

OM6220 Specification

Version 0.5

Liuc
2023/4/19

1 Introduction	4
1.1 Features.....	5
1.2 Block.....	6
2 Pin Information.....	7
2.1 Pad assignment	7
2.2 Pad function.....	8
2.3 Reference design.....	8
3 Absolute maximum ratings	9
4 Operating conditions.....	10
5 Electrical specifications	11
6 Radio Control	12
6.1 Operational Modes.....	12
6.1.1 State diagram	12
6.1.2 Power-Down Mode.....	14
6.1.3 Sleep Mode	14
6.1.4 Standby Modes	14
6.1.5 RX mode.....	14
6.1.6 TX mode	15
6.1.7 Operational modes configuration.....	15
6.1.8 Timing Information.....	15
6.2 Air data rate	16
6.3 RF channel frequency	16
6.4 PA control	16
6.5 RX/TX control	16
7 Protocol Engine	17
7.1 Features.....	17
7.2 Protocol engine overview	17
7.3 Protocol engine packet format	18
7.3.1 Preamble	18
7.3.2 Address	18
7.3.3 Guard	18
7.3.4 Packet Control Field (PCF).....	18
7.3.5 Payload	19
7.3.6 CRC (Cyclic Redundancy Check)	20
7.3.7 Automatic packet assembly	20
7.3.8 Automatic packet disassembly.....	22
7.4 Automatic packet transaction handling.....	23
7.4.1 Auto Acknowledgement.....	23
7.4.2 Auto Retransmission (ART).....	23
7.5 Protocol engine flowcharts	24
7.5.1 PTX operation.....	24

7.5.2 PRX operation	26
7.6 Multi-Slave	28
7.7 Protocol engine timing.....	29
7.8 Protocol engine transaction diagram.....	31
7.8.1 Single transaction with ACK packet and interrupts.....	31
7.8.2 Single transaction with a lost packet.....	31
7.8.3 Single transaction with a lost ACK packet	32
7.8.4 Single transaction with ACK payload packet	32
7.8.5 Single transaction with ACK payload packet and lost packet	33
7.8.6 Two transactions with ACK payload packet and the first ACK packet lost.....	33
7.8.7 Two transactions where max retransmissions is reached.....	34
8 Data and control interface.....	35
8.1 Features.....	35
8.2 Functional description	35
8.3 SPI operation	35
8.3.1 SPI commands	35
8.3.2 SPI timing	37
8.4 Data FIFO	38
8.5 Interrupt	38
9 Register map	40
10 Glossary of Terms	41
11 Package Information	42
12 Ordering Information.....	43

1 Introduction

OM6220 is a single chip 2.4GHz transceiver with an embedded baseband protocol engine, suitable for ultralow power wireless applications. OM6220 is designed for operation in the worldwide ISM frequency band at 2.400 - 2.4835GHz.

To design a radio system with the OM6220, you simply need an MCU (microcontroller) and a few external passive components.

You can operate and configure the OM6220 through a Serial Peripheral Interface (SPI). The register map, which is accessible through the SPI, contains all configuration registers in the OM6220 and is accessible in all operation modes of the chip.

The embedded baseband protocol engine is based on packet communication and supports various modes from manual operation to advanced autonomous protocol operation. Internal FIFO ensure a smooth data flow between the radio front end and the system's MCU. Protocol engine reduces system cost by handling all the high speed link layer operations.

The radio front end uses GFSK/FSK modulation. It has user configurable parameters like frequency channel, output power and air data rate. OM6220 supports an air data rate of 1Mbps and 2Mbps. The high air data rates combined with two power saving modes make the OM6220 very suitable for ultralow power designs.

The addition of internal filtering to OM6220 has improved the margins for meeting RF regulatory standards.

Internal voltage regulators ensure a high Power Supply Rejection Ratio (PSRR) and a wide power supply range.

1.1 Features

Features of the OM6220 include:

- Radio:
 - Compatible to OM6200
 - Support Long packet mode (Max 255 bytes @ 2.4G mode)
 - Common RX and TX interface
 - GFSK/FSK modulation
 - 1 and 2Mbps air data rate
 - 1MHz non-overlapping channel spacing at 1Mbps
 - 2MHz non-overlapping channel spacing at 2Mbps
- BLE:
 - BLE 5.1 Compatible
 - GAP-Peripheral, or Human Interface Devices
 - ATT / GATT – Client and Server
 - L2CAP
- Transmitter:
 - Programmable output power: -34 ~ +8dBm
 - 15mA at 0dBm output power
- Receiver:
 - Fast AGC for improved dynamic range
 - Integrated channel filters
 - 17mA at 2Mbps
 - -87dBm sensitivity at 2Mbps
 - -91dBm sensitivity at 1Mbps
- RF Synthesizer:
 - Fully integrated synthesizer
 - No external loop filter
 - Accepts low cost ± 60 ppm 16MHz crystal
- Protocol engine:
 - 1 to 32 bytes dynamic payload length
 - Automatic packet handling
 - Auto packet transaction handling
 - 3 data pipe for 1:3 star networks
- Power Management:
 - Integrated voltage regulator
 - 1.9 to 3.6V supply range
 - Idle modes with fast start-up times for advanced power management
 - 2 μ A Power Down mode
 - 40 μ A Standby mode
- Host Interface:
 - 3/4-pin hardware SPI
 - Max 8Mbps

- 1 separate 32 bytes TX and RX FIFO

1.2 Block

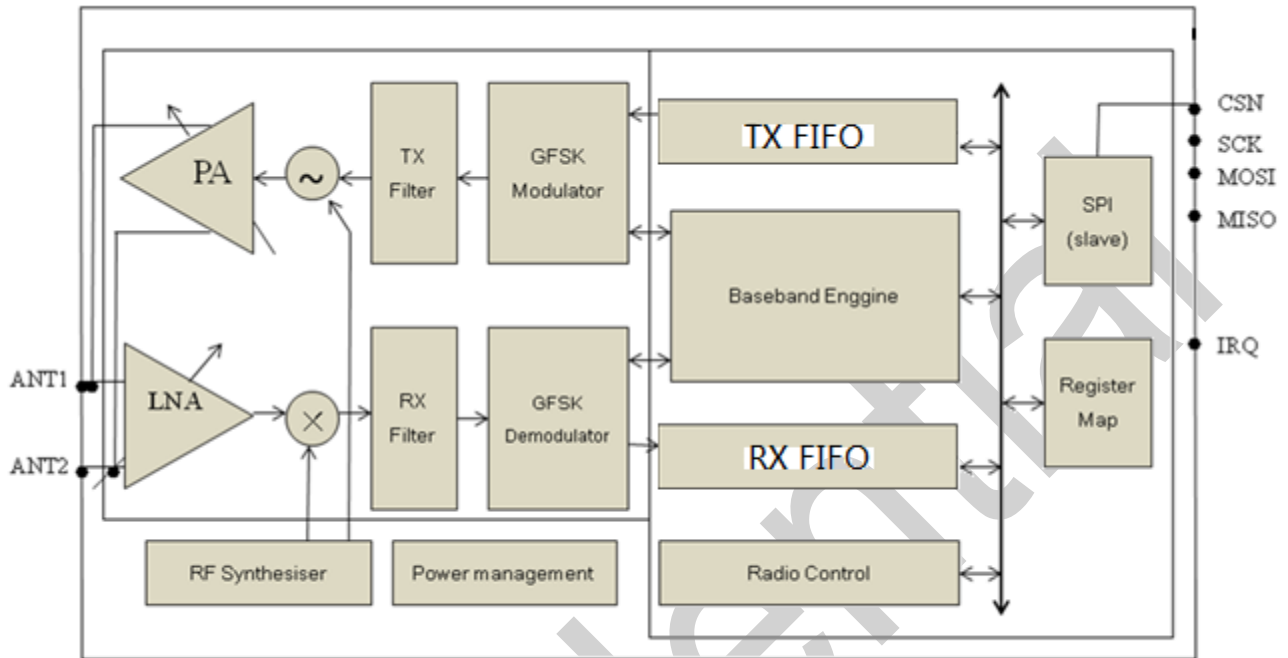


Figure1.1OM6220 block diagram

2 Pin Information

2.1 Pad assignment

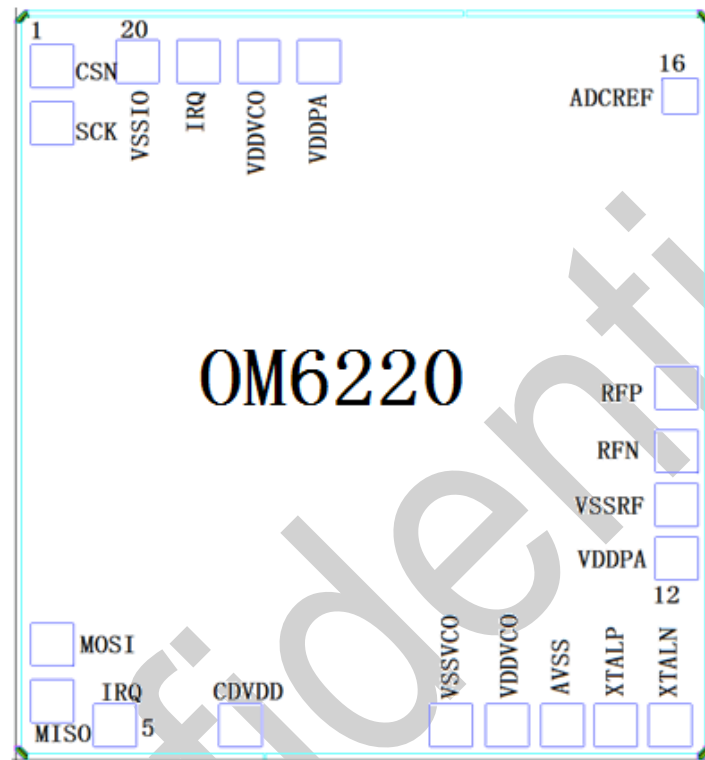


Figure2.1 OM6220 pad assignment

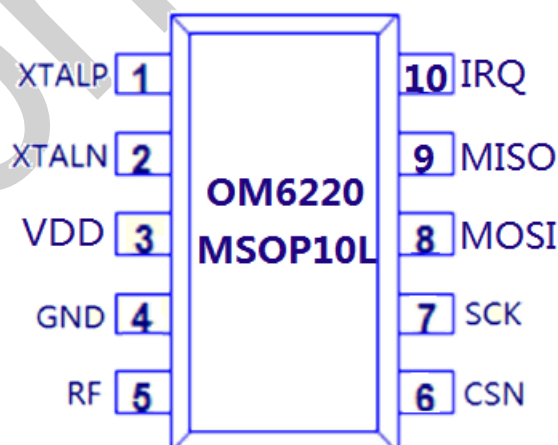


Figure2.2 OM6220 MSOP10L package

2.2 Pad function

Pad	Name	Pad function	Description
1	CSN	Digital Input	SPI Chip Select
2	SCK	Digital Input	SPI Clock
3	MOSI/SDA	Digital Input/output	SPI Slave Data Input or 3-wire SPI SDA
4	MISO	Digital Output	SPI Slave Data Output, with tri-state option
5	IRQ	Digital Output	Mask-able interrupt pin. Active low
6	CDVDD	Analog output	Internal digital supply output for de-coupling purposes
7	VSSVCO	Power	Ground
8	VDDVCO	Power	Power Supply (+1.9- +3.6V)
9	AVSS	Power	Ground
10	XTALP	Analog input	Crystal+
11	XTALN	Analog input	Crystal-
12	VDDPA	Power	Power Supply (+1.9- +3.6V)
13	VSSRF	Power	Ground
14	RFN	Analog input	RF-
15	RFP	Analog input	RF+
16	ADCREF	Analog output	Internal ADC reference output for de-coupling purposes
17	VDDPA	Power	Power Supply (+1.9V- +3.6V)
18	VDDVCO	Power	Power Supply (+1.9V- +3.6V)
19	IRQ	Digital Output	Mask-able interrupt pin. Active low
20	VSSIO	Power	Ground

Table2.1 OM6220 pad function

2.3 Reference design

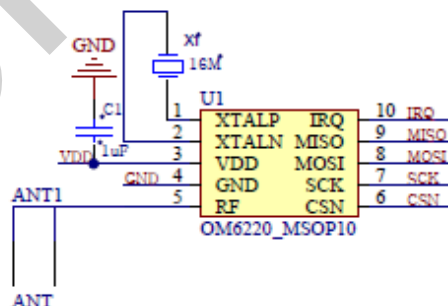


Figure2.3 OM6220 MSOP10L reference design

3 Absolute maximum ratings

Note: Exceeding one or more of the limiting values may cause permanent damage to OM6220.

Operating conditions	Minimum	Maximum	Units
Supply voltages			
V _{DD}		3.6	V
V _{SS}		0	V
Input voltage			
V _I	-0.3	5	V
Output voltage			
V _O	V _{SS} to V _{DD}	V _{SS} to V _{DD}	
Temperatures			
Operating Temperature	-40	+85	°C
Storage Temperature	-40	+125	°C

Table3.1 Absolute maximum ratings

4 Operating conditions

Symbol	Parameter(condition)	Notes	Min.	Typ.	Max.	Units
V _{DD}	Supply voltage		1.9	3.0	3.6	V
V _{DD}	Supply voltage if input signals>3.6V		1.9	3.0	3.6	V
TEMP	Operating Temperature		-40	+27	+85	°C

Table 4.1 Operating conditions

5 Electrical specifications

Power consumption

No.	Parameter	Symbol	Conditions	MIN	TYP	MAX	UNIT
1	Current in power down	I _{pd}	Register retention		2		uA
2	Current in standby	I _{stb}	Crystal on		40		uA
5	Current in TX 0dBm	I _{tx}	PA under 0dBm		15		mA
8	Current in RX 2Mbps	I _{rx}	RX mode		17		mA
9	Operation frequency	F _{req}		2400		2525	MHz
10	PLL frequency step	Delta F			1		MHz
11	Freq deviation@1Mbps	D _f			250		KHz
12	Freq deviation@2Mbps	D _f			320		KHz

Receiver performance

No.	Parameter	Symbol	Conditions	MIN	TYP	MAX	UNIT
1	Max RX signal	P _{in,max}	<0.1% BER		-10		dBm
2	1Mbps	Sensitivity	<0.1%BER		-91		dBm
3	2Mbps	Sensitivity	<0.1%BER		-87		dBm

Transmitter performance

No.	Parameter	Symbol	Conditions	MIN	TYP	MAX	UNIT
1	Max Output Power	P _{max}	50ohm antenna		+8		dBm
2	Min Output Power	P _{min}	50ohm antenna		-34		dBm
3	RF power control range	P _r range	50ohm antenna		42		dB

Crystal performance

No.	Parameter	Symbol	Conditions	MIN	TYP	MAX	UNIT
1	Crystal Frequency	F _{xtal}		16	16	16	MHz
2	Tolerance	D _{fxtal}		-60		+60	ppm
3	Load capacitance	C _{xtal}			12		pF

DC characteristic

No.	Parameter	Symbol	Conditions	MIN	TYP	MAX	UNIT
1	HIGH level input	V _{high}		0.7VDD		3.6	V
2	Low level input	V _{low}		0		0.3VDD	V

6 Radio Control

This chapter describes the OM6220 radio transceiver's operating modes and the parameters used to control the radio.

The OM6220 has a built-in state machine that controls the transitions between the chip's operating modes. The state machine takes input from user defined register values and internal signals.

6.1 Operational Modes

You can configure the OM6220 in power down, sleep, standby, RX or TX mode. This section describes these modes in detail.

6.1.1 State diagram

Figure 6.1 shows the operating modes and how they function. There are three types of distinct states highlighted in the state diagram:

- Recommended operating mode: is a recommended state used during normal operation.
- Possible operating mode: is a possible operating state, but is not used during normal operation.
- Transition state: is a time limited state used during startup of the oscillator and settling of the PLL.

When the VDD reaches 1.9V or higher OM6220 enters the Power on reset state where it remains in reset until entering the PowerDown mode.

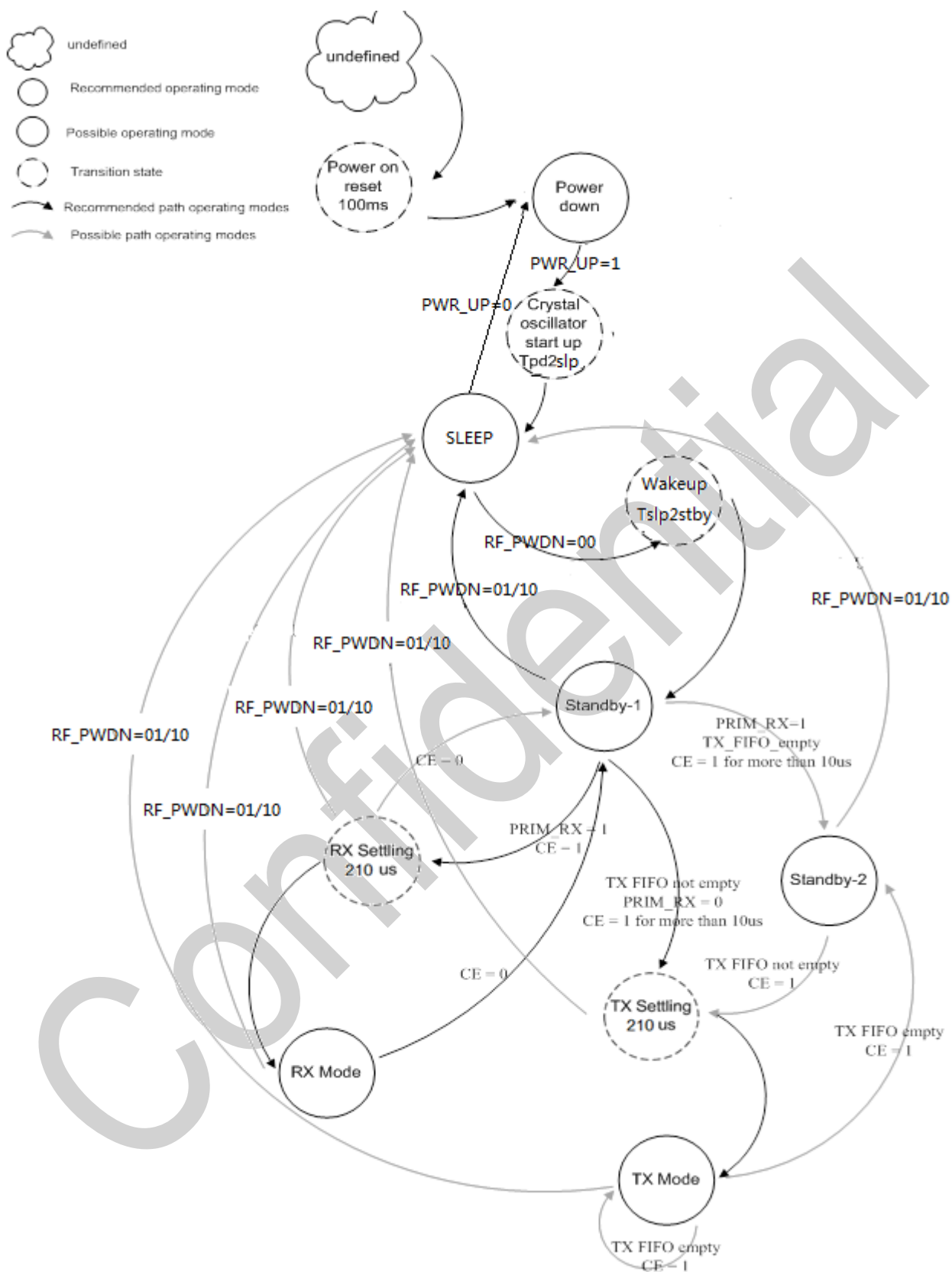


Figure 6.1 Radio control state diagram

6.1.2 Power-Down Mode

In power down mode, the OM6220 is disabled using minimize average current consumption. All register values available are maintained and the SPI is kept active, enabling change of configuration and the uploading/down-loading of data registers. For starting up times see Table 6.2. Power down mode is entered by setting the PWR_UP bit in the CONFIG register low, when OM6220 is in the sleep mode.

6.1.3 Sleep Mode

In sleep down mode, the OM6220 is disabled using minimize average current consumption. All register values available are maintained and the SPI and crystal oscillator are kept active.

6.1.4 Standby Modes

6.1.4.1 Standby-I mode

By setting the PWR_UP bit in the CONFIG register to 1, the device enters sleep mode, then clear RF_PWDN two bit, the device enters Standby-I mode. Standby-I mode is used to minimize average current consumption while maintaining short start up times. In this mode only part of the crystal oscillator is active. Change to active modes only happens if CE is set high and when CE is set low, the OM6220 returns to standby-I mode from both the TX and RX modes.

6.1.4.2 Standby-II mode

In standby-II mode extra clock buffers are active and more current is used compared to standby-I mode. The OM6220 enters standby-II mode if CE is held high on a PTX device with an empty TX FIFO. If a new packet is uploaded to the TX FIFO, the PLL immediately starts and the packet is transmitted after the normal PLL settling delay (210µs). Register values are maintained and the SPI can be activated during both standby modes.

Notice: From Standby-I mode to standby-II mode CE more than 20µs

6.1.5 RX mode

The RX mode is an active mode where the OM6220 radio is used as a receiver. To enter this mode, the chip must have the PWR_UP bit, PRIM_RX bit and the CE pin set high.

In RX mode the receiver demodulates the signals from the RF channel, constantly presenting the demodulated data to the baseband protocol engine. The baseband protocol engine constantly searches for a valid packet. If a valid packet is found (by a matching address and a valid CRC) the payload of the packet is presented in a vacant slot in the RX FIFO. If the RX FIFO is full, the received packet is discarded.

The chip remains in RX mode until the MCU configures it to standby-I mode or power down mode. However, if the automatic protocol features in the baseband protocol engine are enabled, the chip can enter other modes in order to execute the protocol.

6.1.6 TX mode

The TX mode is an active mode for transmitting packets. To enter this mode, the chip must have the PWR_UP bit set high, PRIM_RX bit set low, a payload in the TX FIFO and a high pulse on the CE for more than 20μs.

The OM6220 stays in TX mode until it finishes transmitting a packet. If CE = 0, the chip returns to standby-I mode. If CE = 1, the status of the TX FIFO determines the next action. If the TX FIFO is not empty the OM6220 remains in TX mode and transmits the next packet. If the TX FIFO is empty the chip goes into standby-II mode.

6.1.7 Operational modes configuration

The following (table 6.1) describes how to configure the operational modes:

Mode	PWR_UP register	RF_PWDN register	PRIM_RX register	CE Command	FIFO state
RX mode	1	00	1	1	-
TX mode	1	00	0	1	Data TX FIFO. Will empty in TX FIFO
TX mode	1	00	0	Minimum 20us high pulse	Data TX FIFO. Will empty in TX FIFO
Standby-I mode	1	00	-	0	No ongoing packet transmission
Standby-II mode	1	00	0	1	TX FIFO empty
Sleep Mode	1	01/10	-	-	-
Power Down	0	01	-	-	-

Table6.1 the OM6220 main modes

6.1.8 Timing Information

The timing information in this section relates to the transitions between modes and the timing for the CE pin. The transition from TX mode to RX mode or vice versa is the same as the transition from the standby modes to TX mode or RX mode, as described in Table6.2

Name	The chip	Max.	Min.	Comments
T _{pd2stb}	Power Down→Standby mode		2ms	Depended on crystal
T _{stby2a}	Standby mode→TX/RX mode		130us	
T _{hce}	Minimum CE high		20us	
T _{pece2csn}	Delay from CE positive edge to CSN low		4us	

Table6.2 operational timing of the OM6220 chip

For OM6220 to go from power down mode to TX or RX mode it must first pass through sleep and stand-by mode. There must be a delay of T_{pd2slp} and T_{slp2stby} (see Table 6.2) after the OM6220 leaves power down mode before the CE is set high.

Note: If VDD is turned off the register value is lost and you must configure chip before entering the TX or RX modes.

6.2 Air data rate

The air data rate is the modulated signaling rate the chip uses when transmitting and receiving data. It can be 1Mbps or 2Mbps. Using lower air data rate gives better receiver sensitivity than higher air data rate. But high air data rate gives lower average current consumption and reduced probability of on-air collisions. The air data rate is set by the RF_DR bit in the RF_SETUP register. A transmitter and a receiver must be programmed with the same air data rate to communicate with each other.

6.3 RF channel frequency

The RF channel frequency determines the center of the channel used by the chip. The channel occupies a bandwidth of less than 1MHz at 1Mbps and a bandwidth of less than 2MHz at 2Mbps. The chip can operate on frequencies from 2.400GHz to 2.525GHz. The programming resolution of the RF channel frequency setting is 1MHz.

At 2Mbps the channel occupies a bandwidth wider than the resolution of the RF channel frequency setting. To ensure non-overlapping channels in 2Mbps mode, the channel spacing must be 2MHz or more. At 1Mbps, the channel bandwidth is the same or lower than the resolution of the RF frequency.

The RF channel frequency is set by the RF_CH register according to the following formula:

$$F_0 = 2400 + \text{RF_CH [MHz]}$$

You must program a transmitter and a receiver with the same RF channel frequency to communicate with each other.

6.4 PA control

The PA (Power Amplifier) control is used to set the output power from the chip power amplifier. In TX mode PA control has four programmable steps, see Table 6.3.

The PA control is set by the PA_PWR bits in the RF_SETUP register.

SPI RF-SETUP (PA_PWR[3:0])	RF output power	DC current consumption
1111	8dBm	37mA
0110	4dBm	21.5mA
0011	0dBm	15mA
0010	-4dBm	12.5mA

Conditions: VDD = 3.0V, VSS = 0V, TA = 27°C

Table 6.3 RF output power setting for the OM6220

6.5 RX/TX control

The RX/TX control is set by PRIM_RX bit in the CONFIG register and sets the OM6220 chip in transmit/receive mode.

7 Protocol Engine

Protocol engine is a packet based data link layer that features automatic packet assembly and timing, automatic acknowledgement and retransmissions of packets. Protocol engine enables the implementation of ultralow power and high performance communication. The Protocol engine features enable significant improvements of power efficiency for bi-directional and un-directional systems, without adding complexity on the host controller side.

7.1 Features

The main features of Protocol engine are:

- 1 to 32 bytes dynamic payload length
- Automatic packet handling
- Automatic packet transaction handling
 - Auto Acknowledgement with payload
 - Auto retransmit
- 3 data pipe for 1:3 star networks

7.2 Protocol engine overview

Protocol engine uses self-defined protocol for automatic packet handling and timing. During transmit, Protocol engine assembles the packet and clocks the bits in the data packet for transmission. During receive, Protocol engine constantly searches for a valid address in the demodulated signal. When Protocol engine finds a valid address, it processes the rest of the packet and validates it by CRC. If the packet is valid the payload is moved into a vacant slot in the RX FIFO. All high speed bit handling and timing is controlled by protocol engine.

Protocol engine features automatic packet transaction handling for the easy implementation of a reliable bi-directional data link. A protocol engine packet transaction is a packet exchange between two transceivers, with one transceiver acting as the Primary Receiver (PRX) and the other transceiver acting as the Primary Transmitter (PTX). A protocol engine packet transaction is always initiated by a packet transmission from the PTX, the transaction is complete when the PTX has received an acknowledgment packet (ACK packet) from the PRX. The PRX can attach user data to the ACK packet enabling a bi-directional data link.

The automatic packet transaction handling works as follows:

1. You begin the transaction by transmitting a data packet from the PTX to the PRX. Protocol engine automatically sets the PTX in receive mode to wait for the ACK packet.
2. If the packet is received by the PRX, Protocol engine automatically assembles and transmits an acknowledgment packet (ACK packet) to the PTX before returning to receive mode.
3. If the PTX does not receive the ACK packet immediately, Protocol engine automatically retransmits the original data packet after a programmable delay and sets the PTX in receive mode to wait for the ACK packet.

In Protocol engine it is possible to configure parameters such as the maximum number of retransmits and the delay from one transmission to the next retransmission. All automatic handling is done without the involvement of the MCU.

7.3 Protocol engine packet format

The format of the Protocol engine packet is described in this section. The Protocol engine packet contains a preamble field, address field, packet control field, payload field and a CRC field. Figure 7.1 shows the packet format with MSB to the left.

Preamble 1 byte	Address 4-5 byte	2byte guard	Packet control field 9 bit	Payload 0-32 bytes	CRC 1-2 bytes
-----------------	------------------	-------------	----------------------------	--------------------	---------------

Figure 7.1 Protocol engine packet with payload (0-32 bytes)

7.3.1 Preamble

The preamble is a bit sequence used to synchronize the receivers demodulator to the incoming bit stream. The preamble is one byte long and is either 01010101 or 10101010. If the first bit in the address is 1 the preamble is automatically set to 10101010 and if the first bit is 0 the preamble is automatically set to 01010101. This is done to ensure there are enough transitions in the preamble to stabilize the receiver.

7.3.2 Address

This is the address for the receiver. An address ensures that the packet is detected and received by the correct receiver, preventing accidental cross talk between multiple OM6220 systems. You can configure the address field width in the AW register to be 5 bytes or 4 bytes address.

7.3.3 Guard

Figure 7.1 shows the format of the 2byte guard packet has better synchronous characteristics.

7.3.4 Packet Control Field (PCF)

Figure 7.2 shows the format of the 9 bit packet control field, MSB to the left.

Payload length 6bit	PID 2bit	NO_ACK 1bit
---------------------	----------	-------------

Figure 7.2 Packet control field (PCF)

The packet control field contains a 6 bit payload length field, a 2 bit PID (Packet Identity) field and a 1 bit NO_ACK flag.

7.3.4.1 Payload length

This 6 bit field specifies the length of the payload in bytes. The length of the payload can be from 0 to 32 bytes.

Coding: 000000 = 0 byte (only used in empty ACK packets. The 0 length packet also need to be read out use R_RX_PAYLOAD with no data following) 100000 = 32 byte, 100001 = Don't care.

This field is only used if the Dynamic Payload Length function is enabled.

7.3.4.2 PID (Packet identification)

The 2 bit PID field is used to detect if the received packet is new or retransmitted. PID prevents the PRX operation from presenting the same payload more than once to the MCU. The PID field is incremented at the TX side for each new packet received through the SPI. The PID and CRC field are used by the PRX operation to determine if a packet is retransmitted or new. When several data packets are lost on the link, the PID fields may become equal to the last received PID. If a packet has the same PID as the previous packet, the RF transceiver compares the CRC sums from both packets. If the CRC sums are also equal, the last received packet is considered a copy of the previously received packet and discarded.

7.3.4.3 No Acknowledgment flag (NO_ACK)

The Selective Auto Acknowledgement feature controls the NO_ACK flag.

This flag is only used when the auto acknowledgement feature is used. Setting the flag high, tells the receiver that the packet is not to be auto acknowledged.

On the PTX you can set the NO_ACK flag bit in the Packet Control Field with this command:
W_TX_PAYLOAD_NOACK

However, the function must first be enabled in the FEATURE register by setting the EN_DYN_ACK bit. When you use this option, the PTX goes directly to standby-I mode after transmitting the packet. The PRX does not transmit an ACK packet when it receives the packet.

7.3.5 Payload

The payload is the user defined content of the packet. It can be 0 to 32 bytes wide and is transmitted on-air when it is uploaded to the device.

Protocol engine provides two alternatives for handling payload lengths; static and dynamic.

The default is static payload length. With static payload length all packets between a transmitter and a receiver have the same length. Static payload length is set by the RX_PW_Px registers on the receiver side. The payload length on the transmitter side is set by the number of bytes clocked into the TX_FIFO and must equal the value in the RX_PW_Px register on the receiver side.

Dynamic Payload Length (DPL) is an alternative to static payload length. DPL enables the transmitter to send packets with variable payload length to the receiver. This means that for a system with different payload lengths it is not necessary to scale the packet length to the longest payload.

With the DPL feature the OM6220 can decode the payload length of the received packet automatically instead of using the RX_PW_Px registers. The MCU can read the length of the received payload by using the R_RX_PL_WID command.

Note: Always check if the packet width reported is 32 bytes or shorter when using the R_RX_PL_WID command. If its width is longer than 32 bytes then the packet contains errors and must be discarded. Discard the packet by using the Flush_RX command.

In order to enable DPL the EN_DPL bit in the FEATURE register must be enabled. In RX mode the DYNPD register must be set. A PTX that transmits to a PRX with DPL enabled must have the DPL_P0 bit in DYNPD set.

7.3.6 CRC (Cyclic Redundancy Check)

The CRC is the error detection mechanism in the packet. It may either be 1 or 2 bytes and is calculated over the address, Packet Control Field and Payload.

The polynomial for 1 byte CRC is $X^8 + X^2 + X + 1$. Initial value 0Xff.

The polynomial for 2 byte CRC is $X^{16} + X^{12} + X^5 + 1$. Initial value 0Xffff.

The number of bytes in the CRC is set by the CRCO bit in the CONFIG register. No packet is accepted by protocol engine if the CRC fails.

7.3.7 Automatic packet assembly

The automatic packet assembly assembles the preamble, address, packet control field, payload and CRC to make a complete packet before it is transmitted.

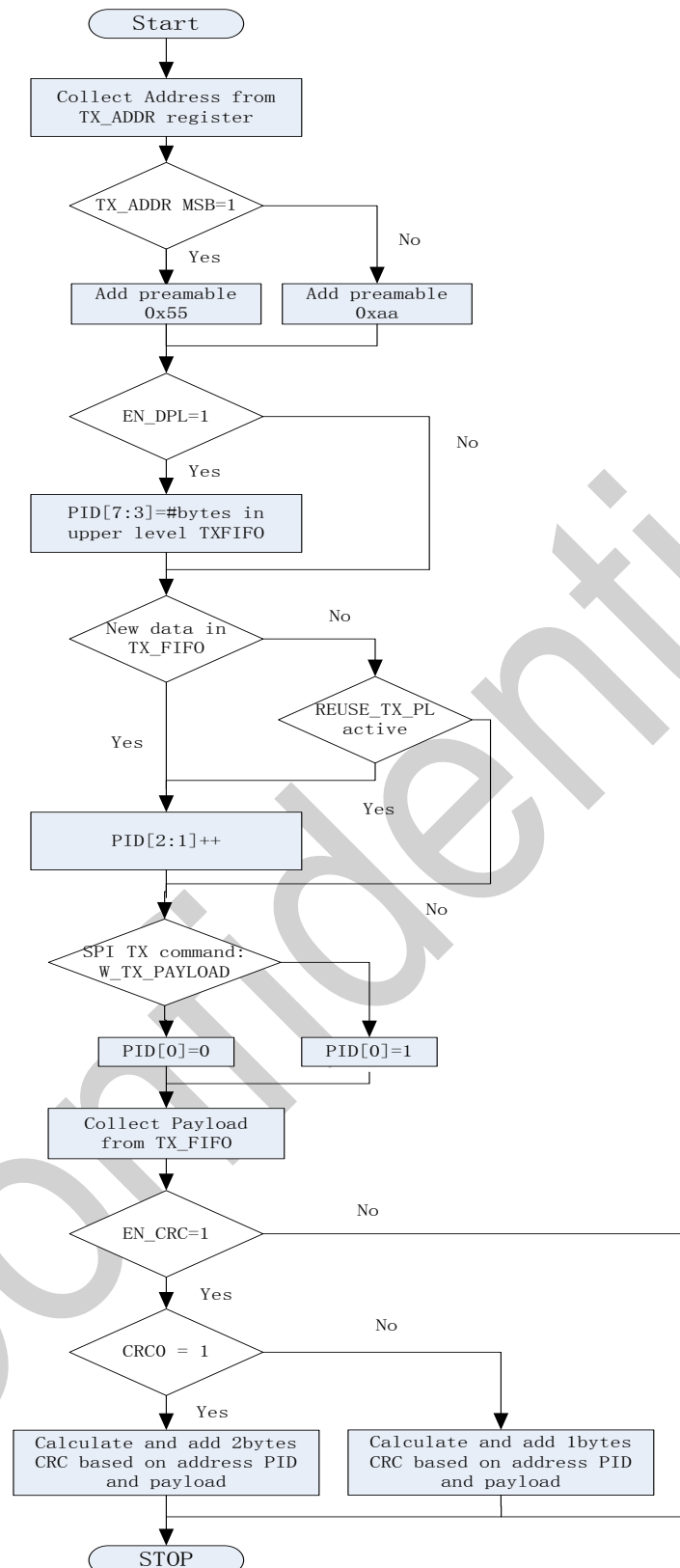


Figure 7.3 Automatic packet assembly

7.3.8 Automatic packet disassembly

After the packet is validated, Protocol engine disassembles the packet and loads the payload into the RX FIFO, and asserts the RX_DR IRQ.

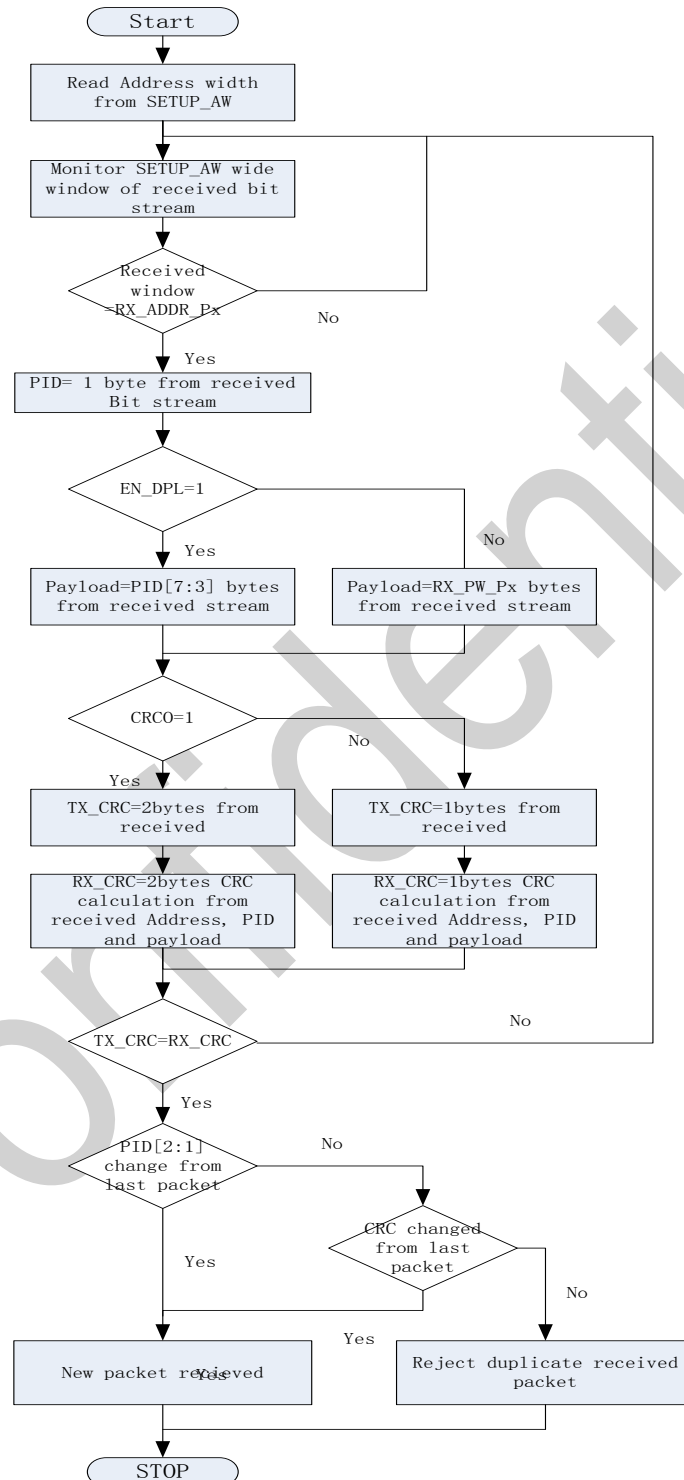


Figure 7.4 Automatic packet disassembly

7.4 Automatic packet transaction handling

Protocol engine features two functions for automatic packet transaction handling; auto acknowledgement and auto re-transmit.

7.4.1 Auto Acknowledgement

Auto Acknowledgment is a function that automatically transmits an ACK packet to the PTX after it has received and validated a packet. The Auto Acknowledgement function reduces the load of the system MCU and reduces average current consumption. The Auto Acknowledgement feature is enabled by setting the EN_AA register.

Note: If the received packet has the NO_ACK flag set, auto acknowledgement is not executed.

An ACK packet can contain an optional payload from PRX to PTX. In order to use this feature, the Dynamic Payload Length (DPL) feature must be enabled. The MCU on the PRX side has to upload the payload by clocking it into the TX FIFO by using the W_ACK_PAYLOAD command. The payload is pending in the TX FIFO (PRX) until a new packet is received from the PTX. The RF transceiver can have three ACK packet payloads pending in the TX FIFO (PRX) at the same time.

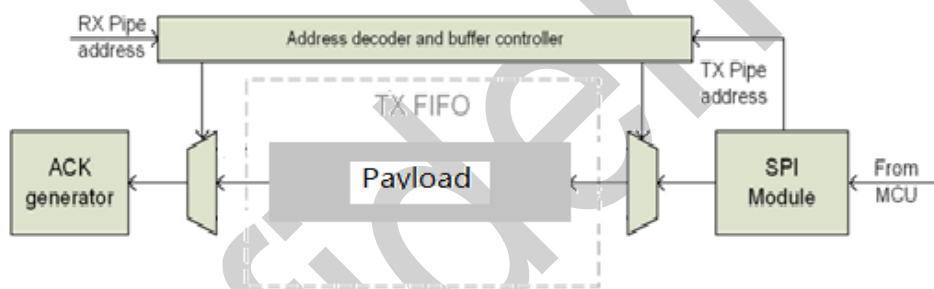


Figure 7.5 TX FIFO (PRX) with pending payloads

Figure 7.5 shows how the TX FIFO (PRX) is operated when handling pending ACK packet payloads. From the MCU the payload is clocked in with the W_ACK_PAYLOAD command. The address decoder and buffer controller ensure that the payload is stored in a vacant slot in the TX FIFO (PRX). When a packet is received, the address decoder and buffer controller are notified with the PTX address. This ensures that the right payload is presented to the ACK generator.

If the TX FIFO (PRX) contains more than one payload to a PTX, payloads are handled using the first in–first out principle. The TX FIFO (PRX) is blocked if all pending payloads are addressed to a PTX where the link is lost. In this case, the MCU can flush the TX FIFO (PRX) by using the FLUSH_TX command.

In order to enable Auto Acknowledgement with payload the EN_ACK_PAY bit in the FEATURE register must be set.

7.4.2 Auto Retransmission (ART)

The auto retransmission is a function that retransmits a packet if an ACK packet is not received. It is used in an Auto Acknowledgement system on the PTX. When a packet is not acknowledged, you can set the number of times it is allowed to retransmit by setting the ARC bits in the SETUP_RETR register. PTX enters RX mode and waits a time period for an ACK packet each time a packet is transmitted. The amount of time the PTX is in RX mode is based on the following conditions:

- Auto Retransmit Delay (ARD) has elapsed.
- No address match within 256 μ s.
- After received packet (CRC correct or not) if address match within 256 μ s.

The RF transceiver asserts the TX_DS IRQ when the ACK packet is received.

The RF transceiver enters standby-I mode if there is no more un-transmitted data in the TX FIFO and the CE is low. If the ACK packet is not received, the RF transceiver goes back to TX mode after a delay defined by ARD and retransmits the data. This continues until acknowledgment is received, or the maximum number of retransmits is reached.

Two packet loss counters are incremented each time a packet is lost, ARC_CNT and PLOS_CNT in the OBSERVE_TX register. The ARC_CNT counts the number of retransmissions for the current transaction. You reset ARC_CNT by initiating a new transaction. The PLOS_CNT counts the total number of retransmissions since the last channel change. You reset PLOS_CNT by writing to the RF_CH register. It is possible to use the information in the OBSERVE_TX register to make an overall assessment of the channel quality.

The ARD defines the time from the end of a transmitted packet to when a retransmit starts on the PTX. ARD is set in SETUP_RETR register in steps of 256 μ s. A retransmit is made if no ACK packet is received by the PTX.

There is a restriction on the length of ARD when using ACK packets with payload. The ARD time must never be shorter than the sum of the startup time and the time on-air for the ACK packet.

- For 2 Mbps data rate and 5-byte address; 15 byte is maximum ACK packet payload length for ARD=256 μ s (reset value).
- For 1 Mbps data rate and 5-byte address; 5 byte is maximum ACK packet payload length for ARD=256 μ s (reset value).

ARD=512 μ s is long enough for any ACK payload length in 1 or 2 Mbps mode.

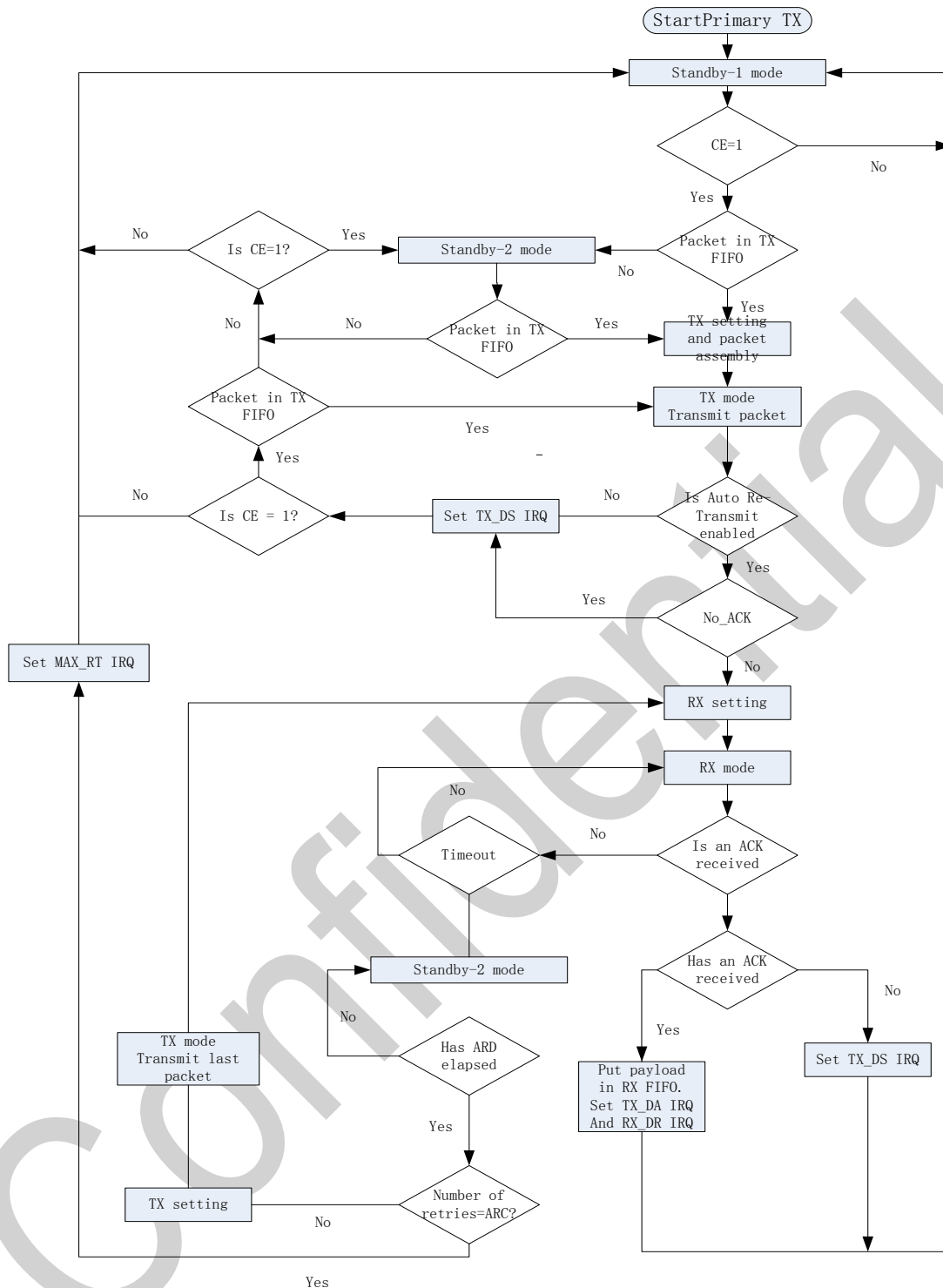
As an alternative to Auto Retransmit it is possible to manually set the RF transceiver to retransmit a packet a number of times. This is done by the REUSE_TX_PL command. The MCU must initiate each transmission of the packet with a pulse on the CE pin when this command is used.

7.5 Protocol engine flowcharts

This section contains flowcharts outlining PTX and PRX operation in Protocol engine.

7.5.1 PTX operation

The flowchart in Figure 7.6 outlines how a RF transceiver configured as a PTX behaves after entering standby-I mode.



Note: Protocol engine operation is outlined with a dashed square.

Figure 7.6PTX operations in Protocol engine

Activate PTX mode by setting the CE high. If there is a packet present in the TX FIFO the RF transceiver enters TX mode and transmits the packet. If Auto Retransmit is enabled, the state machine checks if the NO_ACK flag is set. If it is not set, the RF transceiver enters RX mode to receive an ACK packet. If the received ACK packet is empty, only the TX_DS IRQ is asserted. If the ACK packet contains a payload, both TX_DS IRQ and RX_DR IRQ are asserted simultaneously before the RF transceiver returns to standby-I mode.

If the ACK packet is not received before timeout occurs, the RF transceiver returns to standby-II mode. It stays in standby-II mode until

the ARD has elapsed. If the number of retransmits has not reached the ARC, the RF transceiver enters TX mode and transmits the last packet once more.

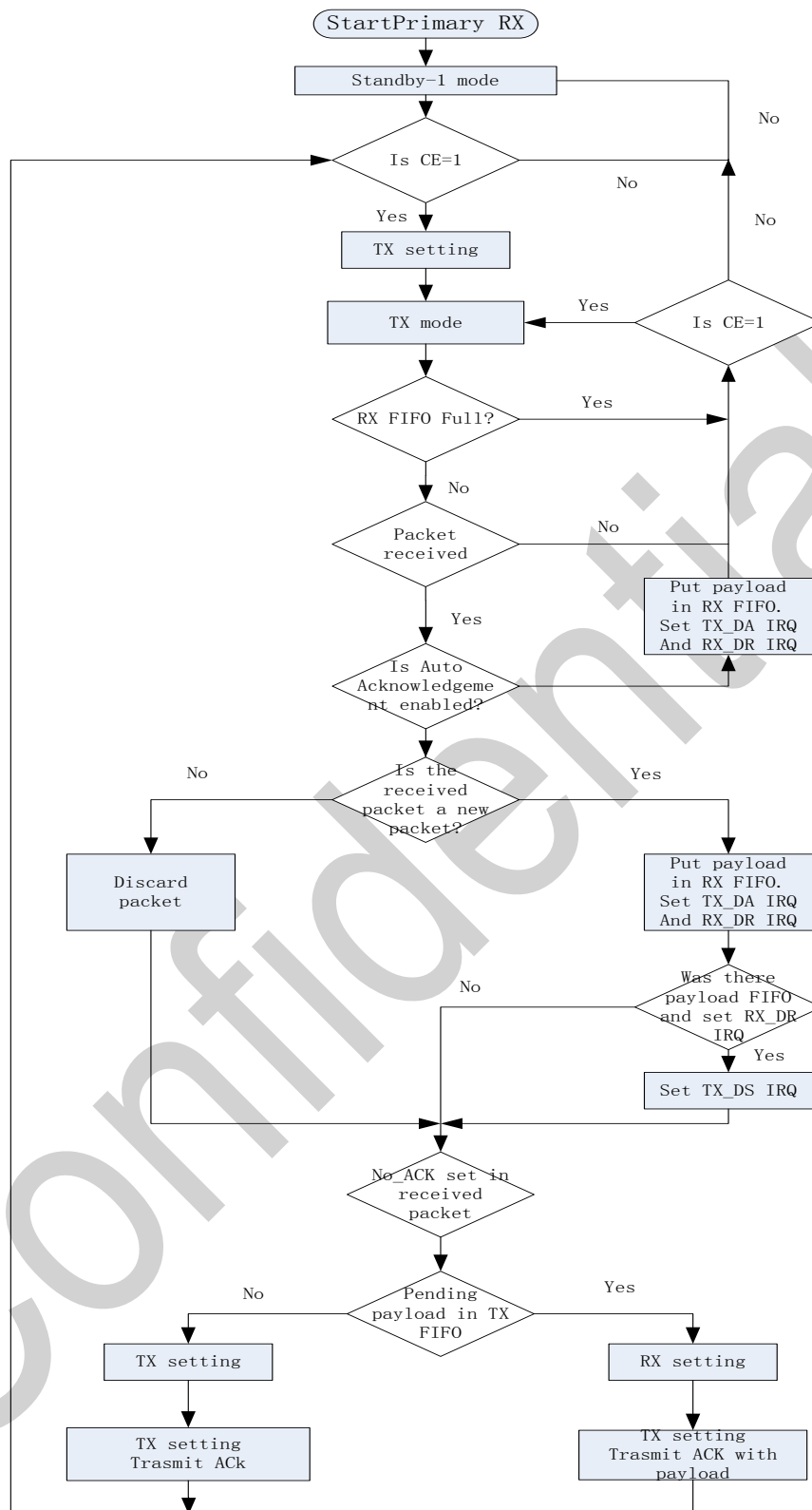
While executing the Auto Retransmit feature, the number of retransmits can reach the maximum number defined in ARC. If this happens, the RF transceiver asserts the MAX_RT IRQ and returns to standby-I mode.

If the CE bit in the RFCON register is high and the TX FIFO is empty, the RF transceiver enters Standby-II mode.

7.5.2 PRX operation

The flowchart in Figure 7.7 outlines how a RF transceiver configured as a PRX behaves after entering standby-I mode.

Confidential



Note: Protocol engine operation is outlined with a dashed square.

Figure7.7PRX operations in Protocol engine

Activate PRX mode by setting the CE bit in the RFCON register high. The RF transceiver enters RX mode and starts searching for packets. If a packet is received and Auto Acknowledgement is enabled, the RF transceiver decides if the packet is new or a copy of a previously received packet. If the packet is new payload is made available in the RX FIFO and the RX_DR IRQ is asserted. If the last

received packet from the transmitter is acknowledged with an ACK packet with payload, the TX_DS IRQ indicates that the PTX received the ACK packet with payload. If the No_ACK flag is not set in the received packet, the PRX enters TX mode. If there is a pending payload in the TX FIFO it is attached to the ACK packet. After the ACK packet is transmitted, the RF transceiver returns to RX mode.

A copy of a previously received packet might be received if the ACK packet is lost. In this case, the PRX discards the received packet and transmits an ACK packet before it returns to RX mode.

7.6 Multi-Slave

Multi-Slave is a feature used in RX mode that contains a set of three parallel data pipes with unique addresses. A data pipe is a logical channel in the physical RF channel. Each data pipe has its own physical address (data pipe address) decoding in the RF transceiver.

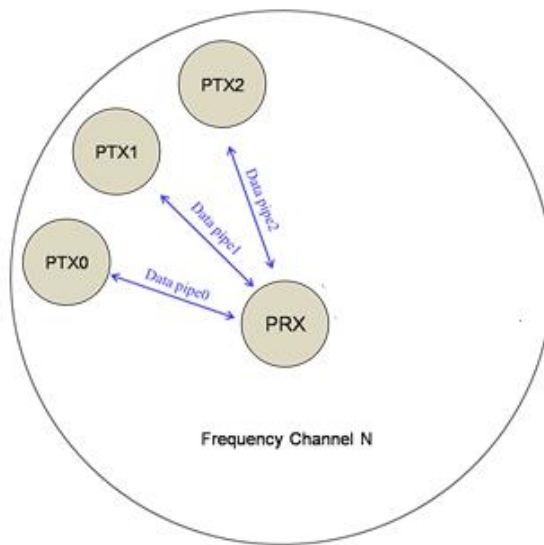


Figure 7.8 PRX using Multi-Slave

The RF transceiver configured as PRX (primary receiver) can receive data addressed to three different data pipes in one frequency channel as shown in Figure 7.8. Each data pipe has its own unique address and can be configured for individual behavior.

Up to three RF transceivers configured as PTX can communicate with one RF transceiver configured as PRX. All data pipe addresses are searched for simultaneously. Only one data pipe can receive a packet at a time. All data pipes can perform Protocol engine functionality.

The following settings are common to all data pipes:

- CRC enabled/disabled (CRC always enabled when Protocol engine is enabled)
- CRC encoding scheme
- RX address width
- Frequency channel
- Air data rate
- LNA gain

The data pipes are enabled with the bits in the EN_RXADDR register. By default only data pipe 0 and 1 are enabled. Each data pipe address is configured in the RX_ADDR_PX registers.

Note: Always ensure that none of the data pipes have the same address.

Each pipe can have up to a 5 byte configurable address. Data pipes 0-2 share the four most significant address bytes. The LSByte must be unique for all six pipes. Figure 7.9 is an example of how data pipes 0-2 are addressed. Only pipe 0 can have up to a 5 byte configurable address, other's pipes have 1 byte configurable address.

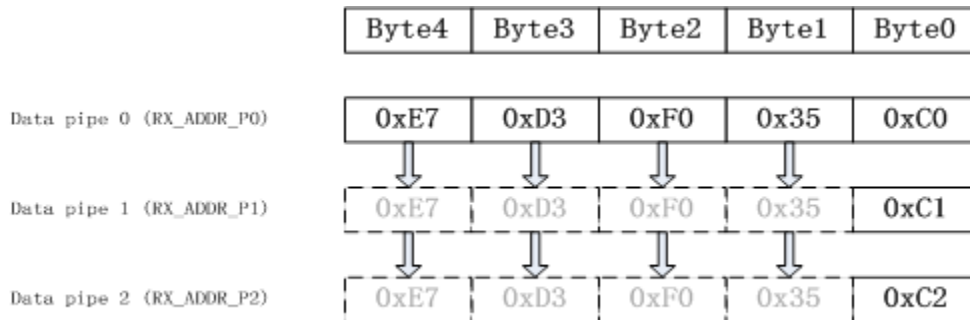


Figure 7.9 Addressing data pipes 0-2

The PRX, using Multi-Slave and Protocol engine, receives packets from more than one PTX. To ensure that the ACK packet from the PRX is transmitted to the correct PTX, the PRX takes the data pipe address where it received the packet and uses it as the TX address when transmitting the ACK packet. Figure 7.10 is an example of an address configuration for the PRX and PTX. On the PRX the RX_ADDR_Px, defined as the pipe address, must be unique. On the PTX the TX_ADDR must be the same as the RX_ADDR_P0 and as the pipe address for the designated pipe.

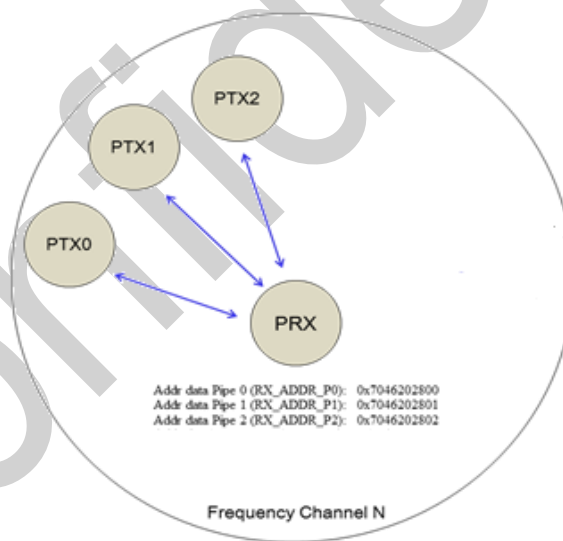


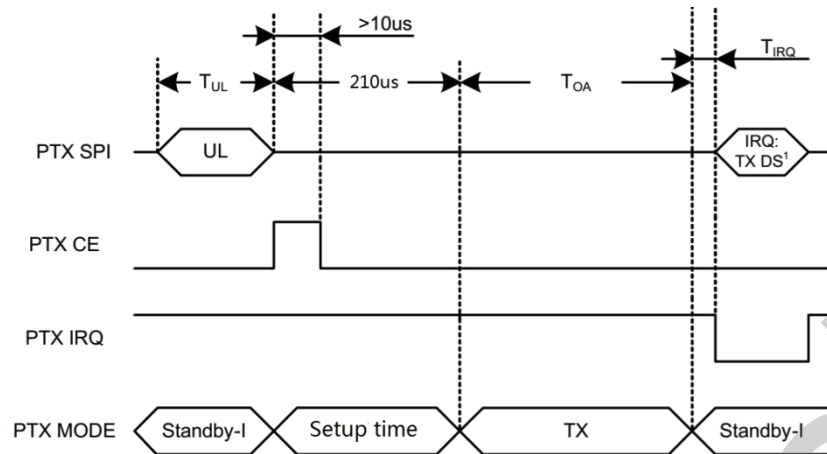
Figure 7.10 Example of data pipe addressing in Multi-Slave

Only when a data pipe receives a complete packet can other data pipes begin to receive data. When multiple PTXs are transmitting to a PRX, the ARD can be used to skew the auto retransmission so that they only block each other once.

7.7 Protocol engine timing

This section describes the timing sequence of Protocol engine and how all modes are initiated and operated. The Protocol engine timing is controlled through the Data and Control interface. The RF transceiver can be set to static modes or autonomous modes where the internal state machine controls the events. Each autonomous mode/sequence ends with a RFIRQ interrupt. All the interrupts are indicated

as IRQ events in the timing diagrams.



1 IRQ if No Ack is on.

$T_{IRQ}=3\mu s$ @ 1Mbps, @2Mbps

Figure 7.11 Transmitting one packet with NO_ACK on

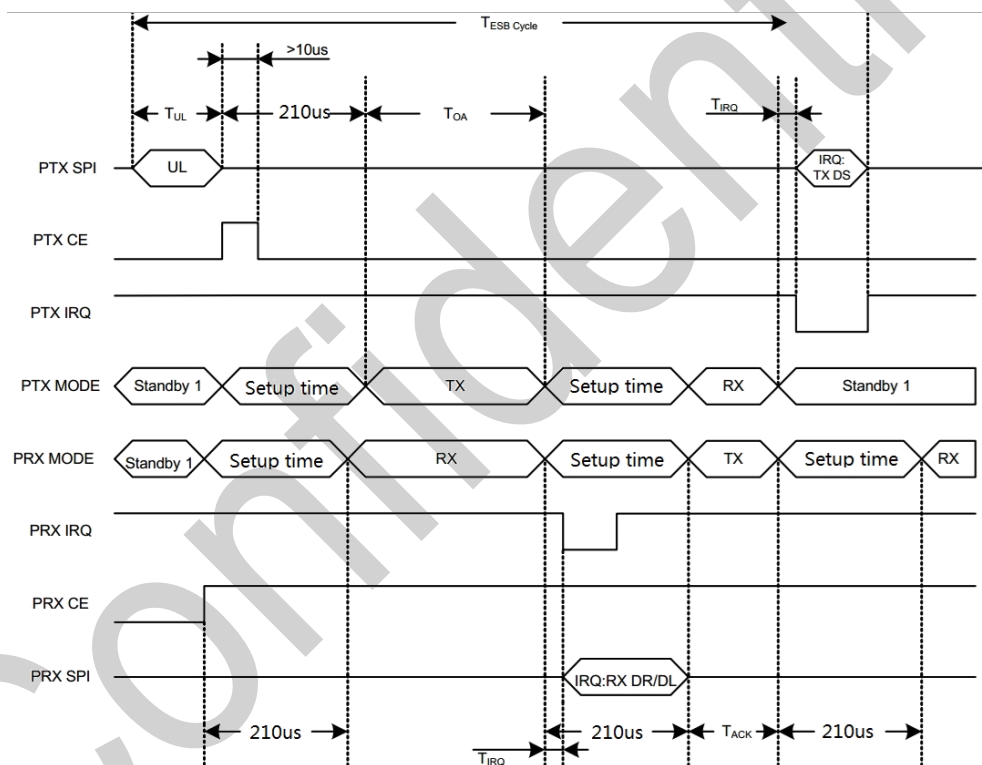


Figure 7.12 Timing of Protocol engine for one packet upload(2Mbps)

In Figure 7.12, the transmission and acknowledgement of a packet is shown. The PRX operation activates RX mode ($CE=1$), and the PTX operation is activated in TX mode ($CE=1$ for minimum $20\mu s$). After $210\mu s$ the transmission starts and finishes after the elapse of T_{OA} .

When the transmission ends the PTX operation automatically switches to RX mode to wait for the ACK packet from the PRX operation. When the PRX operation receives the packet it sets the interrupt for the host MCU and switches to TX mode to send an ACK. After the PTX operation receives the ACK packet it sets the interrupt to the MCU and clears the packet from the TX FIFO.

In Figure 7.13, the PTX timing of a packet transmission is shown when the first ACK packet is lost.

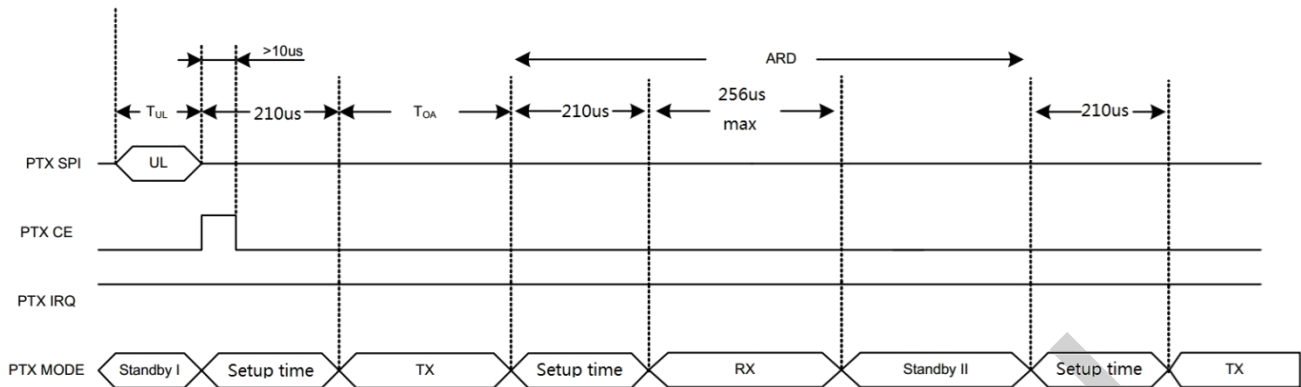


Figure 7.13 Timing of Protocol engine when the first ACK packet is lost (2 Mbps)

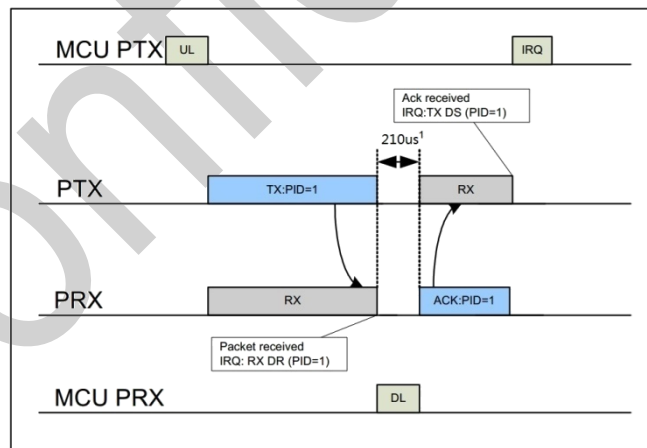
7.8 Protocol engine transaction diagram

This section describes several scenarios for the Protocol engine automatic transaction handling. The call outs in this section's figures indicate the IRQs and other events. For MCU activity the event may be placed at a different timeframe.

Note: The figures in this section indicate the earliest possible download (DL) of the packet to the MCU and the latest possible upload (UL) of payload to the transmitter.

7.8.1 Single transaction with ACK packet and interrupts

In Figure 7.14, the basic auto acknowledgement is shown. After the packet is transmitted by the PTX and received by the PRX the ACK packet is transmitted from the PRX to the PTX. The RX_DR IRQ is asserted after the packet is received by the PRX, whereas the TX_DS IRQ is asserted when the packet is acknowledged and the ACK packet is received by the PTX.



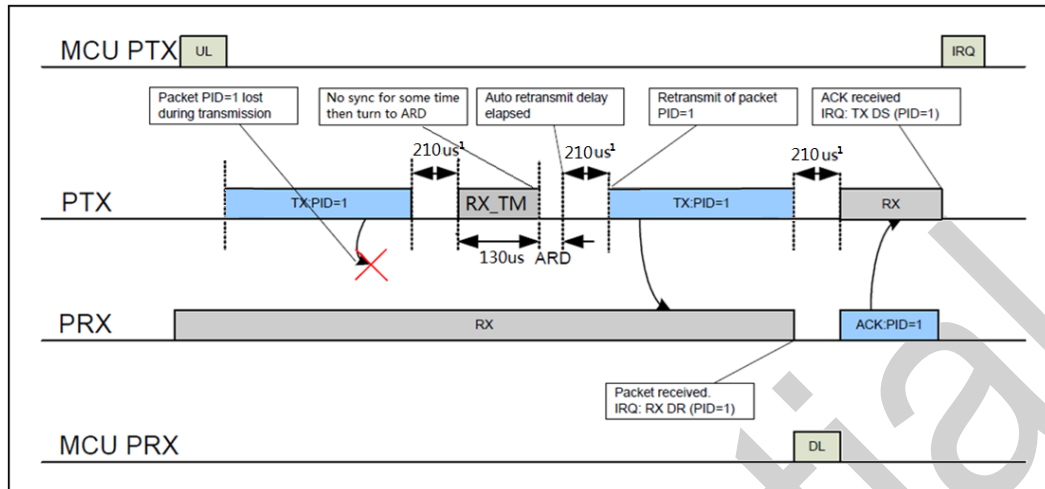
1 Radio Turn Around delay

Figure 7.14 TX/RX cycles with ACK and the according interrupts

7.8.2 Single transaction with a lost packet

Figure 7.15 is a scenario where a retransmission is needed due to loss of the first packet transmits. After the packet is transmitted, the PTX enters RX mode to receive the ACK packet. After the first transmission, the PTX waits a specified time (including setup time, RX_TM and ARD) for the ACK packet, if it is not in the specific time slot the PTX retransmits the packet as shown in Figure 7.15. PTX

will turn to RX mode after 210us setup time when packet is transmitted, after 130us RX timeout (RX_TM is RX timeout for PTX, it can be set shorter), then PTX turn to ARD (can be set to 0us, 256us, 512us to 3840us).



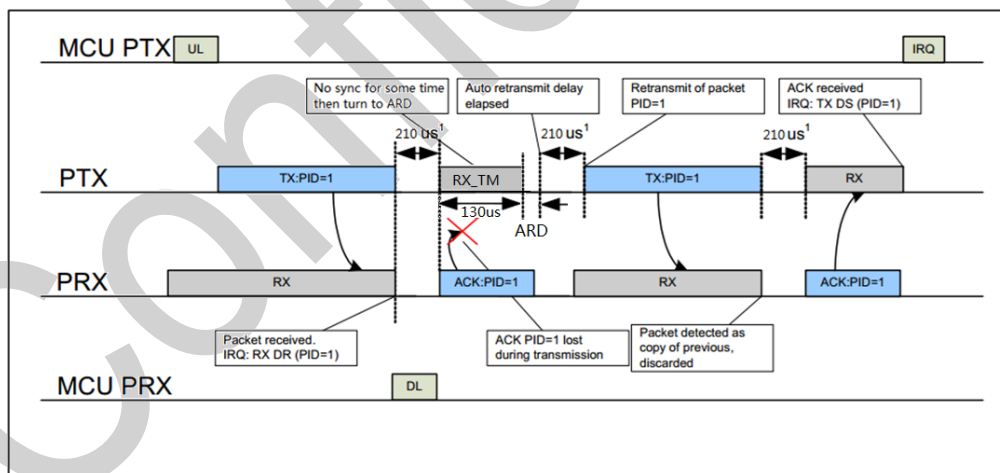
1 Radio Turn Around delay

Figure 7.15 TX/RX cycles with ACK and the according interrupts when the first packet transmit fails

When an address is detected the PTX stays in RX mode until the packet is received. When the retransmitted packet is received by the PRX (see Figure 7.15). The RX_DR IRQ is asserted and an ACK is transmitted back to the PTX. When the ACK is received by the PTX, the TX_DS IRQ is asserted.

7.8.3 Single transaction with a lost ACK packet

Figure 7.16 is a scenario where a retransmission is needed after a loss of the ACK packet. The corresponding interrupts are also indicated.



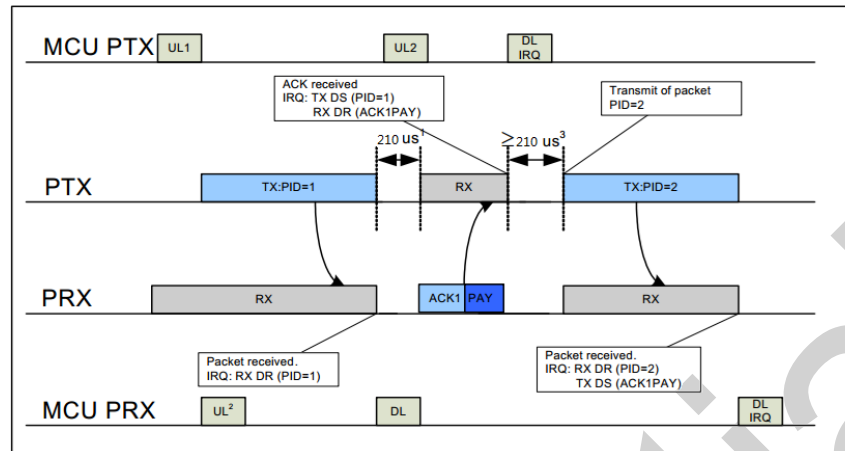
1 Radio Turn Around delay

Figure 7.16 TX/RX cycles with ACK and the according interrupts when the ACK packet fails

7.8.4 Single transaction with ACK payload packet

Figure 7.17 is a scenario of the basic auto acknowledgement with payload. After the packet is transmitted by the PTX and received by the PRX the ACK packet with payload is transmitted from the PRX to the PTX. The RX_DR IRQ is asserted after the packet is received

by the PRX, whereas on the PTX side the TX_DS IRQ is asserted when the ACK packet is received by the PTX. On the PRX side, the TX_DS IRQ for the ACK packet payload is asserted after a new packet from PTX is received. The position of the IRQ in Figure 7.17 shows where the MCU can respond to the interrupt.

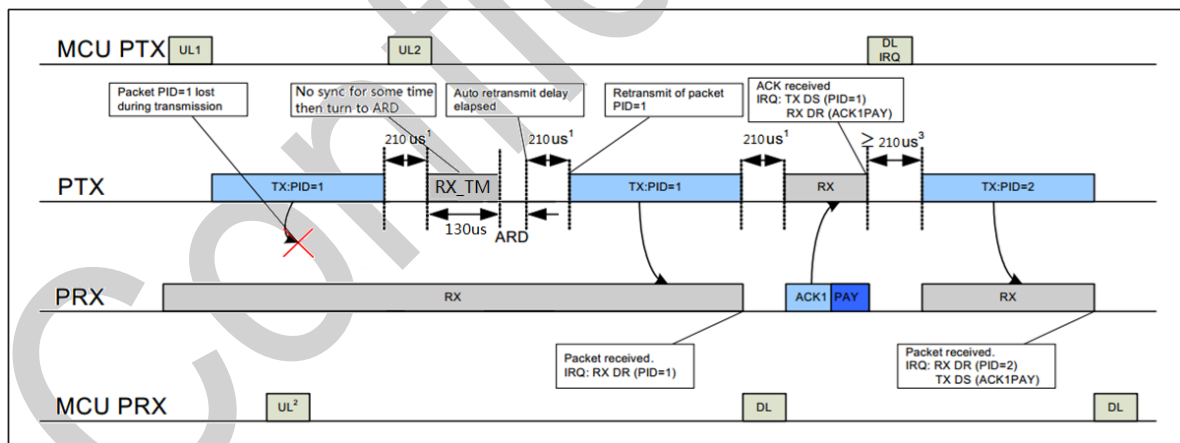


- 1 Radio Turn Around delay
- 2 Uploading payload for Ack Packet
- 3 Delay defined by MCU on PTX side, $\geq 210\mu s$

Figure 7.17 TX/RX cycles with ACK Payload and the according interrupts

7.8.5 Single transaction with ACK payload packet and lost packet

Figure 7.18 is a scenario where the first packet is lost and a retransmission is needed before the RX_DR IRQ on the PRX side is asserted. For the PTX both the TX_DS and RX_DR IRQ are asserted after the ACK packet is received. After the second packet (PID=2) is received on the PRX side both the RX_DR (PID=2) and TX_DS (ACK packet payload) IRQ are asserted.



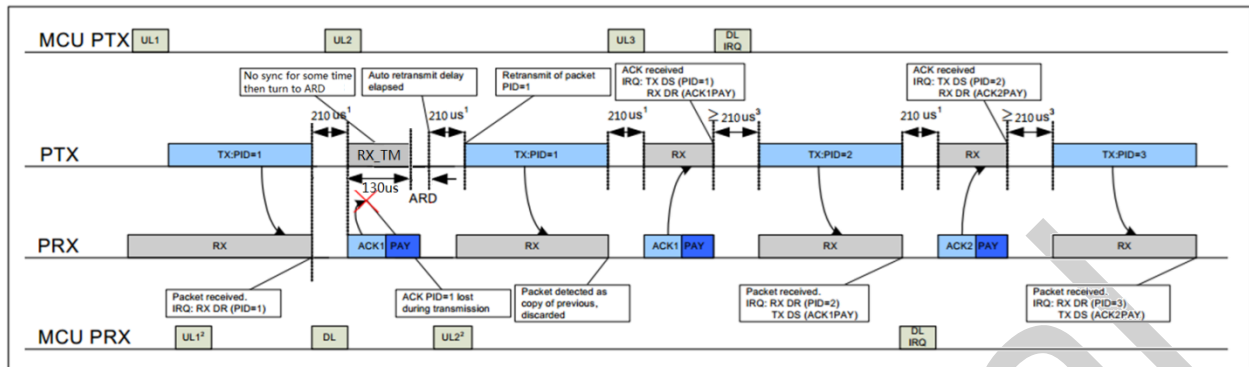
- 1 Radio Turn Around delay
- 2 Uploading payload for Ack Packet
- 3 Delay defined by MCU on PTX side, $\geq 210\mu s$

Figure 7.18 TX/RX cycles and the according interrupts when the packet transmission fails

7.8.6 Two transactions with ACK payload packet and the first ACK packet lost

Figure 7.19 the ACK packet is lost and a retransmission is needed before the TX_DS IRQ is asserted, but the RX_DR IRQ is asserted

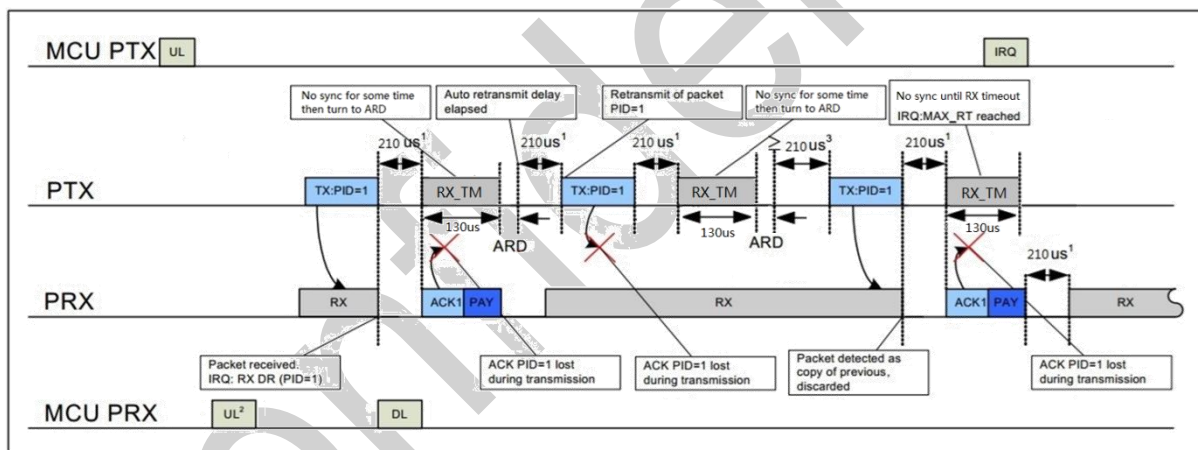
immediately. The retransmission of the packet (PID=1) results in a discarded packet. For the PTX both the TX_DS and RX_DR IRQ are asserted after the second transmission of ACK, which is received. After the second packet (PID=2) is received on the PRX both the RX_DR (PID=2) and TX_DS (ACK1PAY) IRQ is asserted. The callouts explains the different events and interrupts.



- 1 Radio Turn Around delay
- 2 Uploading payload for Ack Packet
- 3 Delay defined by MCU on PTX side, $\geq 210\mu s$

Figure 7.19 TX/RX cycles with ACK Payload and the according interrupts when the ACK packet fails

7.8.7 Two transactions where max retransmissions is reached



- 1 Radio Turn Around delay
- 2 Uploading payload for Ack Packet
- 3 Delay defined by MCU on PTX side, $\geq 210\mu s$

Figure 7.20 TX/RX cycles with ACK Payload and the according interrupts when the transmission fails. ARC is set to 2.

MAX_RT IRQ is asserted if the auto-retransmit counter (ARC_CNT) exceeds the programmed maximum limit (ARC). In Figure 7.20, the packet transmission ends with a MAX_RT IRQ. The payload in TX FIFO is NOT removed and the MCU decides the next step in the protocol. A toggle of the CE bit in the RCON register starts a new transmitting sequence of the same packet. The payload can be removed from the TX FIFO using the FLUSH_TX command.

8 Data and control interface

The data and control interface gives you access to all the features in the RF transceiver.

The data and control interface consists of the following five digital signals for 4 wire SPI.

- IRQ (this signal is active low and controlled by three mask-able interrupt sources)
- CSN (4 wire SPI signal)
- SCK (4 wire SPI signal)
- MOSI (4 wire SPI signal)
- MISO (4 wire SPI signal)

The data and control interface consists of the following four digital signals for 3 wire SPI

- IRQ (this signal is active low and controlled by three mask-able interrupt sources)
- CSN (3 wire SPI signal)
- SCK (3 wire SPI signal)
- SDA(3 wire SPI signal)

Using 1 byte SPI commands, you can activate the OM6220 data FIFO or the register map during all modes of operation.

8.1 Features

- Special SPI commands for quick access to the most frequently used features
- 0-8Mbps 3/4-wire SPI
- 8 bit command set
- Easily configurable register map
- Full one level FIFO for both TX and RX direction

8.2 Functional description

The SPI is a standard SPI with a maximum data rate of 10Mbps.

8.3 SPI operation

This section describes the SPI commands and timing.

8.3.1 SPI commands

The SPI commands are shown in Table 8.1. Every new command must be started by a high to low transition on CSN.

The STATUS register is serially shifted out on the MISO pin simultaneously to the SPI command word shifting to the MOSI pin.

The serial shifting SPI commands is in the following format:

<Command word: MSBit to LSBit (one byte)>

<Data bytes:LSByte to MSByte, MSBit in each byte first>

See Figure 8.1 and Figure 8.2 for timing information.

Command	Command word (binary)	#Data bytes	Operation
R_RESISTER	000A AAAA	1 to 5 LSByte first	Read command and status registers. AAAAA=5 bit register map address
W_RESISTER	001A AAAA	1 to 5 LSByte first	Write command and status registers. AAAAA=5 bit register map address Executable in power down or standby modes only.
R_TX_PAYLOAD	0110 0001	1 to 32 LSByte first	Read RX-payload: 1-32 bytes. A read operation always starts at byte 0. Payload is deleted from FIFO after it is read. Used in RX mode.
W_TX_PAYLOAD	1010 0000	1 to 32 LSByte first	Write TX-payload: 1 – 32 bytes. A write operation always starts at byte 0 used in TX payload.
FLUSH_TX	1110 0001	0	Flush TX FIFO, used in TX mode
FLUSH_RX	1110 0010	0	Flush RX FIFO, used in RX mode Should not be executed during transmission of acknowledge, that is, acknowledge package will not be completed
REUSE_TX_PL	1110 0011	0	Used for a PTX operation Reuse last transmitted payload. TX payload reuse is active until W_TX_PAYLOAD or FLUSH TX is executed. TX payload reuse must not be activated or deactivated during package transmission.
R_RX_PL_WID	0110 0000	1	Read RX payload width for the top R_RX_PAYLOAD in the RX FIFO. Note: Flush RX FIFO if the read value is larger than 32 bytes.
W_ACK_PAYLOAD	1010 1PPP	1 to 32 LSByte first	Used in RX mode. Write Payload to be transmitted together with ACK packet on PIPE PPP. (PPP valid in the range from 000 to 101). Maximum three ACK packet payloads can be pending. Payloads with same PPP are handled using first in – first out principle. Write payload: 1– 32 bytes. A write operation always starts at byte 0.
W_TX_PAYLOAD_NO_ACK	1011 0000	1 to 32 LSByte first	Used in TX mode. Disables AUTOACK on this specific packet specific packet.
RF_RSTN	10001011	0	Reset phy and rf state command
RTC_LOCK	10010000	0	Lock RTC Current value command
CE_HIGH	1101 0101	0	CE high command
CE_LOW	1101 0110	0	CE low command
CE_PULSE	11011001	0	CE pulse command

Table 8.1 Command set for the RF transceiver SPI

The W_REGISTER and R_REGISTER commands operate on single or multi-byte registers. When accessing multi-byte registers read or write to the MSBit of LSByte first. You can terminate the writing before all bytes in a multi-byte register are written, leaving the unwritten MSByte(s) unchanged. For example, the LSByte of RX_ADDR_P0 can be modified by writing only one byte to the RX_ADDR_P0 register. The content of the status register is always read to MISO after a high to low transition on CSN.

Note: The 2 bit pipe information in the STATUS register is updated during the IRQ high to low transition. The pipe information is unreliable if the STATUS register is read during an IRQ high to low transition.

8.3.2 SPI timing

SPI operation and timing is shown in Figure 8.1 to Figure 8.4. OM6220 must be in a standby, sleep or power down mode before writing to the configuration registers.

In Figure 8.1 to Figure 8.4 the following abbreviations are used:

Abbreviation	Description
Cn	SPI command bit
Dn	Data Bit (Note: LSByte to MSByte, MSBit in each byte first)

Table 8.2 Abbreviations used in Figure 8.1 to Figure 8.4

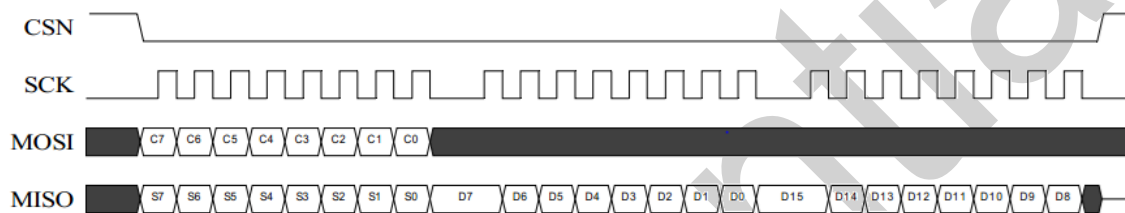


Figure 8.14 wire SPI read operation

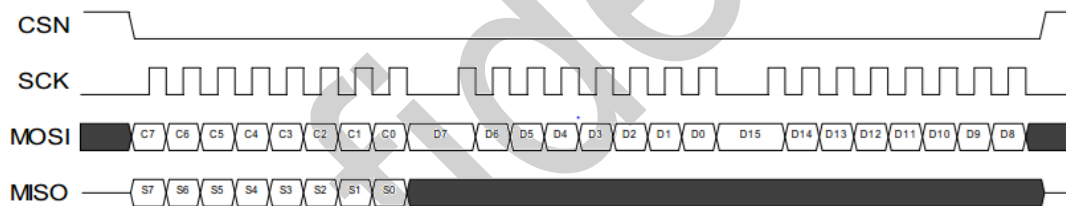


Figure 8.2 4 wire SPI write operation

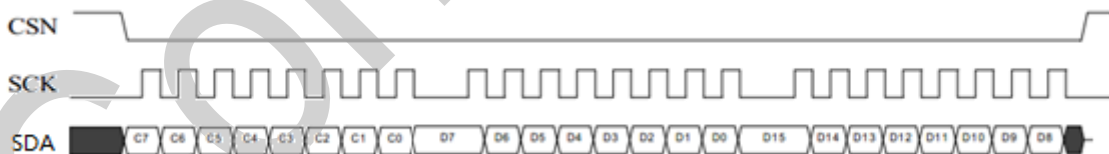


Figure 8.3 3 wire SPI read operation

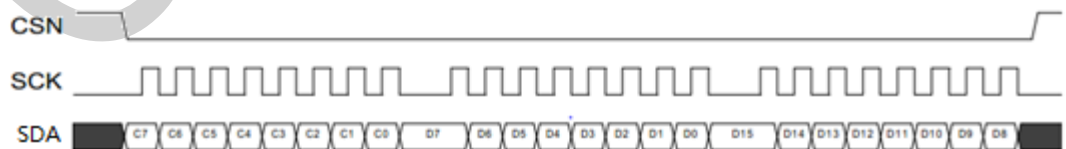


Figure 8.4 3 wire SPI write operation

8.4 Data FIFO

The data FIFOs store transmitted payloads (TX FIFO) or received payloads that are ready to be clocked out (RX FIFO). The FIFO is accessible in both PTX mode and PRX mode. The following FIFO is present in the RF transceiver:

- ◆ TX one level, 32 byte FIFO
- ◆ RX one level, 32 byte FIFO

Both FIFO have a controller and are accessible through the SPI by using dedicated SPI commands. A TX FIFO in PRX can store payloads for ACK packets to one different PTX devices. If the TX FIFO contains one payload to a pipe, payloads are handled using the first in – first out principle. The TX FIFO in a PRX is blocked if all pending payloads are addressed to pipes where the link to the PTX is lost. In this case, the MCU can flush the TX FIFO using the FLUSH_TX command.

The RX FIFO in PRX can contain payloads from up to one PTX device and a TX FIFO in PTX can have up to one payloads stored. You can write to the TX FIFO using these three commands; W_TX_PAYLOAD and W_TX_PAYLOAD_NO_ACK in PTX mode and W_ACK_PAYLOAD in PRX mode. All three commands provide access to the TX_PLD register.

The RX FIFO can be read by the command R_RX_PAYLOAD in PTX and PRX mode. This command provides access to the RX_PLD register.

The payload in TX FIFO in a PTX is not removed if the MAX_RT IRQ is asserted.

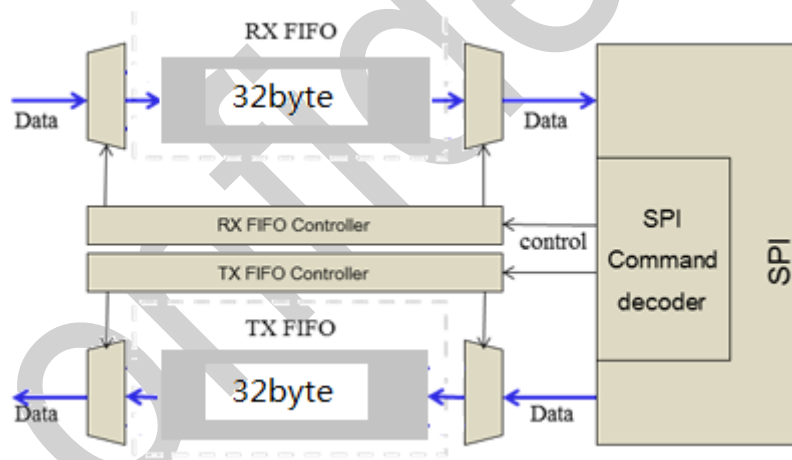


Figure 8.5 FIFO (RX and TX) block diagram

You can read if the TX and RX FIFO are full or empty in the FIFO_STATUS register.

8.5 Interrupt

The OM6220 has an active low interrupt (IRQ) pin. The IRQ pin is activated when TX_DS IRQ, RX_DR IRQ or MAX_RT IRQ are set high by the state machine in the STATUS register. The IRQ pin resets when MCU writes '1' to the IRQ source bit in the STATUS register. The IRQ mask in the CONFIG register is used to select the IRQ sources that are allowed to assert the IRQ pin. By setting one of the MASK bits high, the corresponding IRQ source is disabled. By default all IRQ sources are enabled.

Note: The 2bit pipe information in the STATUS register is updated during the IRQ high to low transition. The pipe information is unreliable if the STATUS register is read during an IRQ high to low transition.

Confidential

9 Register map

You can configure and control the radio by accessing the register map through the SPI.

Confidential

10 Glossary of Terms

Term	Description
ACK	Acknowledgement
ACS	Adjacent Channel Selectivity
AGC	Automatic Gain Control
ART	Auto Re-Transmit
CD	Carrier Detect
CE	Chip Enable
CLK	Clock
CRC	Cyclic Redundancy Check
CSN	Chip Select NOT
PE	Protocol engine
GFSK	Gaussian Frequency Shift Keying
IM	Intermodulation
IRQ	Interrupt Request
ISM	Industrial-Scientific-Medical
LNA	Low Noise Amplifier
LSB	Least Significant Bit
LSByte	Least Significant Byte
Mbps	Megabit per second
MCU	Microcontroller Unit
MISO	Master In Slave Out
MOSI	Master Out Slave In
MSB	Most Significant Bit
MSByte	Most Significant Byte
PCB	Printed Circuit Board
PID	Packet Identity Bits
PLD	Payload
PRX	Primary RX
PTX	Primary TX
PWR_DWN	Power Down
PWR_UP	Power Up
RoHS	Restriction of use of Certain Hazardous Substances
RPD	Received Power Detector
RX	Receive
RX_DR	Receive Data Ready
SPI	Serial Peripheral Interface
TX	Transmit
TX_DS	Transmit Data Sent

Table 10.1 Glossary of Terms

11 Package Information

OM6620 is a 10-pin 3x4.9x1.1mm MSOP10L package

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	-	-	1.10
A1	0.05	-	0.15
A2	0.75	0.85	0.95
A3	0.30	0.35	0.40
b	0.18	-	0.26
b1	0.17	0.20	0.23
c	0.15	-	0.19
c1	0.14	0.15	0.16
D	2.90	3.00	3.10
E	4.70	4.90	5.10
E1	2.90	3.00	3.10
e	0.50BSC		
L	0.40	-	0.70
L1	0.95REF		
θ	0	-	8°

Table 11.1 OM6220 MSOP10L Package Information

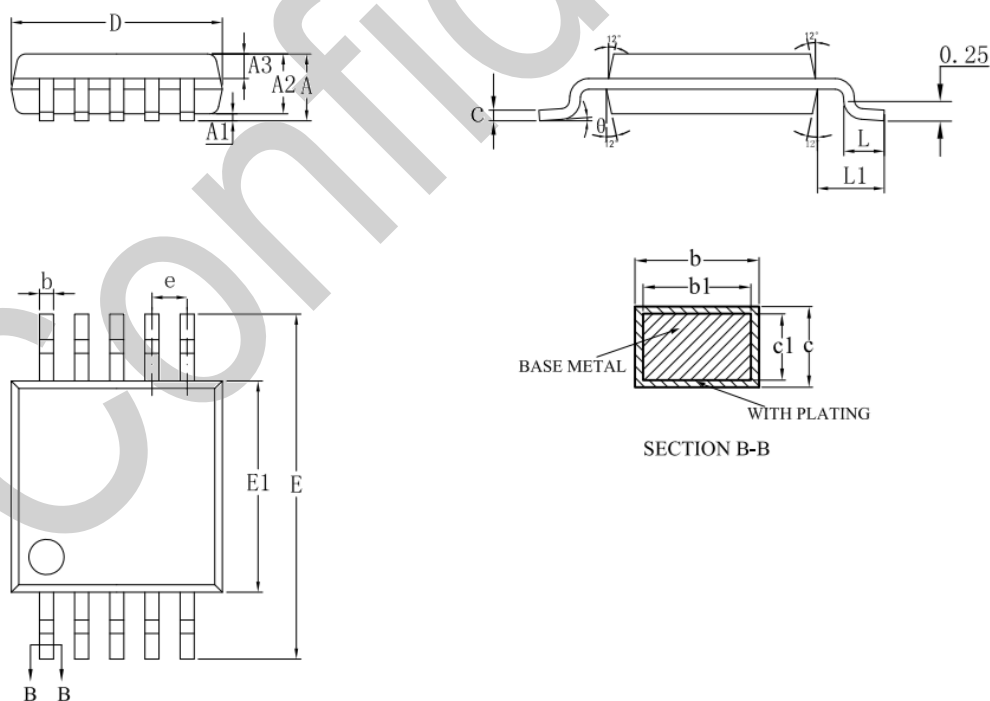


Figure 11.1 OM6220 MSOP10L Package Outlines

12 Ordering Information

OM6220's ordering information is as below.

Part No.	Type	Package Size	Packing	MOQ (PCS)	MPQ (PCS)	Status
OM6220ST111.TR	MSOP10L	3*4.9*1.1mm	Tape/Reel	3000	3000	CS

Table 12.1 OM6220 Ordering Information

Revision History

Revision	Date	Description
V0.1	Apr 2, 2021	1. OM6220 is 12 inch wafer version of HS6220; 2. Add some new feature for BLE
V0.2	Aug 26, 2021	Remove 5V tolerant IO
V0.3	Sep 22, 2022	Update BLE 5.0 feature
V0.4	Oct 9, 2022	Update BLE 5.1 feature
V0.5	Apr 19, 2023	1. Add MSOP10L package. 2. Delete SOP8 package. 3. Add package information. 4. Add ordering information.