signature

19.3.3 Status flags

In failure case besides the generically used device status flags the following status flags are set dependent on the event or failure condition.

Table 249. GET_PROG_SIG status flags

Status byte	Status bit	Event or failure condition
STAT	SF_PAR	SIZE is larger than the length of the downloaded program

20. Dynamic characteristics

20.1 SPI command response delay times

Table 250. SPI command response delay times

T_{amb} = -40 to +105 °C, GND = 0 V, V_{BAT} = 8 V to 18 V, V_{IO} = 2.9 V to 5.5 V, f_C = 125 kHz, T_0 = 1/f_C, $t_{SPI,CLK}$ = 10 µs (100 kHz) or $t_{SPI,CLK}$ = 2 µs (500 kHz), C connected between pins VBAT and GND, Z_{ANT} = 10 to 20 Ω, Q ≤ 25, external components according to Data Sheet, full bridge operation with midlevel control, automatic NXP preamble PREAMB disabled. Unless otherwise specified.

Symbol[1]	Parameter	Conditions	Min	Typ	Max	Unit
$t_{RESP,DLY,CLEAR\_DIAG\_STATUS}$	CLEAR_DIAG_STATUS response delay	1 antenna	0.03		0.3	ms
		6 antennas	0.1		0.6	ms
$t_{RESP,DLY,CLEAR\_IMMO\_STATUS}$	CLEAR_IMMO_STATUS response delay		0.03		0.3	ms
$t_{RESP,DLY,CLEAR\_OP\_STATUS}$	CLEAR_OP_STATUS response delay	All parameters set	0.03		0.3	ms
$t_{RESP,DLY,CLEAR\_POR\_STATUS}$	CLEAR_POR_STATUS response delay	All parameters set	0.03		0.3	ms
$t_{RESP,DLY,CLEAR\_PROT\_STATUS}$	CLEAR_PROT_STATUS response delay	1 antenna	0.03		0.3	ms
		6 antennas	0.03		0.6	ms
$t_{RESP,DLY,CLEAR\_TEMP\_STATUS}$	CLEAR_TEMP_STATUS response delay		0.03		0.3	ms
$t_{RESP,DLY,CLEAR\_WUP\_STATUS}$	CLEAR_WUP_STATUS response delay		0.03		0.3	ms
$t_{RESP,DLY,CONFIG\_ADVANCED}$	CONFIG_ADVANCED response delay		0.03		0.6	ms
$t_{RESP,DLY,CONFIG\_BOOST}$	CONFIG_BOOST response delay		0.03		0.3	ms
$t_{RESP,DLY,CONFIG\_DEVICE}$	CONFIG_DEVICE response delay		0.03		0.3	ms
$t_{RESP,DLY,CONFIG\_IMMO\_BPLM}$	CONFIG_IMMO_BPLM response delay		0.03		0.3	ms
$t_{RESP,DLY,CONFIG\_IMMO\_DRIVER}$	CONFIG_IMMO_DRIVER response delay	Different setting for TX and RX current	0.2		0.6	ms
$t_{RESP,DLY,CONFIG\_IMMO\_RECEIVER}$	CONFIG_IMMO_RECEIVER response delay		0.03		0.3	ms
$t_{RESP,DLY,CONFIG\_LC\_DRIVER}$	CONFIG_LC_DRIVER response delay	1 antenna	0.03		0.3	ms
		6 antennas	0.1		0.6	ms
$t_{RESP,DLY,CONFIG\_LF\_DRIVER}$	CONFIG_LF_DRIVER response delay different value for CURS and CURM	1 antenna (except TX4)	0.2		0.9	ms
		1 antenna TX4	0.6		1.2	ms
		6 antennas	1.8		4.2	ms
$t_{RESP,DLY,CONFIG\_SPI}$	CONFIG_SPI response delay		0.03		0.3	ms
$t_{RESP,DLY,CONFIG\_TEMP}$	CONFIG_TEMP response delay		0.03		0.6	ms
$t_{RESP,DLY,CONFIG\_TIMER\_POLLING}$	CONFIG_TIMER_POLLING response delay	1 parameter block	0.1		0.6	ms
		Maximum number of parameter blocks	0.9		2.1	ms

Symbol[1]	Parameter	Conditions	Min	Typ	Max	Unit
$t_{RESP,DLY,CONFIG\_WUP}$	CONFIG_WUP response delay		0.03		0.3	ms
$t_{RESP,DLY,CONFIG\_WUP\_POLLING}$	CONFIG_WUP_POLLING response delay	1 parameter block	0.1		0.6	ms
		Maximum number of parameter blocks	0.3		1.2	ms
$t_{RESP,DLY,DOWNLOAD\_PROG}$	DOWNLOAD_PROG response delay	250 byte payload	0.2		0.6	ms
$t_{RESP,DLY,ECHO\_SPI}$	ECHO_SPI response delay	2 byte payload	0.03		0.3	ms
		250 bytes payload	0.6		1.2	ms
t_{RESP,DLY,GET_ANT_IMP}	GET_ANT_IMP response delay	1 antenna	0.03		0.3	ms
		6 antennas	0.03		0.6	ms
$t_{RESP,DLY,GET\_ANT\_IMP\_EFFECTIVE}$	GET_ANT_IMP_EFFECTIVE response delay	1 antenna	0.03		0.3	ms
		6 antennas	0.03		0.6	ms
$t_{RESP,DLY,GET\_DIAG\_STATUS}$	GET_DIAG_STATUS response delay	1 antenna [2]	0.03		0.3	ms
		6 antennas [2]	0.03		0.6	ms
$t_{RESP,DLY,GET\_IMMO\_RESPONSE}$	GET_IMMO_RESPONSE response delay	Maximum length of RX data	0.1		0.6	ms
$t_{RESP,DLY,GET\_OP\_STATUS}$	GET_OP_STATUS response delay	[2]	0.03		0.3	ms
$t_{RESP,DLY,GET\_POR\_STATUS}$	GET_POR_STATUS response delay	[2]	0.03		0.3	ms
$t_{RESP,DLY,GET\_PROG\_SIG}$	GET_PROG_SIG response delay	Maximum RAM size	10.2		13.5	ms
$t_{RESP,DLY,GET\_PROT\_STATUS}$	GET_PROT_STATUS response delay	[2]	0.03		0.3	ms
$t_{RESP,DLY,GET\_TEMP\_STATUS}$	GET_TEMP_STATUS response delay	[2]	0.03		0.3	ms
$t_{RESP,DLY,GET\_VERSION}$	GET_VERSION response delay		0.03		0.3	ms
$t_{RESP,DLY,GET\_WUP\_STATUS}$	GET_WUP_STATUS response delay	[2]	0.03		0.3	ms
$t_{RESP,DLY,MEAS\_ANT\_IMP}$	MEAS_ANT_IMP response delay, different current for CURS and CURM, different current for TXCUR and RXCUR	1 antenna	0.03		0.3	ms
		6 antennas	0.03		0.6	ms
$t_{RESP,DLY,MEAS\_ANT\_IMP\_ADVANCED}$	MEAS_ANT_IMP_ADVANCED response delay different current for CURS and CURM, different current for TXCUR and RXCUR	1 antenna	0.03		0.3	ms
		6 antennas	0.03		0.6	ms
t_{RESP,DLY,SET_ANT_IMP}	SET_ANT_IMP response delay, different current for CURS and CURM, different current for TXCUR and RXCUR	1 antenna (except TX4)	0.3		0.9	ms
		1 antenna TX4	0.3		1.5	
		6 antennas	1.5		4.8	ms

Symbol[1]	Parameter	Conditions	Min	Typ	Max	Unit
$t_{RESP,DLY,SET\_IMMO\_MASK}$	SET_IMMO_MASK response delay		0.03		0.3	ms
t_{RESP,DLY,SET_LC_DATA}	SET_LC_DATA response delay	Maximum length of 120 bits	0.03		0.3	ms
t_{RESP,DLY,SET_LF_DATA}	SET_LF_DATA response delay	Maximum length of 120 bits	0.03		0.3	ms
t_{RESP,DLY,SET_OP_MASK}	SET_OP_MASK response delay		0.03		0.3	ms
t_{RESP,DLY,SET_POR}	SET_POR response delay	[2] [3]	0.03		0.6	ms
$t_{RESP,DLY,SET\_PROT\_MASK}$	SET_PROT_MASK response delay		0.03		0.3	ms
$t_{RESP,DLY,SET\_TEMP\_MASK}$	SET_TEMP_MASK response delay		0.03		0.3	ms
$t_{RESP,DLY,SET\_WUP\_MASK}$	SET_WUP_MASK response delay		0.03		0.3	ms
$t_{RESP,DLY,START\_DIAG}$	START_DIAG response delay DIAGPAR = 0x1F	1 antenna	0.03		0.3	ms
		6 antennas	0.03		0.6	ms
$t_{RESP,DLY,START\_IMMO}$	START_IMMO response delay		2.4		3.0	ms
$t_{RESP,DLY,START\_IMMO\_TRANSCEIVE}$	START_IMMO_TRANSCEIVE response delay	4 bits sent, 40 bits received	0.03		0.3	ms
		Maximum bits sent and maximum bits received	0.1		0.6	ms
$t_{RESP,DLY,START\_IMMO\_TRANSMIT}$	START_IMMO_TRANSMIT response delay	4 bits sent	0.03		0.3	ms
		Maximum bits sent	0.03		0.3	ms
$t_{RESP,DLY,START\_LF\_TRANSMIT}$	START_LF_TRANSMIT response delay	1 parameter block $t_{SPI,CLK}$ = 10 µs (100 kHz)	0.03		0.6	ms
		Maximum number of parameter blocks $t_{SPI,CLK}$ = 10 µs (100 kHz)	0.03		0.3	ms
		1 parameter block $t_{SPI,CLK}$ = 2 µs (500 kHz)	0.03		0.6	ms
		Maximum number of parameter blocks $t_{SPI,CLK}$ = 2 µs (500 kHz)	0.03		2.1	ms
$t_{RESP,DLY,START\_LF\_TRANSMIT\_DATA}$	START_LF_TRANSMIT_DATA response delay	2 bytes data $t_{SPI,CLK}$ = 10 µs (100 kHz)	0.03		0.6	ms
		Maximum data length $t_{SPI,CLK}$ = 10 µs (100 kHz)	0.03		0.3	ms
		2 bytes data $t_{SPI,CLK}$ = 2 µs (500 kHz)	0.03		0.6	ms
		Maximum data length $t_{SPI,CLK}$ = 2 µs (500 kHz)	0.03		0.6	ms
$t_{RESP,DLY,START\_PROG}$	START_PROG response delay		0.03		0.3	ms
$t_{RESP,DLY,START\_SLEEP}$	START_SLEEP response delay		0.03		0.3	ms
$t_{RESP,DLY,START\_SLEEP\_FORCED}$	START_SLEEP_FORCED response delay	[2]	0.1		0.9	ms

Symbol[1]	Parameter	Conditions	Min	Typ	Max	Unit
$t_{RESP,DLY,START\_TIMER\_POLLING}$	START_TIMER_POLLING response delay		0.03		0.3	ms
$t_{RESP,DLY,STOP\_IMMO}$	STOP_IMMO response delay	After previous START_IMMO [2]	0.2		0.9	ms
$t_{RESP,DLY,STOP\_LF\_TRANSMIT}$	STOP_LF_TRANSMIT response delay	After previous START_LF_TRANSMIT [2]	0.2		0.9	ms

[1] Any additional processing activity of the integrated µController (like protection handling, SPI communication, WUP events, LF activation/deactivation, channel change) may enlarge the SPI response delay time.

[2] SPI response delay time may be enlarged, if sent during the execution of non-blocking commands

[3] In case any LF is stopped the time will increase by 0.3 ms

20.2 SPI command operation delay times

Table 251. SPI command operation delay times

T_{amb} = -40 to +105 °C, GND = 0 V, V_{BAT} = 8 V to 18 V, V_{IO} = 2.9 V to 5.5 V, f_C = 125 kHz, $T_0 = 1/f_C$, $t_{SPI,CLK}$ = 10 µs (100 kHz) or $t_{SPI,CLK}$ = 2 µs (500 kHz),, C connected between pins VBAT and GND, Z_{ANT} = 10 to 20 Ω, Q ≤ 25, external components according to Data Sheet, full bridge operation with midlevel control, automatic NXP preamble PREAMB disabled. Unless otherwise specified.

Symbol[1]	Parameter	Conditions	Min	Typ	Max	Unit
$t_{OP,DLY,MEAS\_ANT\_IMP}$	MEAS_ANT_IMP operation delay,different current for CURS and CURM, different current for TXCUR and RXCUR	1 antenna (except TX4)	1.5		11.1	ms
		1 antenna TX4	1.8		11.7	ms
		6 antennas	7.2		45.3	ms
$t_{OP,DLY,MEAS\_ANT\_IMP\_ADVANCED}$	MEAS_ANT_IMP_ADVANCED operation delay, different current for CURS and CURM, different current for TXCUR and RXCUR	1 antenna [2]	1.5		12.0	ms
		1 antenna Maximum antenna detuning ±8%	1.5		8.1	ms
		6 antennas [2]	7.2		61.5	ms
		6 antennas Maximum antenna detuning ±8%	7.2		37.5	ms
$t_{OP,DLY,START\_DIAG}$	START_DIAG operation delay DIAGPAR = 0x1F	1 antenna [2]	1.5		7.2	ms
		6 antennas [2]	7.2		22.5	ms
$t_{OP,DLY,START\_IMMO\_TRANSCEIVE}$	START_IMMO_TRANSCEIVE operation delay	4 bits sent, 40 bits received	15.0		18.0	ms
		Maximum bits sent and maximum bits received	178.8		183.9	ms
$t_{OP,DLY,START\_IMMO\_TRANSMIT}$	START_IMMO_TRANSMIT operation delay	4 bits sent	0.6		1.8	ms
		Maximum bits sent	44.1		46.2	ms
$t_{OP,DLY,START\_LF\_TRANSMIT}$	START_LF_TRANSMIT operation delay	1 parameter block $t_{SPI,CLK}$ = 10 µs (100 kHz)	1.2		3.3	ms
		Maximum number of parameter blocks $t_{SPI,CLK}$ = 10 µs (100 kHz)	71.3		86.6	ms
		1 parameter block $t_{SPI,CLK}$ = 2 µs (500 kHz)	1.2		3.6	ms
		Maximum number of parameter blocks $t_{SPI,CLK}$ = 2 µs (500 kHz)	71.3		88.1	ms
$t_{OP,DLY,START\_LF\_TRANSMIT\_DATA}$	START_LF_TRANSMIT_DATA operation delay	2 bytes data $t_{SPI,CLK}$ = 10 µs (100 kHz)	2.7		6. 6	ms
		Maximum data length $t_{SPI,CLK}$ = 10 µs (100 kHz)	121.2		125. 4	ms
		2 bytes data $t_{SPI,CLK}$ = 2 µs (500 kHz)	2.7		6.9	ms
		Maximum data length $t_{SPI,CLK}$ = 2 µs (500 kHz)	121.2		125.7	ms

[1] Any additional processing activity of the integrated µController (like protection handling, SPI communication, WUP events, LF activation/deactivation, channel change) may enlarge the SPI command operation delay time

[2] Time determined without antennas connected

20.3 Further dynamic characteristics

Table 252. Further dynamic characteristics

T_{amb} = -40 to +105 °C, GND = 0 V, V_{BAT} = 8 V to 18 V, V_{IO} = 2.9 V to 5.5 V, f_C = 125 kHz, T_0 = 1/f_C, C connected between pins VBAT and GND, Z_{ANT} = 10 to 20 Ω, Q ≤ 25, external components according to Data Sheet, full bridge operation with midlevel control.
Unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{CH,CHG}$	See NJJ29C0B data sheet					
$t_{LF,START}$	See NJJ29C0B data sheet					
$t_{LF,STOP}$	See NJJ29C0B data sheet					
$t_{MRK3,WAKE}$	See NJJ29C0B data sheet					
$t_{MRK3,\ SHUTDOWN}$	See NJJ29C0B data sheet					

21. Revision history

Table 253. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
NJJ29C0B_SPI_2	20171020	Product data sheet	-	NJJ29C0B_SPI_1
Modifications	• Data sheet status changed from Preliminary data sheet to Product data sheet • Revision history reset			
NJJ29C0B_SPI_1	20170818	Preliminary data sheet	-	-
Modifications	• Initial version			

22. Legal information

22.1 Data sheet status

Document status[1][2]	Product status[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification or product development
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification
Product [short] data sheet	Production	This document contains the product specification

[1] Please consult the most recently issued document before initiating or completing a design.

[2] The term 'short data sheet' is explained in section "Definitions".

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices.

22.2 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

Product specification — The information and data provided in a Product data sheet shall define the specification of the product as agreed between NXP Semiconductors and its customer, unless NXP Semiconductors and customer have explicitly agreed otherwise in writing. In no event however, shall an agreement be valid in which the NXP Semiconductors product is deemed to offer functions and qualities beyond those described in the Product data sheet.

22.3 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use in automotive applications — This NXP Semiconductors product has been qualified for use in automotive applications. Unless otherwise agreed in writing, the product is not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on a weakness or default in the customer application/use or the application/use of customer's third party customer(s) (hereinafter both referred to as "Application"). It is customer's sole responsibility to check whether the NXP Semiconductors product is suitable and fit for the Application planned. Customer has to do all necessary testing for the Application in order to avoid a default of the Application and the product. NXP Semiconductors does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

Quick reference data — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

22.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

23. Contact information

For sales office addresses, please send an email to: salesaddresses@nxp.com

For additional information, please visit: http://www.nxp.com

24. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 20 Oct 2017
Document identifier: NJJ29C0B_SPI_2