SM2200 User Guide

Step by step create your own power line communication system with SM2200

VERSION 2.03

Table of Contents

1. Introduction

Before reading this document, please read SM2200 Datasheet to understand SM2200 basic structure and working principle. This document's intent is to help users to design a PLC system using SM2200.

The SM2200 is a next generation OFDMA (Orthogonal Frequency Division Multiple Access) power line communications transceiver. It contains a complete packet data modem which utilizes a simple PHY (Physical) layer protocol. This allows the development of proprietary point-to-point and star networks. When combined with an appropriate microcontroller (MCU), the SM2200 provides a cost-effective solution for data links and networks. Interface with the MCU is accomplished using a three/ four wire serial peripheral interface (SPI) connection and an interrupt request output which allows for the use of a variety of processors. The SPI port and interrupt request output is used for receive (RX) and transmit (TX) data transfer and control. The software and processor can be scaled to fit applications ranging from simple point-to-point to star networks.

Applications may include, but are not limited to, the following:

- Remote meter reading
- Smart Grid
- Smart Light Control
- Smart Home/Appliance
- Vehicle to Grid
- Alternative Energy
- M2M Data Gathering & Control
- Energy Management (i.e. lighting, HVAC)

The device contains 54 carriers grouped into 18 independent channels. Overall a data rate of up to 175kbps can be achieved with a more robust mode of 132kbps. The transceiver includes an on-chip 2.5V core power supply regulation.

The SM2200 employs orthogonal frequency division multiple access (OFDMA) as a modulation technique to increase reliability and throughput on sensor and control networks. OFDMA has been recently used in the multi user cellular wireless networking domain where it has produced significant gains. OFDMA is particularly suited for power line communications medium as it can be easily used to increase reliability and significantly increase throughput when small packets are being used. OFDMA also has superior abilities in avoiding narrow band interferers and frequency selectivity problems seen by ODFM and narrowband communications.

2. Hardware Design

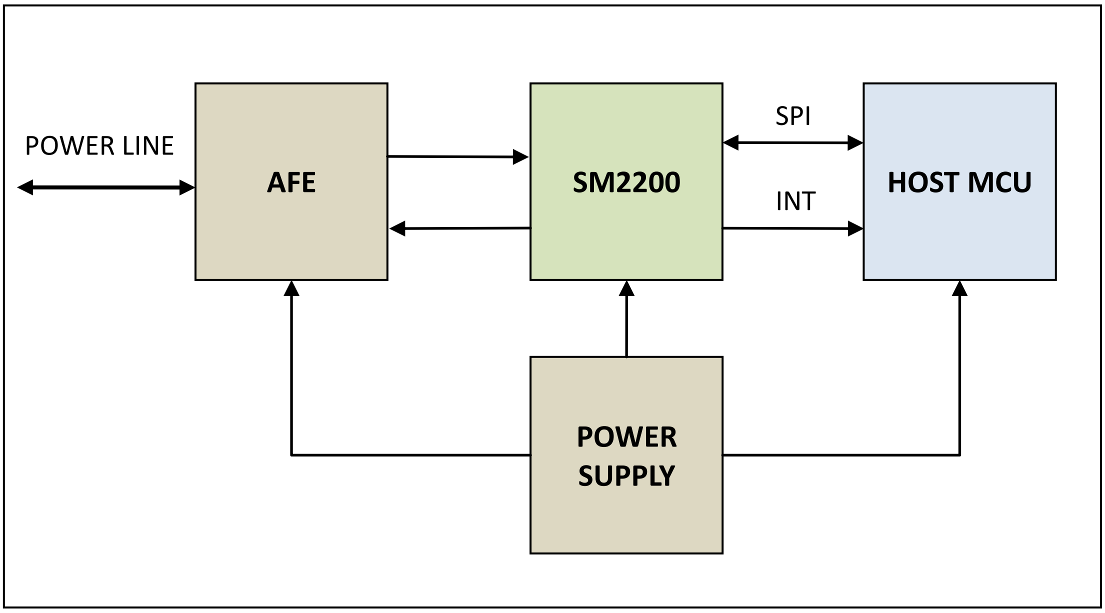


Figure 1 - A simplified SM2200 system block diagram

Figure 1 shows a simplified block diagram of how the SM2200 fits into a simple power line communications system. The system contains 4 basic blocks:

Host MCU: The SM2200 contains a modem that needs to be controlled by a microprocessor with an SPI port for operation. Most microcontrollers and microprocessors today contain some kind of SPI port making it an ideal communications interface. The network and application software (as required) reside on the host processor. The host can vary from a simple 8-bit device up to a sophisticated 64-bit processor depending on application requirements.

SM2200 Transceiver: The SM2200 provides modulation and demodulation of communications signals sent across the power line medium.

AFE: The AFE (Analog Front End) circuit includes a coupling circuit from the modem to the power line and filtering, isolation and protection from the power line voltage. It also provides amplification of the transmitter signal to be injected into the power line and a receiver circuit.

Power supply: The power supply requirements are a 3.3V supply to the SM2200 and 12V or 15V to the AFE circuit.

Please refer to www.semitechsemi.com for latest reference design.

2.1 Host MCU

The data exchange between the host MCU and SM2200 is through SPI interface. Therefore the chosen MCU must have a SPI interface. During operation the SM2200 processes large amount data, it is recommended to choose a MCU with more than 4KB RAM. There are plenty MCUs that meet above requirements, such as 8-bit AVR, PIC18Fxx, PIC24Fxx, Cortex-M0/M3 etc. The schematic below shows the SM2200 connected to an STM32 microcontroller.

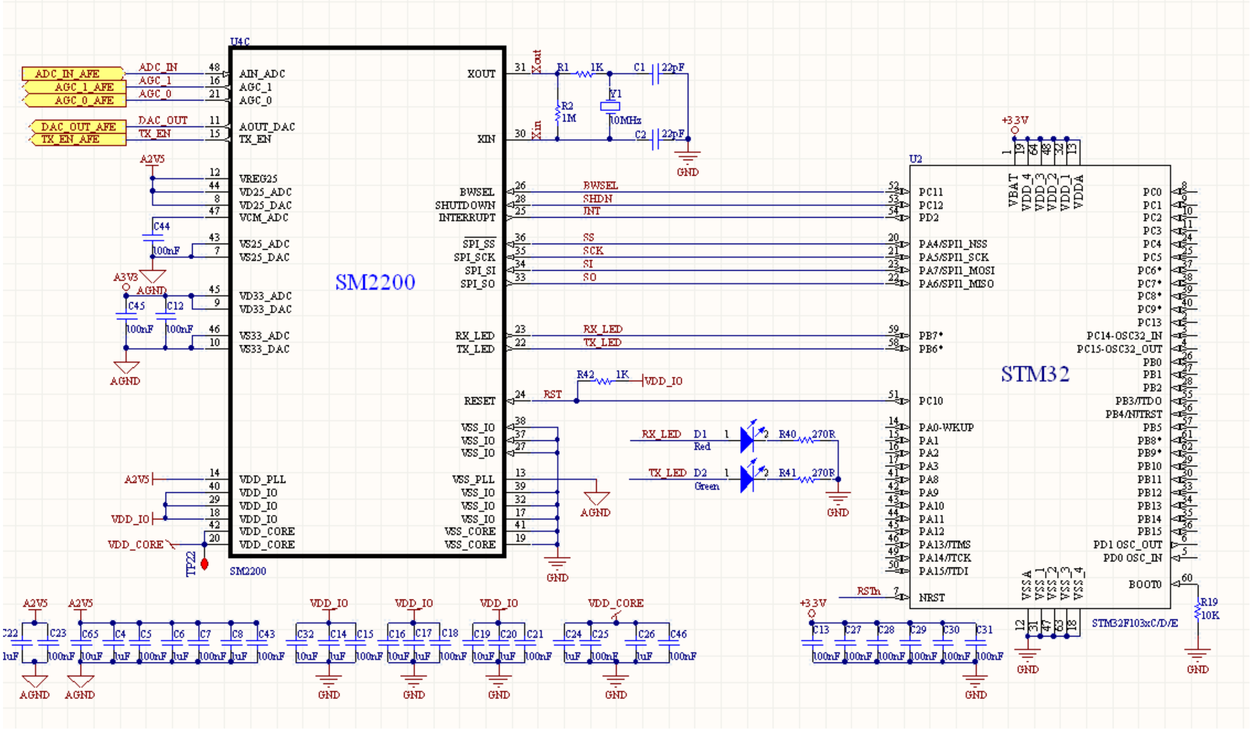


Figure 2 - The connection between STM32 and SM2200

It can be seen in the schematic above that the SM2200 is connected to the microcontroller’s SPI and other GPIOs. The SPI is the main communication interface between the SM2200 and the microcontroller and the GPIOs enable the microcontroller to have additional control over the SM2200.

The SM2200 SPI_SCK pin must remain in idle state while SS is de-asserted. For that reason it is highly recommended that SM2200 does not share an SPI master with other SPI slave devices.

The RST pin enables the microcontroller to reset the SM2200 anytime during operation. The interrupt pin when used and enabled will provide interrupt signals from the SM2200 to the microcontroller when the SM2200 needs servicing (i.e. receives a packet). SM2200 provides different interrupt events and can be enabled/disabled using the mask register. The SHDN pin can be connected directly to ground if not needed. The BWSEL pin can be connected directly to ground (for 500kHz band) or power (for 250kHz band) depending on the frequency that the application needs but connecting this pin to a GPIO gives the application and the user more flexibility for the cost of just one pin. The RX and TX LED pins are usually connected to the dedicated LEDs for visual indication of receiving and transmitting packets.

2.2 SM2200 Transceiver

Apart from the connection to a MCU, the SM2200 requires the support from some external circuitries in order to work properly. These external circuitries include power supply, clock, Tx and Rx status indication, the carrier signal power amplification and signal filtering etc.

2.2.1 Clock Circuitry

The clock circuitry of SM2200 is shown in Figure 3. It consists of a 10 MHz fundamental frequency crystal Y1, R1, R2 resistors and C1, C2 load capacitors. It is recommended to choose C1, C2 and a crystal with suitable resistance based on Y1 requirement. Field test results indicate that a high-precision crystal and appropriate load capacitance help improving the PLC sensitivity and accuracy. Note that the crystal Y1 must be a fundamental crystal oscillator. In the PCB layout, the track between the crystal and XIN and XOUT pin should be as short as possible.

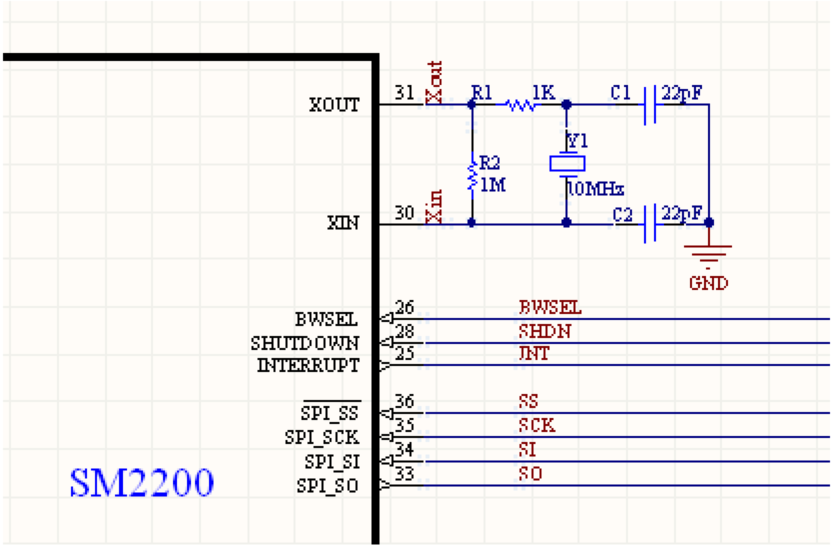


Figure 3 - SM2200 clock circuit

2.2.2 Rx and Tx Status LED

The SM2200 is equipped with TX_LED and RX_LED, the two output status indication pins which can directly drive LED lights to indicate the SM2200 working status. As shown in Figure 4 when the SM2200 sends packets to the power line, the LED indicator D2 is on; when the SM2200 receives packets from the power line, D1 is on. These two LEDs are very useful during debugging and fault diagnosis.

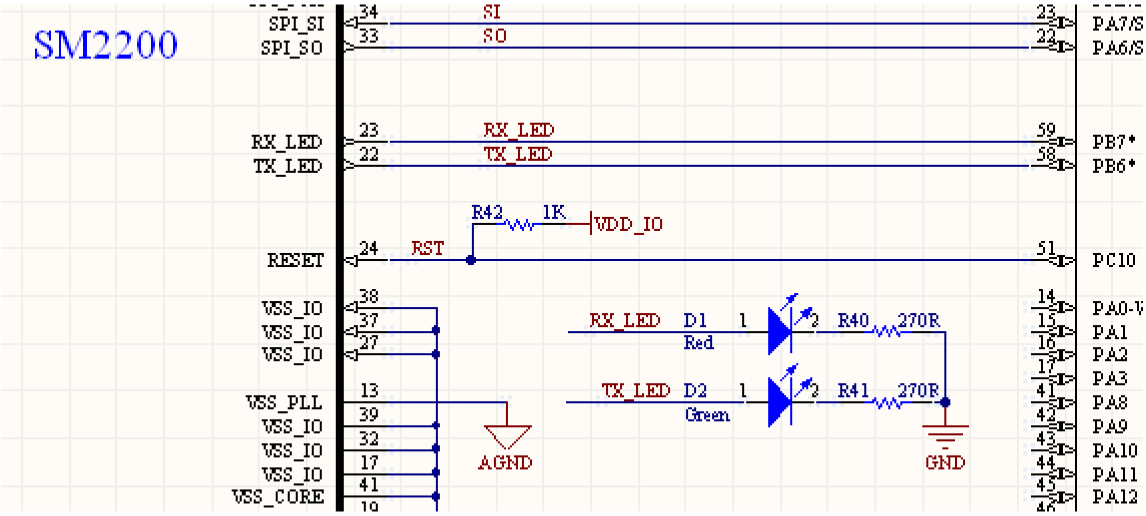


Figure 4 - LED indication circuitry

2.2.3 Transmit Signal Amplifier Circuitry

The signal output of the SM2200 at the DAC pin is not enough to be fed into the power line directly due to the noisy and low impedance nature of the power line. The signals needs to go through anti-aliasing filters and needs to be amplified before coupling to the power line. The on chip operational amplifier together with an external power amplifier are used both as filters and power amplification as shown in the schematic below.

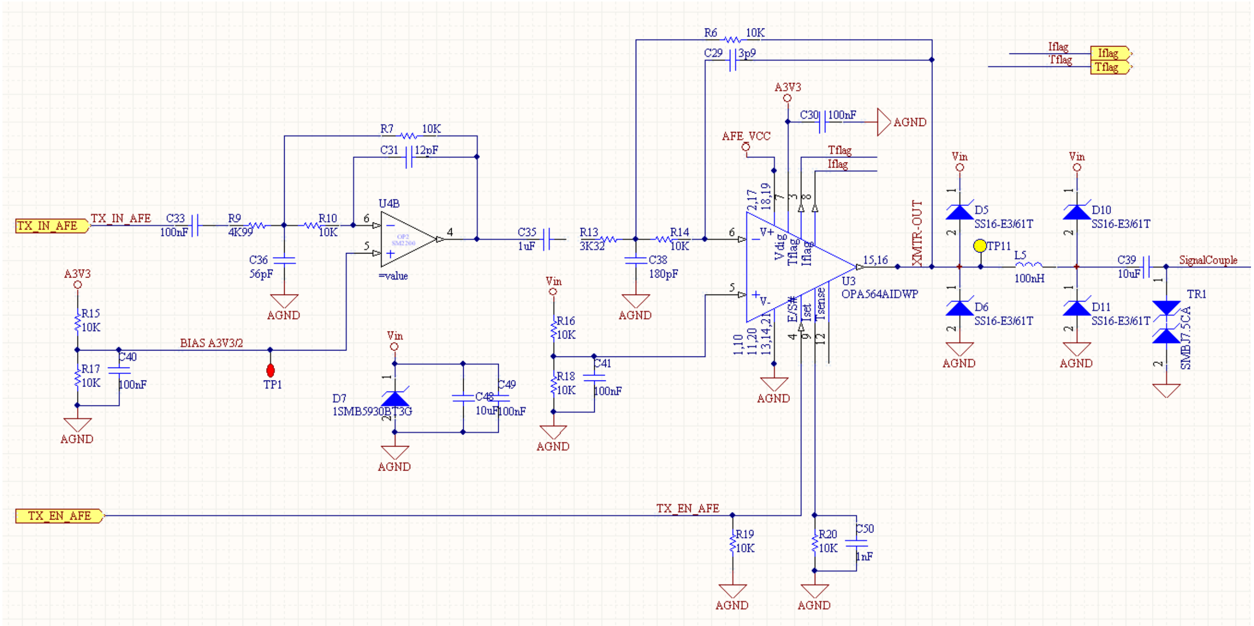


Figure 5 - Signal amplifier circuitry

The power amplifier is controlled by the TX Enable signal from the SM2200. This signal turns on the power amplifier during signal transmission and turns it off when not transmitting. It is essential that the power amplifier is turned off when the SM2200 is not transmitting to save power and to set the transmitter circuit’s output impedance high so that it does not affect the receiver circuit.

2.2.4 Receive Signal Filter Circuit

The SM2200 receive signal filter circuit is shown below, where SignalCouple is the carrier output signal from the coupling circuit. This signal has interference from a large amount of noise that is strong enough to cause saturation of the Op Amp. A passive LC band pass filter is used to block frequencies outside of the 50 kHz – 500 kHz frequency range. This band pass filter will have some signal attenuation on the desired frequency and is balanced out by amplifying the carrier signal using the on chip operational amplifier. Note that the on chip operational amplifier is also used as an active filter to further reject the noise frequencies.

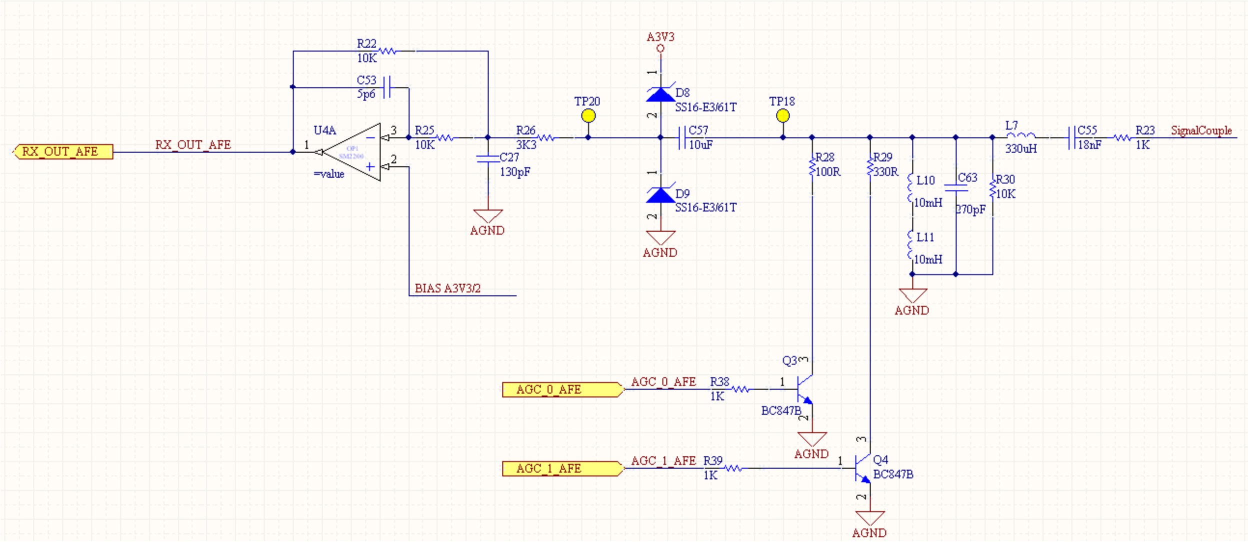


Figure 6 - Carrier signal filter circuit

The active filter using the on chip amplifier also biases the signal to mid rail before being fed to the input of the SM2200 ADC. When the carrier signal (SignalCouple) is too strong, there is a chance of saturating the on chip amplifier. To avoid this saturation, the SM2200 implements two pins for gain control; AGC0 and AGC1 as shown in the circuit above. When the pins are asserted, it turns on the transistors effectively reducing the signal coming into the amplifier. The settings of the two AGC pins are controlled in the AGC control register.

2.3 Signal Coupling Circuit

The coupling circuit is shown in the figure below.

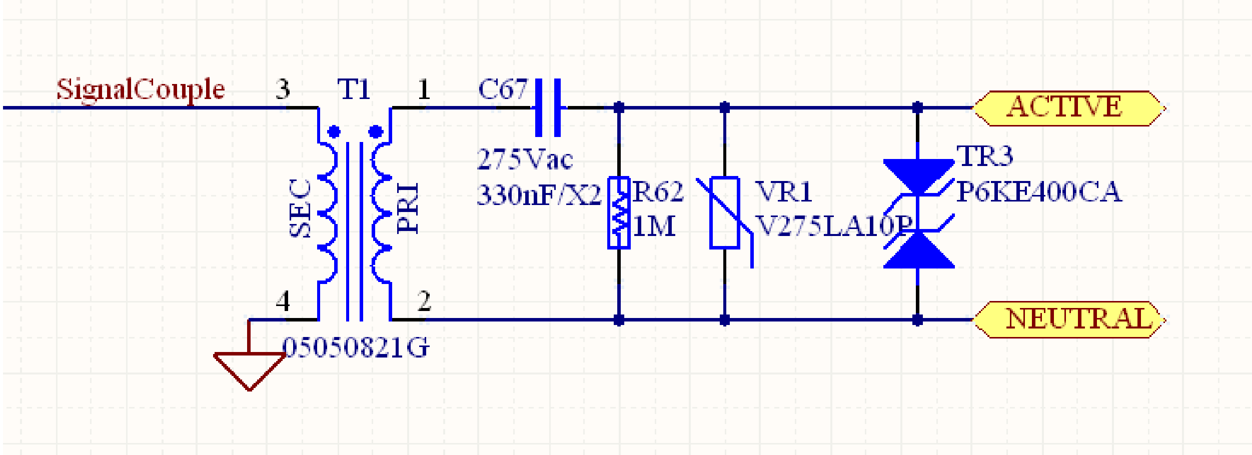


Figure 7 -- Coupling Circuit

where:

- **TR3:** transient voltage suppression (TVS) diode used to protect the coupler from voltage spikes induced on power line
- **VR1:** metal oxide varistor (MOV) used as additional protection circuit
- **C67:** mains rated X2 or X1 coupling capacitor
- **T1**: matching transformer
- **R62:** bleeder resistor

The coupling circuit has two functions, firstly is to couple the signal from the power amplifier output to both the power line and the receiver and filter circuit; and secondly to isolate the low voltage signal from the mains. The coupling circuit is crucial for a PLC system, because the input impedance varies in a large range. In the low voltage network, the rage of variation could be 0.1 – 1000 ohm. It is very difficult to couple the carrier signal to the power line with such low impedance. In order to design an efficient and reliable PLC system, besides amplifying the signal without distortion it is very important to design a suitable coupling circuit based on the power line network impedance characteristics. Besides efficient signal transmission, the coupling circuit should avoid to cause any signal distortion.

2.4 Power Supply

The SM2200 requires 3.3V DC power for its internal digital and analog circuitry. It is recommended to have a power supply with at least 200mA output current. Refer to Figure 2 - The connection between STM32 and SM2200 for the reference design of SM2200 power supply.

The power amplifier in the transmitter requires more power than that of the SM2200 and should be considered during the design. The frequency of packet transmission, expected load on the line and the maximum transmit voltage required all affects the power requirement.

3. SM2200 Software Design

This section is intended for the users who are familiar with STM32F103RD (or other MCU) hardware and have experience with its embedded C programming. The users should also see the SM2200 datasheet to understand the functions of its registers.

The simplest code for the SM2200 consists of 3 parts: SM2200 initialization, sending data through SM2200 and receiving data through SM2200.

3.1 SM2200 Frame Format

The figure below shows the frame format of the SM2200 signal. The Start of Frame (SoF) is a fixed 62 symbol long sequence. The End of Frame (EoF) is a also a fixed sequence 2 symbols long. The length of the Data field varies depending on the setting of the SM2200. Details of the contents of the data field are below.

- **Payload** – this is the data that the user wants to send. When addressing is enabled, the destination address is also part of the payload.
- **Sequence#** – when sequence numbering is enabled, a one byte sequence number is automatically inserted in the Data field.
- **CRC** – this is a two byte CRC that is always inserted in the data field
- **ECC** – when ECC is enabled, the ECC field is inserted in the data field

Start of Frame (SoF)	Data (Payload, Sequence#, CRC, ECC)	End of Frame (EoF)

The maximum size for the Data field is 127 bytes and the maximum payload size depends on the setting of the sequence numbering and ECC.

3.2 SPI Interface Programming

The data exchange between the MCU and the SM2200 is done through the SPI interfaces. The MCU reads and writes the SM2200 internal registers through the SPI interface.

3.2.1 SPI Interface Function

The host MCU directs the SM2200, checks its status, and reads/writes data to the device through the SPI port. The SM2200 transceiver operates as an SPI slave device only. A transaction between the host MCU and the SM2200 occur as multiple 8-bit bursts on the SPI.

The SPI signals are:

- Slave Select (SS) - A transaction on the SPI port can be framed by the active low SS input signal. Slave Select framing is enabled by default and can be disabled via the Physical Control Register (0x24). A transaction is a minimum of 3 SPI bursts and can extend to a greater number of bursts.
- SPI Clock (SCK) - The host drives the SCK input to the SM2200.
- Master Out/Slave In (MOSI) - Incoming data from the host is presented on the MOSI input.
- Master In/Slave Out (MISO) - The SM2200 presents data to the master on the MISO output.

A typical interconnection to a host MCU is shown in Figure 8.

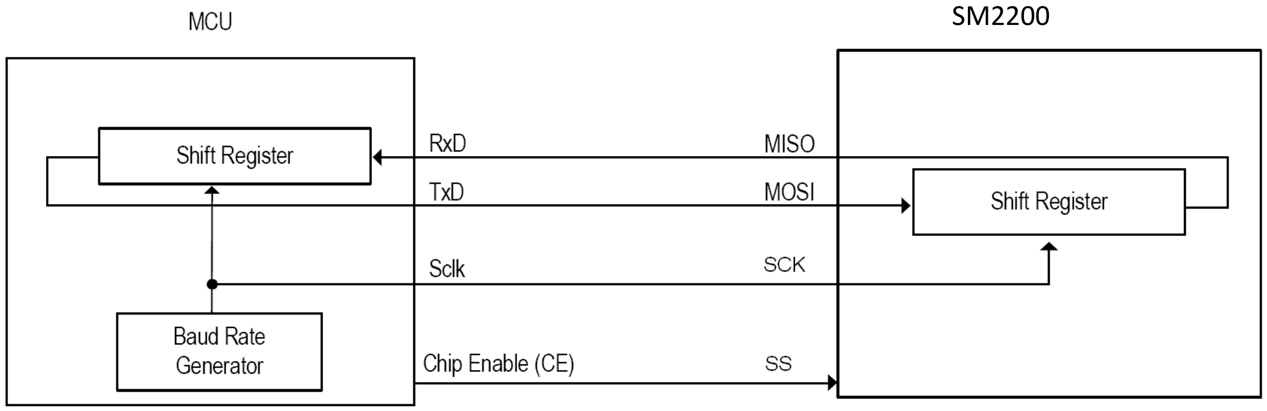


Figure 8 - SPI Interconnect with host MCU

The SM2200 SPI_SCK pin must remain in idle state while SS is de-asserted. For that reason it is highly recommended that SM2200 does not share an SPI master with other SPI slave devices.

3.2.2 SPI Transaction Formats

SPI transactions have three possible formats.

Format 1

R/W Bit	Address	High Data Byte	Low Data Byte
1 bit	7 bits	8 bits	8 bits

Format 2

R/W Bit	Address	High Data Byte	Mid Data Byte	Low Data Byte
1 bit	7 bits	8 bits	8 bits	8 bits

Format 3

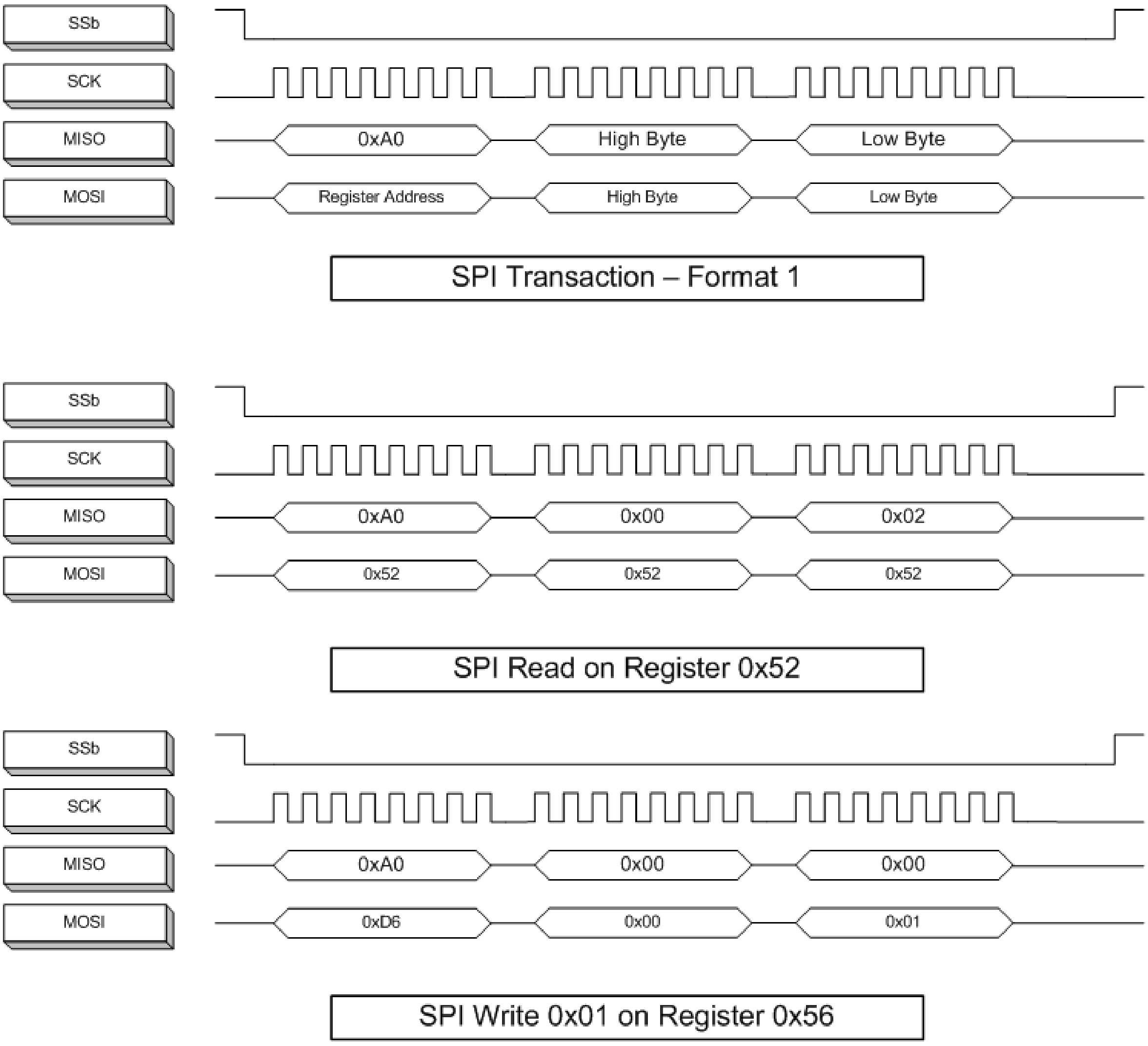
R/W Bit	Address	Data Byte 0	Data Byte 1	Data Byte N
1 bit	7 bits	8 bits	8 bits	8 bits

The R/W Bit tells the state machine if it needs to write to the register (Logic 1) or read from the register (Logic 0). The Address is a 7 bit field that tells the state machine which register to read and write from (Memory map is contained in the Memory map section). The two formats represent 16 and 24 bit SPI transfers. 16 bit transfers are used for most commands. Format 3 is only used for writing/reading to the TX/RX buffers. Format 3 is designed to minimize the overheads involved in sending and receiving packets.

3.2.3 SPI Transfer Format

The SPI slave contained in the SM2200 is designed to work with the widest range of SPI masters. For this reason, mode 0 is implemented as it is the most basic mode of SPI transfer and is very widely supported by microcontrollers.

The first byte read back on the SPI MISO line during the writing of the command byte should always be 0xA0. Subsequently, it depends on the register being read. If a write command is being issued instead of a read command, the byte read back on MISO for the last byte of the transaction should be 0xAA.

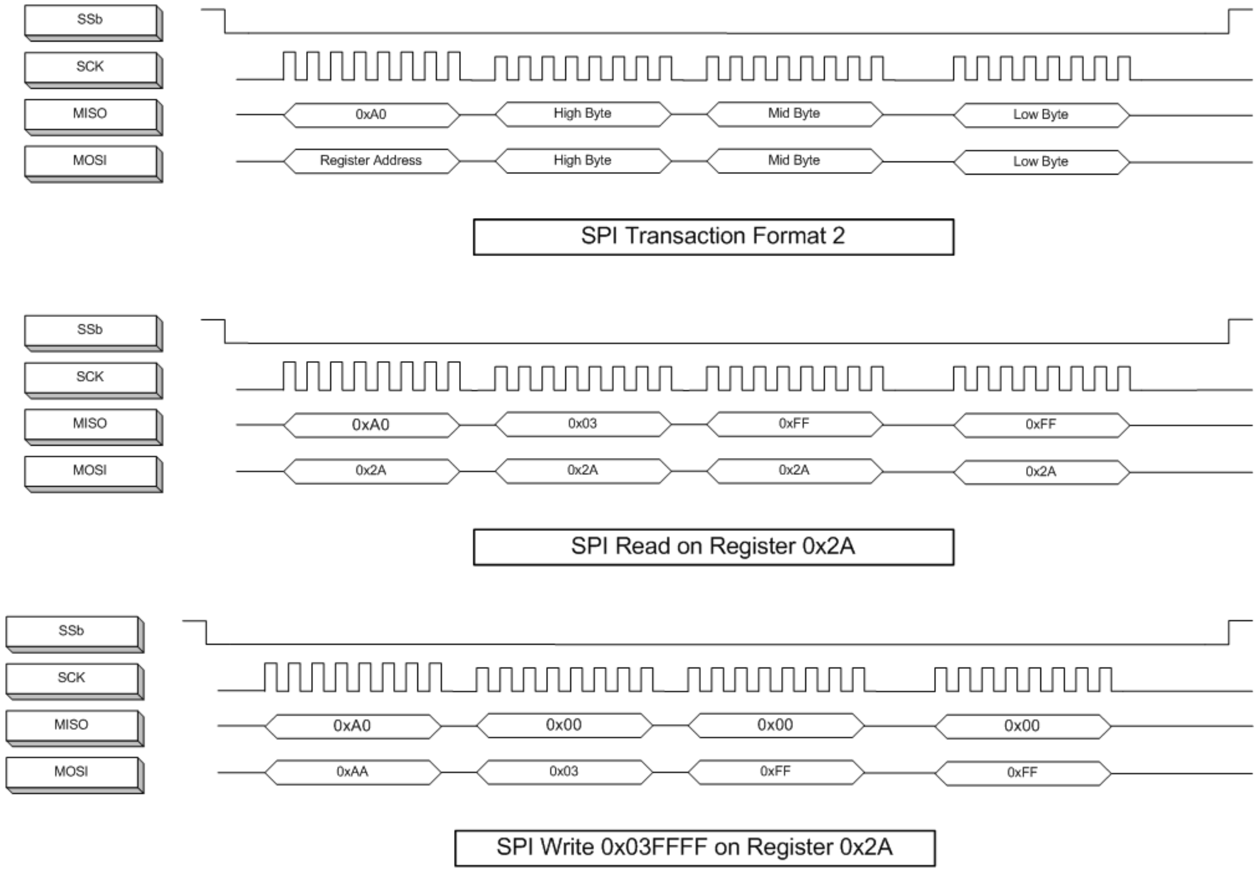


SPI Transaction – Format 1

SPI Read on Register 0x52

SPI Write 0x01 on Register 0x56

Figure 9 -- SPI Transaction Format 1

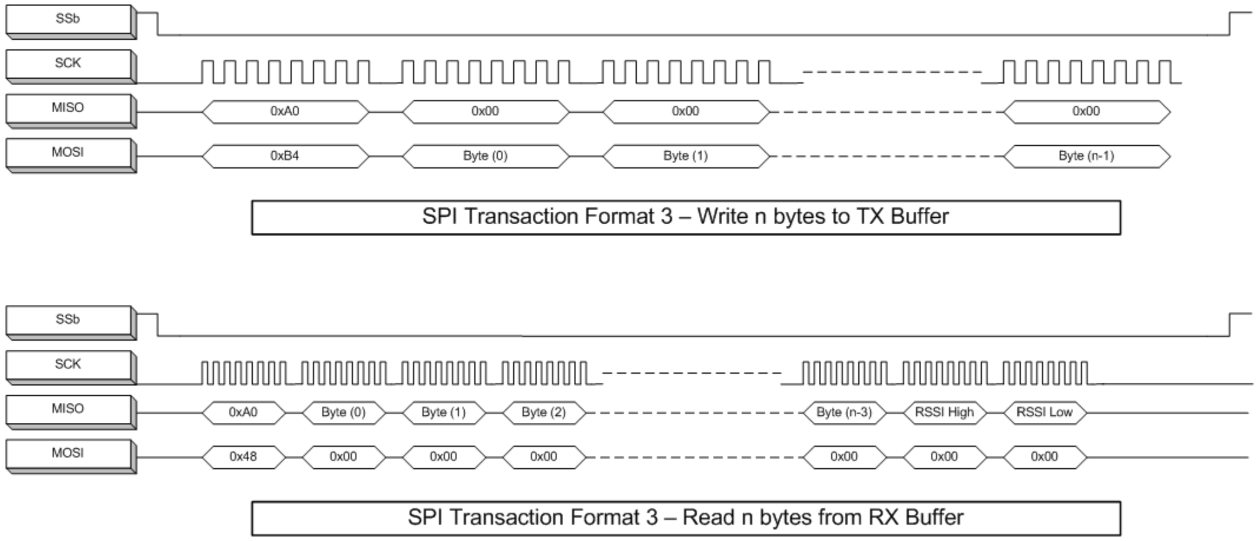


SPI Transaction Format 2

SPI Read on Register 0x2A

SPI Write 0x03FFFF on Register 0x2A

Figure 10 -- SPI Transaction Format 2



SPI Transaction Format 3 – Write n bytes to TX Buffer

SPI Transaction Format 3 – Read n bytes from RX Buffer

Figure 11 -- SPI Transaction Format 3

3.2.4 Slave Select Transaction Framing

It is strongly recommended that the SM2200 is operated in 4-Wire SPI Mode (SSb, SCK, MOSI & MISO) as the use of the Slave Select signal reduces to chance of loss of synchronization between the host microcontroller and the SM2200. Every time the Slave Select signal is de-asserted the SM2200 can resynchronize with the host. Slave Select framing is enabled by default and can be disabled via the Physical Control Register (0x24).

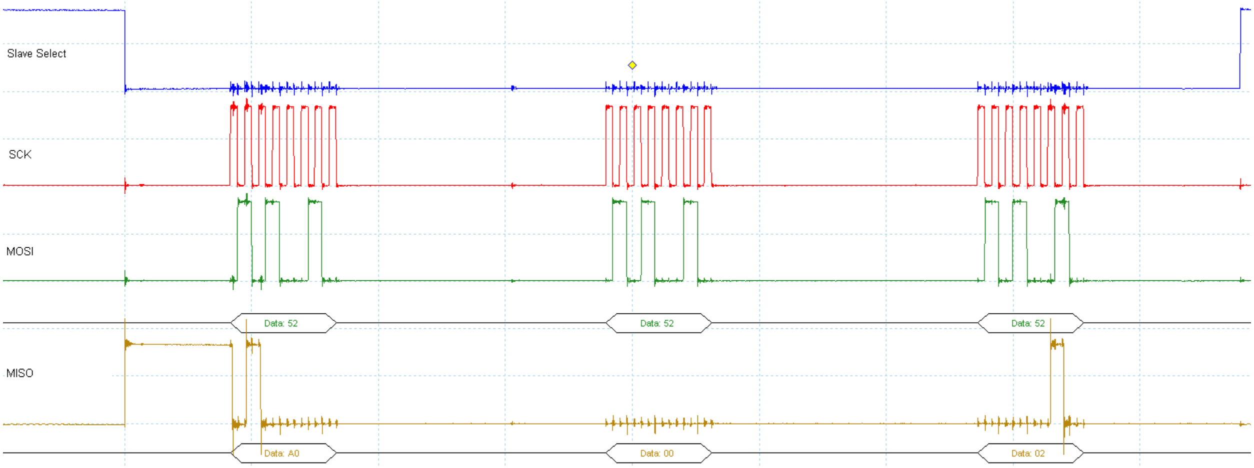


Figure 12 – Sample SPI timing with SS framing

The above example is a read to the interrupt event register (0x52). The SS (Slave Select) signal is driven low telling the SPI slave that a transaction is going to happen. At the start of the transaction, the command is given (0x52) meaning a read on the register (0x52). At this point, the SM2200 replies with 0xA0 and then the value of the register which is 0x0002. At the end of the transaction the slave select signal is returned to a high signifying the end of the transaction.

3.3 SPI Registers

3.3.1 SPI Register Map

Table 1 is an overview of all of the registers available to the SPI. All register addresses are in hex.

Table 1 - SPI Register Map

Address	Register	Index Range	SPI Format
0x20	SM2200_REG_INDEX_SEL		Format 1
0x22	SM2200_REG_MASTER_CARRIER_FREQ_i18	0 … 17	Format 1
0x24	SM2200_REG_PHY_CTRL		Format 1
0x26	SM2200_REG_PKTSIZE_i18	0 … 17	Format 1
0x28	SM2200_REG_BIU_THRESHOLD		Format 1
0x2A	SM2200_REG_CHANNEL_EN		Format 2
0x2C	SM2200_REG_BIU_STATUS		Format 2
0x32	SM2200_REG_CHANNEL_ATTEN_i18	0 … 17	Format 1
0x34	SM2200_REG_TX_BUFFER_i18	0 … 17	Format 3
0x36	SM2200_REG_SEND_PKTS		Format 2
0x3A	SM2200_REG_TX_BUFFER_MAP_i18	0 … 17	Format 1
0x3C	SM2200_REG_TX_LEVEL		Format 1
0x40	SM2200_REG_RX_STATUS		Format 2
0x42	SM2200_REG_CHANNEL_NOISE_i18	0 … 17	Format 1
0x48	SM2200_REG_RX_BUFFER_i18	0 … 17	Format 3
0x50	SM2200_REG_INT_MASK		Format 1
0x52	SM2200_REG_INT_EVENT		Format 1
0x56	SM2200_REG_TXCVR_CONFIG		Format 1
0x60	SM2200_REG_NODE_ADDR0		Format 1
0x62	SM2200_REG_NODE_ADDR1		Format 1
0x64	SM2200_REG_NODE_ADDR2		Format 1
0x66	SM2200_REG_NODE_ADDR3		Format 1
0x68	SM2200_REG_NODE_ADDR_MASK0		Format 1
0x6A	SM2200_REG_NODE_ADDR_MASK0		Format 1
0x6C	SM2200_REG_NODE_ADDR_MASK0		Format 1
0x6E	SM2200_REG_NODE_ADDR_MASK0		Format 1
0x70	SM2200_REG_AGC_CTRL		Format 1
0x72	SM2200_REG_AGC_LEVEL_ARRAY_i8	0 … 7	Format 1

Format 1: 2 Data Bytes
Format 2: 3 Data Bytes
Format 3: (2 + N) Data Bytes

3.3.2 Indexed Registers

The MCU writes SM2200 register through the SPI interface to run tasks of chip initialization, data transmission and receiving. The address of each register and SPI format to use to read or write are listed in the Table 1.

Registers with "_i*N*" appended to their name are indexed. For indexed registers, SM2200 needs to be told first what register index is being addressed via the index select register (SM2200_REG_INDEX_SEL). As an example, if channel noise needs to be read, the SM2200 needs to be told which of the 18 channel noise reading is to be read. If it is the noise reading on channel 5, then 0x05 is first **written** to SM2200_REG_INDEX_SEL register and then SM2200_REG_CHANNEL_NOISE_i18 is **read**.

3.3.3 0x20 – Index Select Register

Register Name	SM2200_REG_INDEX_SEL
Register Address	0x20
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	0	0	0	0	0	0	0	0
Mode	R	R	R	R	R	R	R	R
Reset	-	-	-	-	-	-	-	-

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	0	0	INDX[5]	INDX[4]	INDX[3]	INDX[2]	INDX[1]	INDX[0]
Mode	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	0	0	0	0	0	0

Register address 0x20 is the Index Select register. It is used as a pointer to select the index which is to be read or written.

The valid range for the index registers depends on what register is being accessed. Refer to the register map for the index range of the registers.

3.3.4 0x22 – Master Carrier Frequency Register

Register Name	SM2200_REG_MASTER_CARRIER_FREQ_i18
Register Address	0x22
SPI Format	Format 1 (16 bit)
Index Range	Channel[i]; i = 0…17

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	0	0	0	0	0	0	0	0
Mode	R	R	R	R	R	R	R	R
Reset[i]	-	-	-	-	-	-	-	-

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	0	FREQ[6]	FREQ[5]	FREQ [4]	FREQ [3]	FREQ [2]	FREQ [1]	FREQ [0]
Mode	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset[17]	-	1	0	0	1	1	0	0
Reset[16]	-	1	0	0	1	0	0	1
Reset[15]	-	1	0	0	0	1	1	0
Reset[14]	-	1	0	0	0	0	1	1
Reset[13]	-	1	0	0	0	0	0	0
Reset[12]	-	0	1	1	1	1	0	1
Reset[11]	-	0	1	1	1	0	1	0
Reset[10]	-	0	1	1	0	1	1	1
Reset[9]	-	0	1	1	0	1	0	0
Reset[8]	-	0	1	1	0	0	0	1
Reset[7]	-	0	1	0	1	1	1	0
Reset[6]	-	0	1	0	1	0	1	1
Reset[5]	-	0	1	0	1	0	0	0
Reset[4]	-	0	1	0	0	1	0	1
Reset[3]	-	0	1	0	0	0	1	0
Reset[2]	-	0	0	1	1	1	1	1
Reset[1]	-	0	0	1	1	1	0	0
Reset[0]	-	0	0	1	1	0	0	1

Carriers are arranged into 18 groups of 3 carriers. This means that there are a total of 54 carriers. The group of carriers is called a cluster and represents a channel. The Master carrier is the middle frequency of the 3 carriers. Figure 13 demonstrates the relationship between the lower, master and upper carriers in a cluster.

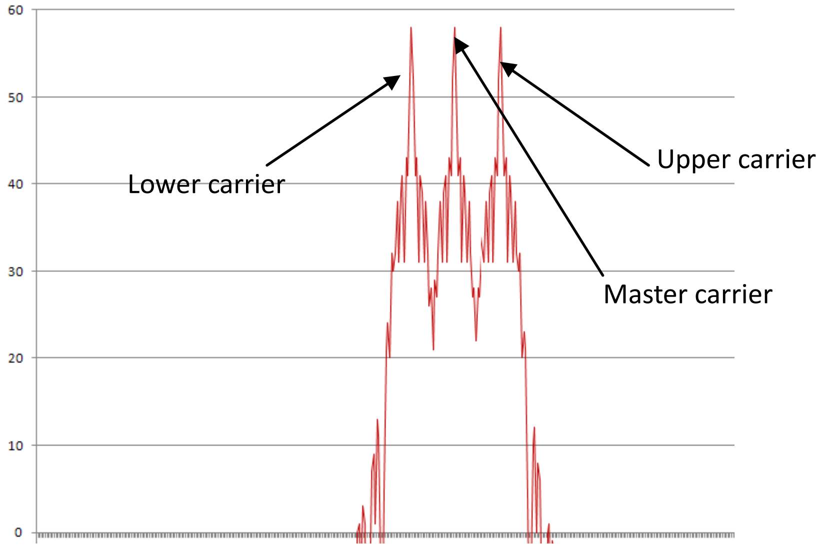


Figure 13 - Carrier cluster containing 3 discrete carriers

The master carrier can be any of the frequencies (except 0) listed in Table 2. The two side carriers are always the frequency above and the frequency below. Each master carrier must be spaced apart at least 3 frequencies apart to allow the operation of the side carriers. For example if the carrier frequency 26 (131.8 kHz) is selected then the lower carrier will be frequency 25 (126.9 kHz) and the upper frequency 27 (136.7 kHz). The spectrum of this example is shown in Figure 14.

The master carrier contains more information that the side carriers and cannot be turned off.

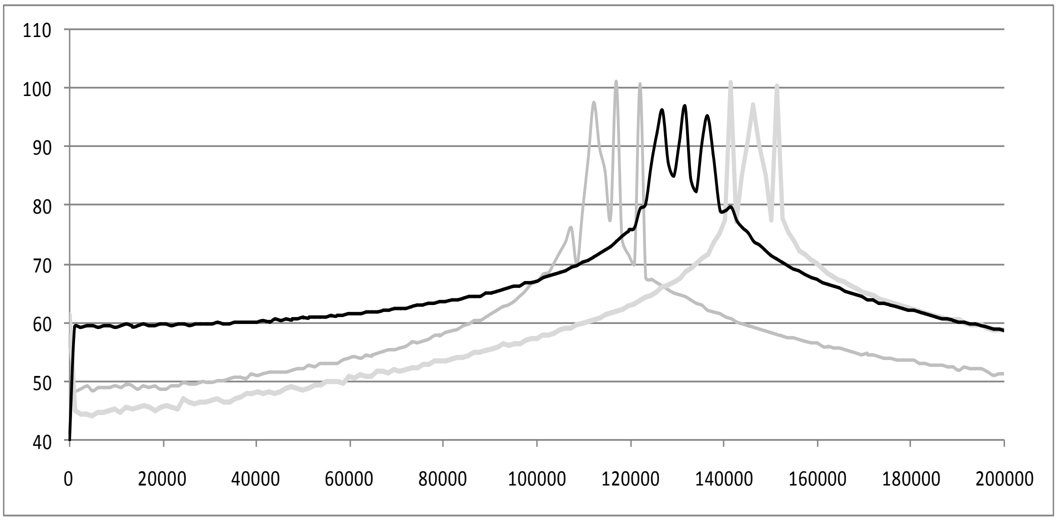


Figure 14 - Spectrum of carrier triplet for the example of 131.8kHz master carrier

The two side carriers can be switched on and off depending on the throughput and reliability needed. If there is only one channel operating then in this mode the chip acts like a BPSK narrowband transceiver. Figure 15 shows the output spectrum for 131.8 kHz master carrier with the side carrier turned off. If the transceiver is operating in this master only mode then they only need to be one frequency apart. The master carrier contains more information than the side carriers and cannot be turned off unless the channel is not being used.

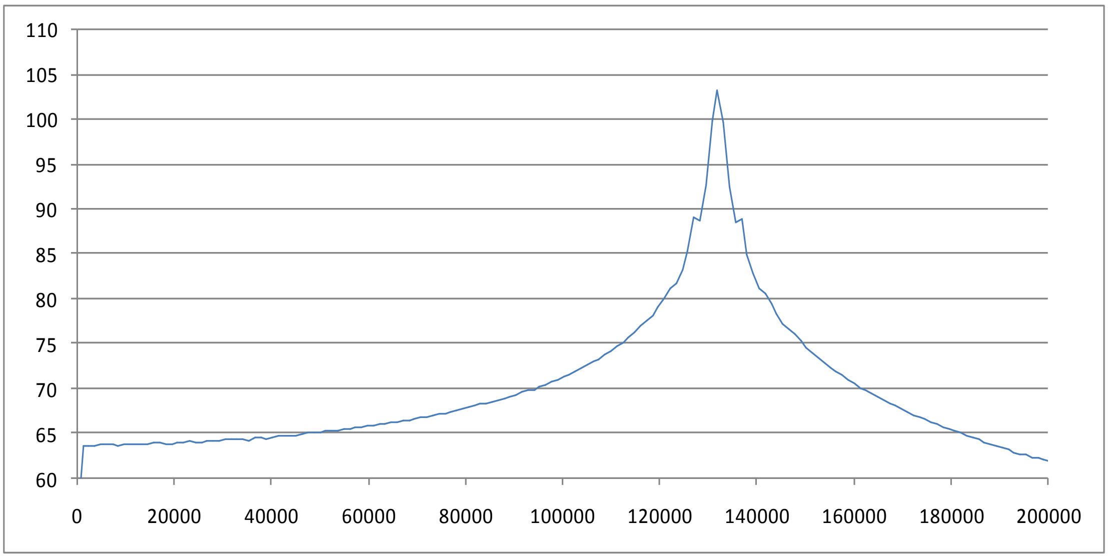


Figure 15 - Spectrum for the example of 131.8kHz master carrier with side carriers turned off

Channels of SM2200 do not need to be arranged in a contiguous block like that of an OFDM system. The 18 channels can be anywhere in the frequency range of 9kHz to 500kHz. Figure 16 shows an arrangement where there are two breaks in the spectrum. This arrangement is good for avoiding parts of the spectrum that may be noisy.

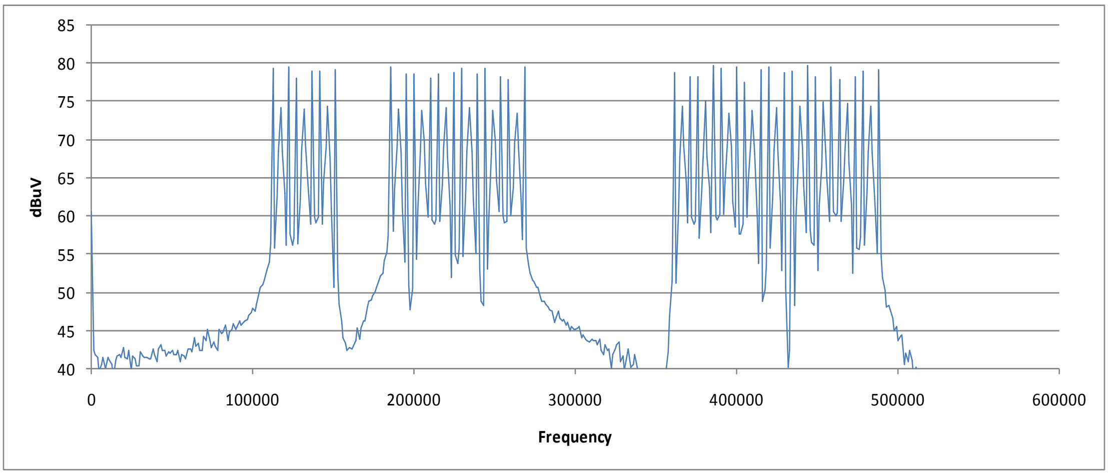


Figure 16 - Non contiguous carrier arrangement

The Master Carrier Frequency register controls the frequency of the 18 master carriers. Firstly the channel is selected with the Index Select register. Then the Master Carrier Frequency register will change the frequency of that channel. It is a 7 bit register FREQ [6:0] having the range between 0 and 101. The table below shows the possible frequencies at which the carriers can operate.

Table 2 - Possible carrier frequencies

Register Value	Frequency Hz	Register Value	Frequency Hz	Register Value	Frequency Hz
0	4882.82	**34**	170898.44	**68**	336914.06
1	9765.63	**35**	175781.25	**69**	341796.88
2	14648.44	**36**	180664.06	**70**	346679.69
3	19531.25	**37**	185546.88	**71**	351562.5
4	24414.06	**38**	190429.69	**72**	356445.31
5	29296.88	**39**	195312.5	**73**	361328.13
6	34179.69	**40**	200195.31	**74**	366210.94
7	39062.5	**41**	205078.13	**75**	371093.75
8	43945.31	**42**	209960.94	**76**	375976.56
9	48828.13	**43**	214843.75	**77**	380859.38
10	53710.94	**44**	219726.56	**78**	385742.19
11	58593.75	**45**	224609.38	**79**	390625
12	63476.56	**46**	229492.19	**80**	395507.81
13	68359.38	**47**	234375	**81**	400390.63
14	73242.19	**48**	239257.81	**82**	405273.44
15	78125	**49**	244140.63	**83**	410156.25
16	83007.81	**50**	249023.44	**84**	415039.06
17	87890.63	**51**	253906.25	**85**	419921.88
18	92773.44	**52**	258789.06	**86**	424804.69
19	97656.25	**53**	263671.88	**87**	429687.5
20	102539.06	**54**	268554.69	**88**	434570.31
21	107421.88	**55**	273437.5	**89**	439453.13
22	112304.69	**56**	278320.31	**90**	444335.94
23	117187.5	**57**	283203.13	**91**	449218.75
24	122070.31	**58**	288085.94	**92**	454101.56
25	126953.13	**59**	292968.75	**93**	458984.38
26	131835.94	**60**	297851.56	**94**	463867.19
27	136718.75	**61**	302734.38	**95**	468750
28	141601.56	**62**	307617.19	**96**	473632.81
29	146484.38	**63**	312500	**97**	478515.63
30	151367.19	**64**	317382.81	**98**	483398.44
31	156250	**65**	322265.63	**99**	488281.25
32	161132.81	**66**	327148.44	**100**	493164.06
33	166015.63	**67**	332031.25	**101**	498046.88

3.3.5 0x24 - Physical Control Register

Register Name	SM2200_REG_PHY_CTRL
Register Address	0x24
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	Reserved	Reserved	Reserved	Reserved	Reserved	BIUQUAL	AGCEN	SSFRAME
Mode	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	1	1	0	1	1	1	1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CPMODE	-	-	RXEN	OPPD	ILOCKENb	LBEN	PHYENb
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	0	1	0	0	1

The physical control register contains various bits that effect the operation of the physical layer transceiver.

PHYENb – Physical Enable (active low)
After reset (or power on) the PHY is disabled and must be enabled by clearing this bit before communications is possible. This bit is toggled via the SPI interface during normal operation (i.e. when the POR and RESET pin are not asserted). It does not affect the register settings so it can be used as a standby mode.

The PHY takes up to 500µs before being fully functional after clearing the PHYENb bit. It is possible to enable the PHY at the beginning of the initialization sequence.

LBEN – Internal Loopback Enable
This bit enables an internal loopback mode. This can be used to determine if the transceiver is operating correctly. It can also be used for developing code on a single transceiver as you will receive what you have sent.

ILOCKENb – Interlock Enable (active low)
In the default mode the Interlock is enabled so that the receiver is stopped from receiving what it has sent. The table below shows different configurations when toggling LBEN and ILOCKENb. The figure below illustrates the two modes of loopback.

Table 3 - Configuration of Loopback Modes

ILOCKENb	**LBEN**	**Mode**	
0	0	Normal operation.	Receiver does not receive what it has sent but receives from external sources.
0	1	Null mode.	This is an invalid mode as information will not be coming in from external sources but the internal sources are also interlocked
1	0	Local/Analog Loopback	The transmit signal will be transmitted through the analog front end where it will fed back into the receiver through the shared medium. Use small TXV value or enable AGC.
1	1	Internal/Digital Loopback	The transmit signal is fed back into the receiver switching out the ADC signal.

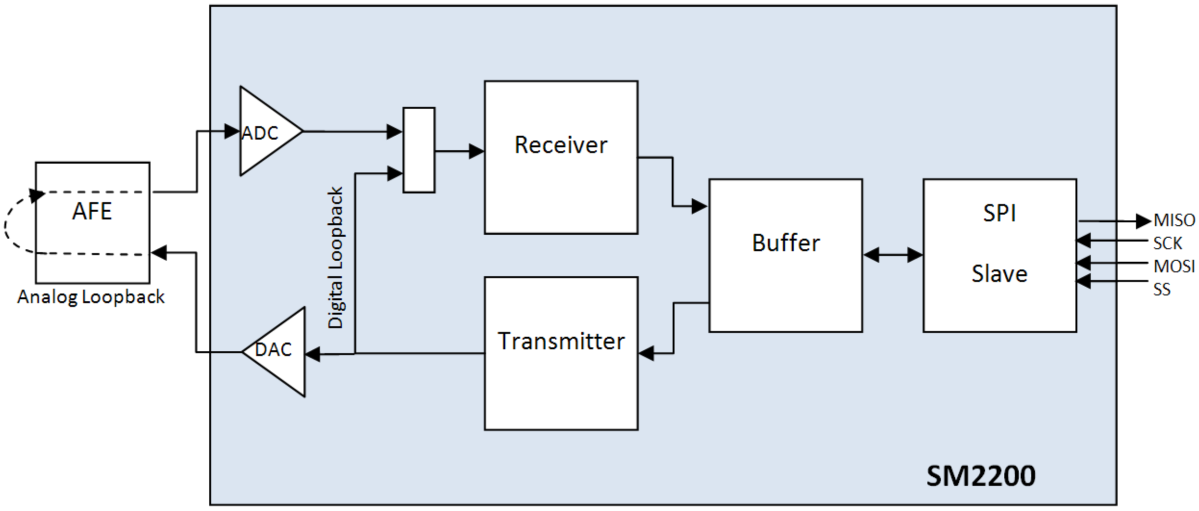


Figure 17 - Loopback Modes

OPPD – OpAmp Power Down

The SM2200 has two on-chip operational amplifiers (op-amps). The OPPD bit allows the user to disable these op-amps to reduce power consumption. This is set by default and should be cleared during the initialization process when these op-amps are used.

RXEN – Receive Enable

When set, this bit enables the receive function of the SM2200. This bit is cleared by default and should be set during the initialization process for normal operation.

CPMODE – Short Cyclic Prefix Mode

When this bit is set, a short cyclic prefix sample is used. Using short cyclic prefix results in a higher data rate at the expense of lesser robust communication. Therefore, it is highly recommended to use CP mode 0 (default) for a reliable power line communication.

SSFRAME – Slave Select Framing

When set this bit allows slave select framing as in Section 3.2.4. This prevents the SPI master and SPI slave from getting out of sync.

AGCEN – AGC Enable

When set this will enable the AGC pins on the SM2200. Note that when enabled, need to configure the other AGC settings on the AGC Control Register.

BIUQUAL – Band In Use Qualification

When BIU qualification is enabled the SM2200 not only considers the energy when determining if the channel is in use, but also the "quality" of the signal. (This is discussed in more detail in the BIU section).

3.3.6 0x26 - Packet Size Register

Register Name	SM2200_REG_PKTSIZE_i18
Register Address	0x26
SPI Format	Format 1 (16 bit)
Index Range	BUFFER[i]; i = 0…17

The packet size register is shared between the transmitter and receiver. It is an indexed register pointing to any of the 18 receive or transmit buffers. On read (receive) operation, it contains the size of the received packet in the receive buffer and whether the packet is a new packet or not. During write (transmit) operation, it contains the size of the packet that will be transmitted from the transmit buffers. These are explained in more detail below.

Read Operation

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	0	0	0	0	0	0	0	NEWPKT
Mode	R	R	R	R	R	R	R	R
Reset[i]	-	-	-	-	-	-	-	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	0	RXSIZE [6]	RXSIZE [5]	RXSIZE [4]	RXSIZE [3]	RXSIZE [2]	RXSIZE [1]	RXSIZE[0]
Mode	R	R	R	R	R	R	R	R
Reset[i]	-	0	0	0	0	0	0	0

RXSIZE[6:0] – Packet Size

This contains the size of the received packet in the receive buffer selected. Each receive buffer has a size of 128 bytes. In addition to the payload size, the received packet size always includes the RSSI of the packet (2 bytes) and depending on the setting of the transceiver configuration register (0x56), the sequence number and address.

NEWPKT – New Packet

This bit indicates if the associated RX buffer contains new data (set ‘1’) or not (cleared ‘0’).

The associated receive buffers are “locked” when a new packet is received as indicated by bit 8 of the Packet Size Register. This means that no new packets can be written to the receive buffer. Reading the Packet Size Register clears this bit and “unlocks” the receive buffer so that a new packet can be written to it. If the host processor does not read all of the data in the receive buffer before a new packet arrives, the data in the receive buffer will be overwritten by the new incoming packet. One way to avoid this is shown in the Packet Reception section 3.4.4.

Write Operation

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	-	-	-	-	-	-	PT[1]	PT[0]
Mode	-	-	-	-	-	-	W	W
Reset[i]	-	-	-	-	-	-	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	-	TXSIZE [6]	TXSIZE [5]	TXSIZE [4]	TXSIZE [3]	TXSIZE [2]	TXSIZE [1]	TXSIZE[0]
Mode	-	W	W	W	W	W	W	W
Reset[i]	-	0	0	0	0	0	0	0

TXSIZE [6:0] – Packet Size

This contains the size of the data to be written to the TX buffer and transmitted. Although the buffer size is 128 bytes, the maximum payload size varies depending on the setting of the transceiver configuration register (0x56); whether ECC and/or sequence numbering is on.

Table 4 -- Supported Maximum Payload Size

ECC	**Sequence Number**	**Max Payload Size (bytes)**
OFF	OFF	125
OFF	ON	124
ON	OFF	89
ON	ON	88

Note that transmitting payloads more than the supported maximum size could have unpredictable effects on the SM2200 transactions.

PT[1:0] – Packet Type

Packet type refers to the mode of transmission to be used as shown in the table below.

Table 5 - Configuration of packet type

Value PT[1:0]	**Master Carrier**	**Side Carrier**	**Description**
0	QPSK	QPSK	Full rate mode QPSK on all carriers in a channel.
1	BPSK	BPSK	Half rate. BPSK on all carriers in a channel
2	BPSK	OFF	Master carrier only using BPSK modulation. All other carriers in a channel are off.

3.3.7 0x28 - BIU Upper Threshold Register

Register Name	SM2200_REG_BIU_THRESHOLD
Register Address	0x28
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	UTHRESH[15]	UTHRESH[]	UTHRESH[]	UTHRESH[]	UTHRESH[]	UTHRESH[]	UTHRESH[]	UTHRESH[]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	UTHRESH[]	UTHRESH[]	UTHRESH[]	UTHRESH[]	UTHRESH[]	UTHRESH[]	UTHRESH[]	UTHRESH[0]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	0	0	0	0

This is a 16 bit register which is used to compare the amount of energy measured in a channel. If the energy exceeds the upper threshold value then the BIU signal is asserted for that channel.

3.3.8 0x28 - BIU Lower Threshold Register

Register Name	SM2200_REG_BIU_THRESHOLD
Register Address	0x28
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LTHRESH[15]	LTHRESH[]	LTHRESH[]	LTHRESH[]	LTHRESH[]	LTHRESH[]	LTHRESH[]	LTHRESH[]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LTHRESH[]	LTHRESH[]	LTHRESH[]	LTHRESH[]	LTHRESH[]	LTHRESH[]	LTHRESH[]	LTHRESH[0]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

This is a 16 bit register which is used to compare the amount of energy measured in a channel. If the energy is below the lower threshold value then the BIU signal is deasserted for that channel.

ERRATA Write behaviour – The register address for BIU lower threshold and upper threshold overlap, all write operations at this address will update both the lower and upper threshold registers.

ERRATA Read behaviour – The register address for BIU lower threshold and upper threshold overlap, all reads operations at this address will read the value of the upper threshold register.

3.3.9 0x2A - Receiver Channel Enable Register

Register Name	SM2200_REG_CHANNEL_EN
Register Address	0x2A
SPI Format	Format 2 (24 bit)
Index Range	NA

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	0	0	0	0	0	0	CHEN[17]	CHEN[]
Mode	R	R	R	R	R	R	R/W	R/W
Reset	-	-	-	-	-	-	1	1

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	CHEN[]	CHEN[]	CHEN[]	CHEN[]	CHEN[]	CHEN[]	CHEN[]	CHEN[]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CHEN[]	CHEN[]	CHEN[]	CHEN[]	CHEN[]	CHEN[]	CHEN[]	CHEN[0]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

The receiver channel enable register allows the user to switch off channels that are not being used in order to save power. Each bit in the register corresponds to one of the 18 channels. Setting the bit to ‘1’ enables the channel and clearing it (‘0’) disables the channel. When a channel is disabled, it also clears the averaged noise reading on that channel.

3.3.10 0x2C - BIU Status Register

Register Name	SM2200_REG_BIU_STATUS
Register Address	0x2C
SPI Format	Format 2 (24 bit)
Index Range	NA

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	0	0	0	0	0	0	BIU[17]	BIU[]
Mode	R	R	R	R	R	R	R	R
Reset	-	-	-	-	-	-	0	0

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	BIU[]	BIU[]	BIU[]	BIU[]	BIU[]	BIU[]	BIU[]	BIU[]
Mode	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	BIU[]	BIU[]	BIU[]	BIU[]	BIU[]	BIU[]	BIU[]	BIU[0]
Mode	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

This is an 18-bit register which is used to determine which channel is currently in use.

3.3.11 0x32 - Channel Attenuation Register

Register Name	SM2200_REG_CHANNEL_ATTEN_i18
Register Address	0x32
SPI Format	Format 1 (16 bit)
Index Range	Channel[i]; i = 0…17

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	0	0	0	0	0	0	0	0
Mode	R	R	R	R	R	R	R	R
Reset[i]	-	-	-	-	-	-	-	-

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	0	0	0	0	0	ATT[2]	ATT[1]	ATT[0]
Mode	R	R	R	R	R	R/W	R/W	R/W
Reset[i]	-	-	-	-	-	1	1	1

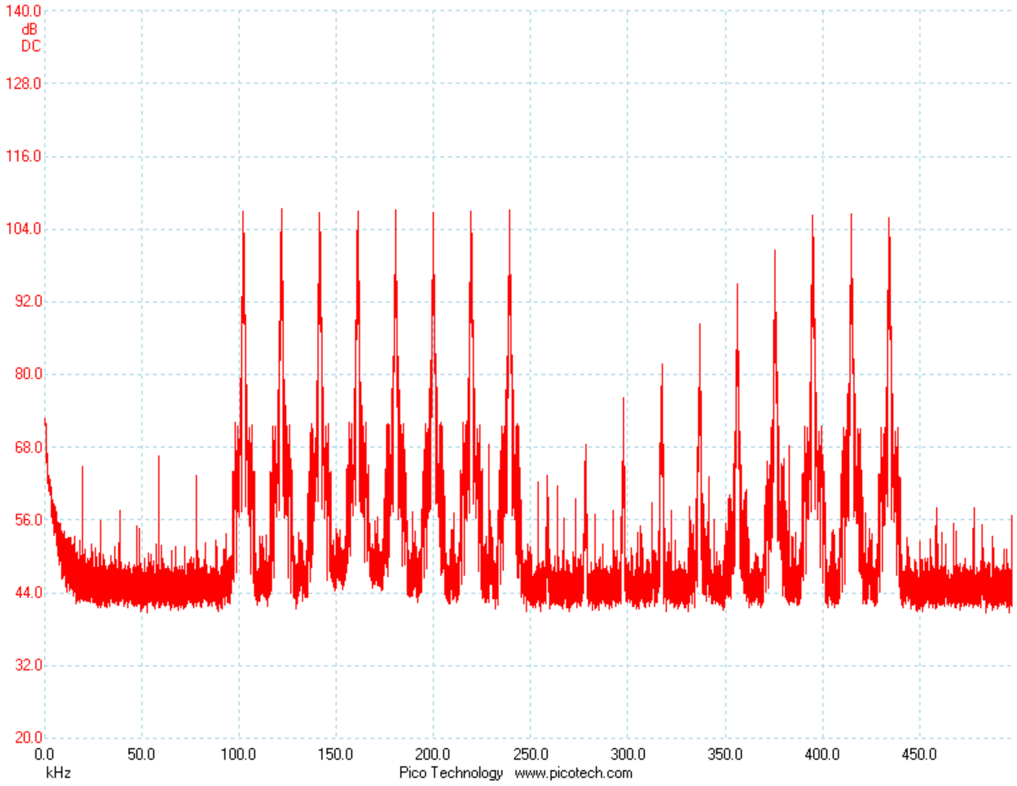
This register allows each channel’s attenuation to be controlled separately. It is applied before all active channels are combined together and allows for user controllable pre-emphasis filter. After the pre-emphasis filter the channels are combined and the SM2200_REG_TX_LEVEL register controls the final output voltage.

This allows for the transmission attenuation to be set in a range 0dB to -21dB in 3dB steps. The Table 6 below gives the various attenuation settings.

Table 6 – Channel Attenuation Selection

ATT[2:0]	Attenuation Level
7	0dB
6	3dB
5	6dB
4	9dB
3	12dB
2	15dB
1	18dB
0	21dB

The figure below shows the effect of each attenuation setting. Starting at channel 8, the attenuation was set from 0 (21dB) to 7 (0dB).



3.3.12 0x34 - Transmit Buffer Register

Register Name	SM2200_REG_TX_BUFFER_i18
Register Address	0x34
SPI Format	Format 3 (2 + N Byte)
Index Range	TX_BUFFER[i]; i = 0…17

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TXDATA[7]	TXDATA[6]	TXDATA[5]	TXDATA[4]	TXDATA[3]	TXDATA[2]	TXDATA[1]	TXDATA[0]
Mode	W	W	W	W	W	W	W	W
Reset[i]	X	X	X	X	X	X	X	X

The transmit buffer register holds the data that will be transmitted on the channel where the register is mapped to.

The transmit buffer register is an indexed register where the buffer number to be written to is selected by the value in the Index Select Register. After writing the value of the buffer number to be written to in the Index Select Register, the buffer data pointer is always reset – the data will be written starting on TxData (0) as shown below. SPI transactions used for the transmit buffer is not like the other registers as the packet is written in a burst – it uses SPI format 3. Shown below the length of the transaction is the length of the packet (size N).

A sample way of writing to the TX buffer is shown in the flowchart on the transmission section 3.4.3.

3.3.13 0x36 - Transmit Send Packets Register

Register Name	SM2200_REG_SEND_PKTS
Register Address	0x36
SPI Format	Format 2 (24 bit)
Index Range	NA

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	-	-	-	-	-	-	GO[17]	GO[16]
Mode	-	-	-	-	-	-	W	W
Reset	-	-	-	-	-	-	0	0

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	GO[15]	GO[14]	GO[13]	GO[12]	GO[11]	GO[10]	GO[9]	GO[8]
Mode	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	GO[7]	GO[6]	GO[5]	GO[4]	GO[3]	GO[2]	GO[1]	GO[0]
Mode	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

After the data is written in the TX buffer register, it will not be transmitted until the channel mapped to that register is given the “GO” signal to transmit. The Transmit Send Packets Register is used to initiate packet transmission by the SM2200; during transmit, writing to this register is usually the last operation.

GO[17:0] corresponds to the physical channels CH[17:0]. Each channel that has the corresponding GO[n] bit set will initiate the transmission of data from its configured TX Buffer.

3.3.14 0x3A - Transmit Buffer Mapping Register

Register Name	SM2200_REG_TX_BUF_MAP_i18
Register Address	0x3A
SPI Format	Format 1 (16 bit)
Index Range	Channel[i]; i = 0...17

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	0	0	0	0	0	0	0	0
Mode	R	R	R	R	R	R	R	R
Reset[i]	-	-	-	-	-	-	-	-

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	0	0	0	BUFFER[4]	BUFFER[3]	BUFFER[2]	BUFFER[1]	BUFFER[0]
Mode	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset[i]	-	-	-	0	0	0	0	0

BUFFER[4:0] – This is used for Tx buffer selection

The Transmit buffer mapping register is an indexed register used for the TX Buffer – Channel mapping.

The transmitter has a very flexible buffering structure. This flexible nature allows the transceiver to be used in many different ways. The transceiver contains 18 separate transmit buffers. These buffers can be mapped to an individual channel or multiple channels to create some kind of redundancy. The redundancy is the number of channels that are mapped to a buffer.

The diagrams below show some possible configurations of the buffers to allow different modes. The first is a one to one buffer relationship or Redundancy 1. In this mode, there is no redundancy since data on each buffer will be sent only “once”.

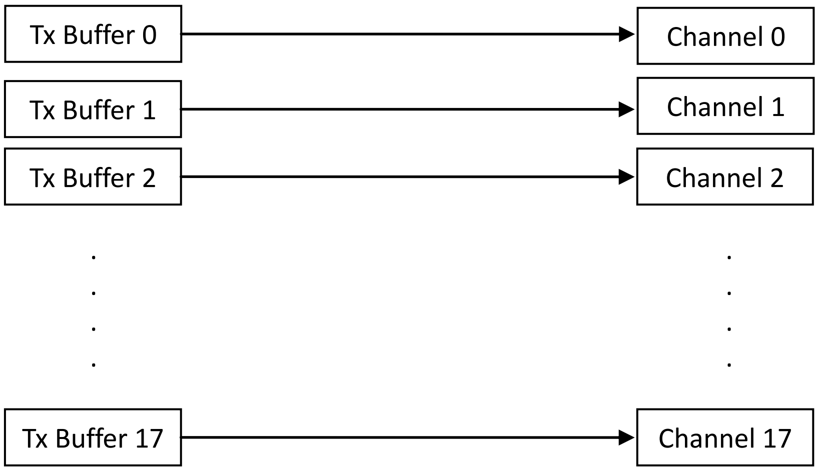


Figure 18 - One to One buffer relationship: Redundancy 1

The second is a one to many buffer relationship. The figure below shows an 18-redundancy scheme wherein the data on Tx Buffer 0 will be effectively sent "18 times", once for each channel.

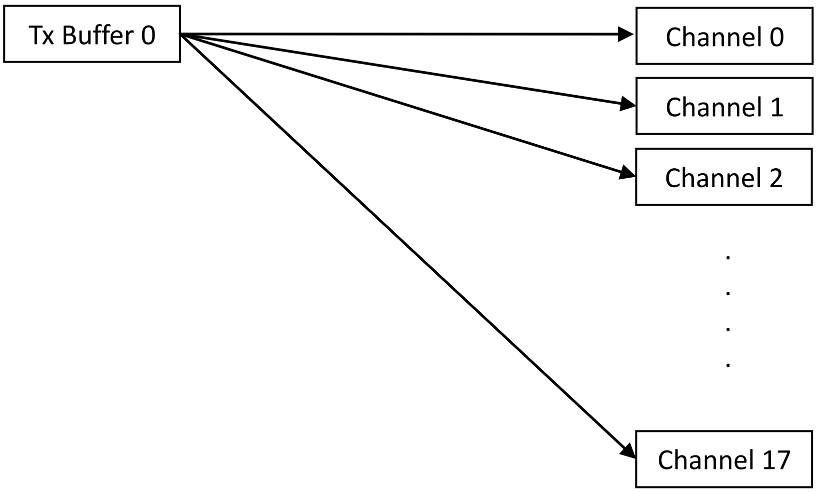


Figure 19 - One to 18 buffer relationship: Redundancy 18

3.3.15 0x3C - Transmit Level Register

Register Name	SM2200_REG_TX_LEVEL
Register Address	0x3C
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	0	0	0	0	0	0	0	TXV[8]
Mode	R	R	R	R	R	R	R	R/W
Reset	-	-	-	-	-	-	-	1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TXV[7]	TXV[6]	TXV[5]	TXV[4]	TXV[3]	TXV[2]	TXV[1]	TXV[0]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

TXV[8:0] – Transmit level

The transmit level register is used to regulate the DAC output voltage level. The transmit voltage can be regulated to a certain level to ensure that the DAC is able to output the maximum voltage for a given number of carriers enabled. See the transmission voltage and control section 3.5.

The maximum transmit level that can be set is 0x1FF or 511. By default, the transmit level is set to half (0x100).

3.3.16 0x40 - Receive Status Register

Register Name	SM2200_REG_RX_STATUS
Register Address	0x40
SPI Format	Format 2 (24 bit)
Index Range	NA

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	0	0	0	0	0	0	RXSTAT[17]	RXSTAT[16]
Mode	R	R	R	R	R	R	R	R
Reset	-	-	-	-	-	-	0	0

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RXSTAT[15]	RXSTAT[14]	RXSTAT[13]	RXSTAT[12]	RXSTAT[11]	RXSTAT[10]	RXSTAT[9]	RXSTAT[8]
Mode	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RXSTAT[7]	RXSTAT[6]	RXSTAT[5]	RXSTAT[4]	RXSTAT[3]	RXSTAT[2]	RXSTAT[1]	RXSTAT[0]
Mode	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

The receive status register quickly tells the host which channels have a new packet available to read. This register can be polled to check for packet receptions or use the interrupt signal.

3.3.17 0x42 - Channel Noise Register

Register Name	SM2200_REG_CHANNEL_NOISE_i18
Register Address	0x42
SPI Format	Format 1 (16 bit)
Index Range	Channel[i]; i = 0…17

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	NOISE[15]	NOISE[14]	NOISE[13]	NOISE[12]	NOISE[11]	NOISE[10]	NOISE[9]	NOISE[8]
Mode	R	R	R	R	R	R	R	R
Reset[i]	0	0	0	0	0	0	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	NOISE[7]	NOISE[6]	NOISE[5]	NOISE[4]	NOISE[3]	NOISE[2]	NOISE[1]	NOISE[0]
Mode	R	R	R	R	R	R	R	R
Reset[i]	0	0	0	0	0	0	0	0

This indexed register indicates the average noise that the channel detects. This noise is averaged heavily due to the noise fluctuating wildly on the power line. The averaged noise reading for a channel can be read by first selecting the channel number through the Index Select register (0x20), and then reading the noise value from NOISE[15:0].

The averaged noise reading for a pair of channels may be reset by disabling and then re-enabling the given pair of channels using the Channel Enable register (0x2A). The pair of channels is a consecutive pair, starting from an even-numbered channel. For example, channels 0 and 1, or channels 2 and 3 may be reset together. Allow some time after resetting for this averaged noise reading to stabilize.

3.3.18 0x48 - Receive Buffer Register

Register Name	SM2200_REG_RX_BUFFER_i18
Register Address	0x48
SPI Format	Format 3 (2 + N Byte)
Index Range	Channel[i]; i = 0…17

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DATA[7]	DATA[6]	DATA[5]	DATA[4]	DATA[3]	DATA[2]	DATA[1]	DATA[0]
Mode	R	R	R	R	R	R	R	R
Reset[i]	X	X	X	X	X	X	X	X

The receive buffer register is an indexed register where the buffer number to be read is selected by the value in the Index Select Register. After writing the value of the buffer number to be read in the Index Select Register, the buffer data pointer is always reset – the data read starts on RxData (0) as shown below. SPI transactions used for the receive buffer is not like the other registers as the packet is written in a burst – it uses SPI format 3. Shown below the length of the transaction is the length of the packet (size N).

0x48	**RxData(0)**	**RxData(1)**	**RxData(2)**	**RxData(3)**	**RxData(4)**	**……**	**RxData(N)**

Channel number is determined from which receive bit gets set in the Receive Status register (0x40) and then it determines which number should be set in the Index Select register (0x20) before reading the Rx Buffer. All the 18 channels in the SM2200 contain their own buffer for holding packets until they are read out of the SPI interface. Each of these 18 buffers has 128 bytes* in total meaning the maximum packet size is 2,304 bytes. This means in total 2,304 bytes of data can be transferred in one parallel transaction. Each buffer is protected from being overwritten by incoming data if the data has not been accessed by the SPI. This is to stop corrupted data from contaminating the buffers.

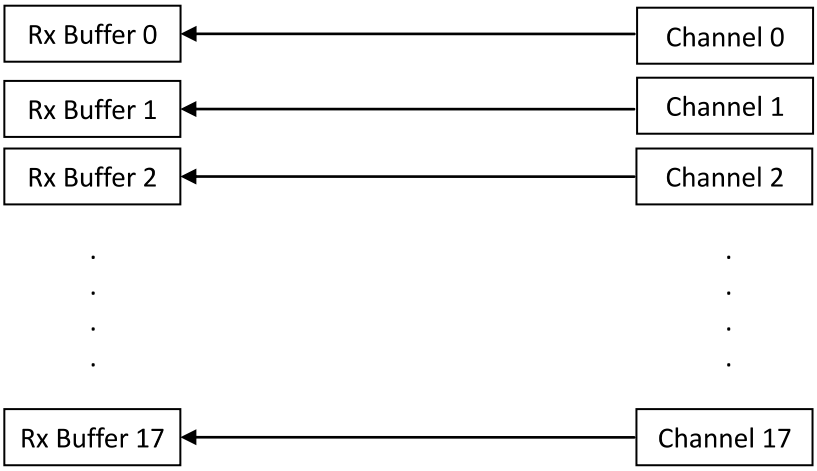


Figure 20 - Rx Buffer and Channel Mapping

* The maximum payload size without Sequence#, ECC and address is 125 bytes.
The minimum size of Packet Data (payload) is 5 bytes.

The packet length is determined from Packet Size register (0x26) after setting Index Select register with the Channel number and which then determines how many bytes to subsequently read from the Rx Buffer to obtain the Packet Data.

The format of the Packet Data read from the Rx buffer is shown below. Note that availability and size of some fields depends on the setting of the transceiver configuration register.

Address (n bytes)	Packet Data	Seq# (1byte)	RSSI (High byte)	RSSI (Low byte)

Figure 21 shows a simplified version of the state machine. The buffer memory is locked upon receiving a correct packet for protection. The interrupt pin is then asserted telling the host processor that a reception event has occurred. The interrupt pin will stay asserted until the host reads the interrupt register. The host then acknowledges the packet reception by reading the packet size register. Once this has occurred the buffer is no longer locked and is free for more packets. After the size of the packet has been read it is important to read out the data that is contained in the buffer before the data is overwritten again.

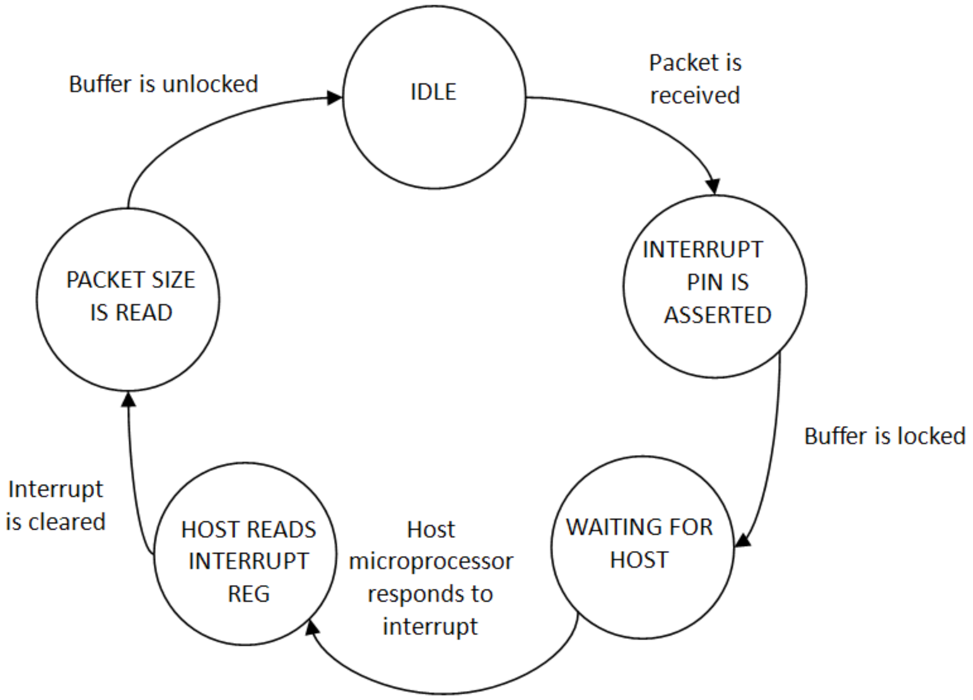


Figure 21 - Receive process state machine for a single channel

3.3.19 0x50 - Interrupt Mask Register

Register Name	SM2200_REG_INT_MASK
Register Address	0x50
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	0	0	0	0	0	0	0	0
Mode	R	R	R	R	R	R	R	R
Reset	-	-	-	-	-	-	-	-

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	INTMSK7	INTMSK6	INTMSK5	INTMSK4	INTMSK3	INTMSK2	INTMSK1	INTMSK0
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	0

The interrupt pin (Pin 25) of SM2200 is used to alert the host MCU that an event has occurred in the SM2200. This allows the host MCU to do other calculations or even go to sleep when there are no events that need its attention. There are two registers associated with controlling the interrupt as described below.

The interrupt mask register is used to turn off certain interrupt sources. For example if you do not require the BIU interrupt then it can be turned off by masking out the appropriate interrupt bit. Please refer to 0x52 – Interrupt Source register for details of which interrupt can be masked.

3.3.20 0x52 - Interrupt Event Register

Register Name	SM2200_REG_INT_EVENT
Register Address	0x52
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	0	0	0	0	0	0	0	0
Mode	R	R	R	R	R	R	R	R
Reset	-	-	-	-	-	-	-	-

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved	Reserved	Reserved	Reserved	TXBUF Empty	BIUCHG	RESET	PKTRCVD
Mode	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

The Interrupt Event register tells the host MCU what was the cause of the interrupt to trigger. Below is a list of the interrupt sources contained in the register. All of these sources can be masked out via the Interrupt Mask Register (0x50), the event register bit will still get set even when the corresponding Mask Register bit is set, it will not cause the IRQ pin to be asserted though.

PKTRCVD – Packet Received Interrupt
The interrupt signal is asserted every time a valid error free packet is successfully received. If an interrupt has previously occurred and the interrupt has not been acknowledged by the host MCU then the interrupt pin will stay low even though a new event has occurred. The interrupt pin is cleared when the interrupt event register has been read. Figure 22 shows how the interrupt pin stays asserted if the host MCU does not read the interrupt register.

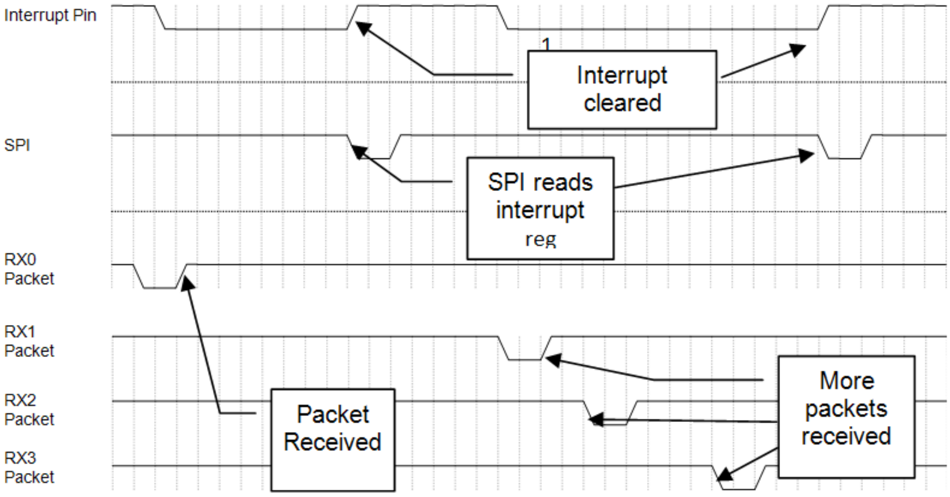


Figure 22 - Packets and interrupt timing

RESET – Reset Interrupt
Asserted whenever the SM2200 comes out of a hard reset.

BIUCHG – BIU Changed Interrupt
Asserted whenever the BIU Status register changes.

TXBUF EMPTY – Transmit Buffer Done Interrupt
Asserted when the transmit buffers for the current transmission have been read. Note that this interrupt only monitors the buffers mapped to the channels initiated by the Tx Send Packet register. For example, in a one-to-one buffer-channel mapping, when all 18 buffers have data in them and only channel 0 was sent (only bit 0 in the TX packet send register is set), TXBUF Empty interrupt will be asserted once all of the data in buffer 0 has been read but buffers 1 to 17 still has their data in them.

Because the PHY needs to add a few more bytes at the end of the packet, the TXBuf Empty interrupt will always be asserted before the actual transmission is complete.

All of the interrupts are cleared upon reading the register, at which time the interrupt pin is de-asserted. Below is a diagram of the timing of a packet reception and interrupt signal.

3.3.21 0x56 - Transceiver Configuration Register

Register Name	SM2200_REG_TXCVR_CONFIG
Register Address	0x56
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	Reserved	Reserved	Reserved	Reserved	ALEN[3]	ALEN[2]	ALEN[1]	ALEN[0]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved	Reserved	Reserved	Reserved	Reserved	TXREG	SNEN	ECCEN
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

The transceiver configuration register is used to set the different options in the SM2200 PHY. The parameters for register are explained below.

ECCEN – Error Correction Coding Enable

SM2200 implements error correction by default. The error correction coding increases the effective length of the packet but makes it more robust against power line noise. The ECCEN bit is used to turn on/off this error correction coding.

SNEN – Sequence Number Enable

This turns on the embedding of a sequence number into the packet. The sequence number is one byte long and is inserted after the data/payload. Sequence numbering allows the SM2200 to automatically identify duplicate packets when using redundancy 18. If other redundancy is used, it is recommended to implement packet duplicate detection on the host application.

TXREGENb – Transmission Voltage Regulation Enable (active low)

This bit enables/disables the regulation of the level of the transmitted signal. Please refer to Transmission Voltage Control and Regulation section.

0 : Enabled
1 : Disabled

ALEN[3:0] – Address Length

This field defines the length of the address, in bytes. Valid ranges are from 0 to 8 bytes. 0 means addressing is disabled (broadcast).

3.3.22 0x60 - Node Address Register 0

Register Name	SM2200_REG_NODE_ADDR0
Register Address	0x60
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADDR[15]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDREG[0]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

3.3.23 0x62 - Node Address Register 1

Register Name	SM2200_REG_NODE_ADDR1
Register Address	0x62
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADDR[31]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDREG[16]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

3.3.24 0x64 - Node Address Register 2

Register Name	SM2200_REG_NODE_ADDR2
Register Address	0x64
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADDR[47]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDREG[32]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

3.3.25 0x66 - Node Address Register 3

Register Name	SM2200_REG_NODE_ADDR3
Register Address	0x66
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADDR[63]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDR[]	ADDREG[48]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Node Address Register 0 – 3 is used to contain the address of the SM2200 (when enabled in the transceiver configuration register). These register equates to 8 bytes in total to correspond to the maximum 8 bytes that can be written in the address field in the transceiver configuration register.

The address has to be at the start of the packet and is compared starting from the most significant byte. For an address length of n bytes, the packet should look like:

Addr(n-1)	**Addr(n-2)**	**...**	**Addr(0)**	**Payload**

Note that address 0xFF is a broadcast command and the value set in the Node Address register will be ignored (i.e. if the node address is 0x0201 with address mask of 0xFFFF and the destination address is 0xFF01, node 0x0201 will still receive the data.)

3.3.26 0x68 - Node Address Mask Register 0

Register Name	SM2200_REG_NODE_ADDR_MASK0
Register Address	0x68
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	AMASK[15]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[0]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

3.3.27 0x6A - Node Address Mask Register 1

Register Name	SM2200_REG_NODE_ADDR_MASK1
Register Address	0x6A
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	AMASK[31]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[16]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

3.3.28 0x6C - Node Address Mask Register 2

Register Name	SM2200_REG_NODE_ADDR_MASK2
Register Address	0x6C
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	AMASK[47]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[32]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

3.3.29 0x6E - Node Address Mask Register 3

Register Name	SM2200_REG_NODE_ADDR_MASK3
Register Address	0x6E
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	AMASK[63]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[]	AMASK[48]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

The node address mask register is used as a 64 bit address filter. The host MCU sets the number of bits used for the address filter (up to a maximum of 64 bits) and the address of the device.

When a data transmission is received by the SM2200 it checks the first number of bits received as set by the filter and compares these to the filter value. If the values don't match, the SM2200 rejects the packet and doesn't write the data to the receive buffer, therefore no interrupt is sent to the host MCU.

Combined with the address mask register, the user can implement a TCP-IP style address mask.

Example: If we have a 16bit address we set the ALEN = 0x02 (two bytes). We can have a section of the address as a domain i.e.:

16 bit address	
8bit Domain	8bit Node Address
0x30	0xAA

So we could have a domain of 0x30 and a node address of 0xAA. If we only want to receive packets for the address 0x30AA then we set the mask to 0xFFFF. If we want to turn the node into a router for the domain then we can set the mask to 0xFF00. This means we will receive any packet addressed to the domain.

If you do not want to use the address filtering then set the ALEN = 0. The Address has to be at the start of the packet:

Address	**Payload**

3.3.30 0x70 - AGC Control Register

Register Name	SM2200_REG_AGC_CTRL
Register Address	0x70
SPI Format	Format 1 (16 bit)
Index Range	NA

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ATTACK[3]	ATTACK[2]	ATTACK[1]	ATTACK[0]	AGCMODE	NLEVEL[2]	NLEVEL[1]	NLEVEL[0]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	0	1	1	1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**AGCU**[3]	**AGCU**[2]	**AGCU**[1]	**AGCU**[0]	**AGCL**[3]	**AGCL**[2]	**AGCL**[1]	**AGCL**[0]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	1	0	0	1	1

The SM2200 has two AGC registers that are used for the setup of the AGC. These are the AGC control register and the AGC Level Array register. The AGC Control register is used for setting the thresholds that the AGC will operate in, the mode in which it will operate, the time taken for the AGC to respond and the number of AGC output bits that will be enabled.

The AGC Control register of SM2200 is set by the user via the host MCU using SPI. There are five settings stored in this register, these are as follows.

AGCU - Automatic Gain Control Upper Level
This is used to set the upper limit that the automatic gain control will start to reduce the gain of the input signal as shown in Figure 23.

AGCL - Automatic Gain Control Lower Level
This is used to set the lower limit that the automatic gain control will start to increase the gain of the input signal as shown in Figure 23.

AGCMODE - Automatic Gain Control Mode Select
This is used to select between Pulse Width Modulation (PWM) mode when low, or direct mode when high. When the AGCMODE is set to PWM mode a pulse width modulated waveform is outputted from the SM2200 on AGC0 (Pin 21) that is to be connected to an external third party PWM driver IC. When the AGCMODE is set to direct mode, the pins for specified in NLEVEL will be used to control external AGC circuitry.

NLEVEL[2:0] – Number of AGC Level
This is used to set the pins that the AGC will use to control the external circuitry. NLEVEL[0], NLEVEL[1] and NLEVEL[2] equate to AGC0 (Pin 21), AGC1 (Pin 16) and TX_LED (Pin 22).

ATTACK[3:0] - AGC Attack Time
This is the amount of time that is required before the AGC adjusts the external gain circuitry as shown in **Error! Reference source not found.**. This allows the user to adjust for small fast transients on the power ine.

AGC Attack time = (ATTACK+1)*100µs.

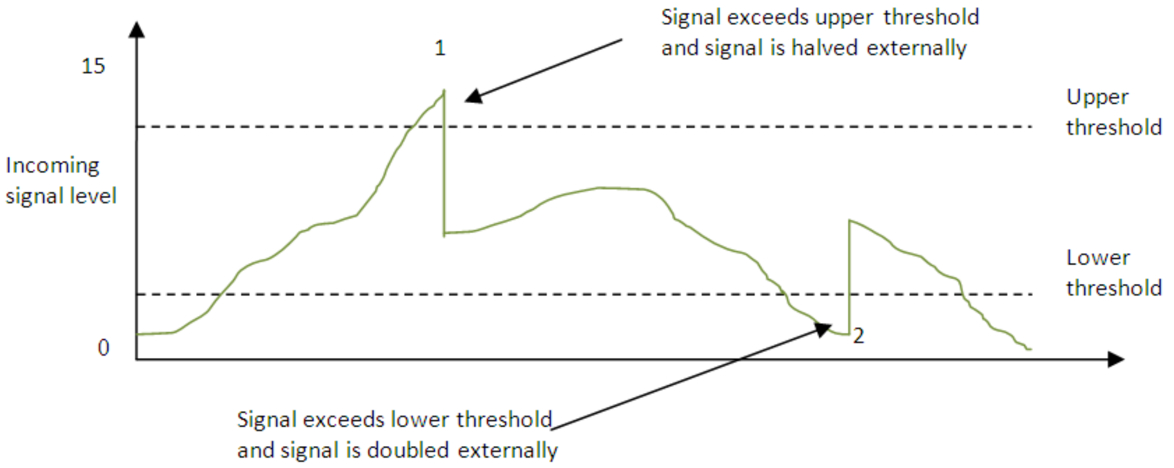

15
0
Incoming signal level
1
Signal exceeds upper threshold and signal is halved externally
Upper threshold
Lower threshold
2
Signal exceeds lower threshold and signal is doubled externally

Figure 23 - AGC Block Upper and Lower Thresholds

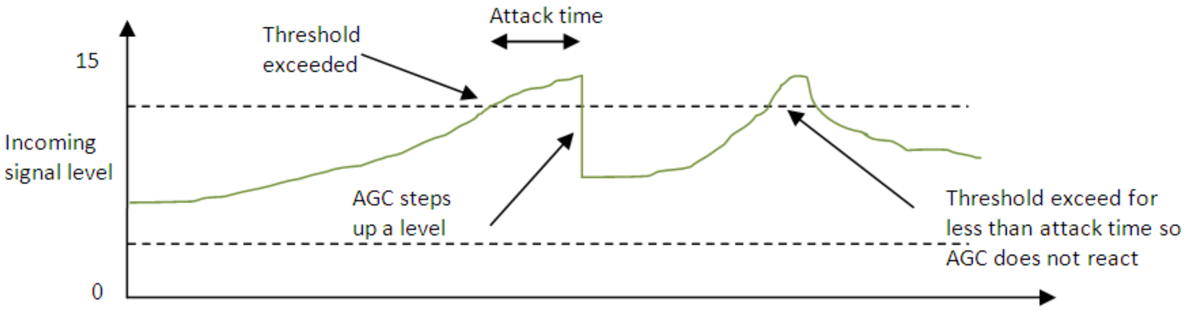


Figure 24 - Basic Design Intent of the AGC Block

3.3.31 0x72 - AGC Level Array Register

Register Name	SM2200_REG_AGC_LEVEL_ARRAY_i8
Register Address	0x72
SPI Format	Format 1 (16 bit)
Index Range	i = 0…7

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	0	0	0	0	0	0	0	0
Mode	R	R	R	R	R	R	R	R
Reset[i]	-	-	-	-	-	-	-	-

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LEVEL[7]	LEVEL[6]	LEVEL[5]	LEVEL[4]	LEVEL[3]	LEVEL[2]	LEVEL[1]	LEVEL[0]
Mode	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset[7]	1	1	1	1	1	1	1	1
Reset[6]	0	1	1	1	1	1	1	1
Reset[5]	0	0	1	1	1	1	1	1
Reset[4]	0	0	0	1	1	1	1	1
Reset[3]	0	0	0	0	1	1	1	1
Reset[2]	0	0	0	0	0	1	1	1
Reset[1]	0	0	0	0	0	0	1	1
Reset[0]	0	0	0	0	0	0	0	0

The AGC levels register is used for setting the level of step that the AGC will take to change the AGC pins. Both of these registers are set by the host MCU using the Serial Peripheral Interface (SPI).

The SM2200 can support up to 8 levels of gain in the AGC block (0-7). The basic algorithm is that each time the upper threshold is exceeded on the input for longer than the “Attack time” the AGC level is incremented. Each time the signal goes below the lower threshold the level is decremented.

The AGC Level register is an array register used to set-up which of the pins will be asserted first. There are 8 registers to correspond to the maximum combination of the 3 AGC pins. The AGCLevel bit that corresponds to each pin is shown in the table below. Please note that you need to write Index Select Register (0x20) first before you write to AGC Levels (0x72). You also need to specify NLEVEL[2:0] in the AGC Control register (i.e. 0x70[10:8]).

For the values in the register array shown below, only AGC0 and AGC1 are used, so only NLEVEL1 and NLEVEL0 are set on the AGC control register. Also, it can be seen that at Level 1, AGC0 is set meaning that AGC0 will be asserted first before AGC1. Once the next AGC level is reached, AGC1 will be asserted and AGC0 de-asserted. When the last level is reached, both AGC1 and AGC0 will be asserted. Note that these values need to correspond to the hardware circuit that is used for AGC control.

Table 7 -- AGC Level Array

			Pin16 (AGC1)	Pin21 (AGC0)	Pin22 (TXLED)				
	AGCLev7	AGCLev6	AGCLev5	AGCLev4	AGCLev3	AGCLev2	AGCLev1	AGCLev0	
Level 0	0	0	0	0	0	0	0	0	0x00
Level 1	0	0	0	1	0	0	0	0	0x10
Level 2	0	0	1	0	0	0	0	0	0x20
Level 3	0	0	1	1	0	0	0	0	0x30
Level 4	0	0	0	0	0	0	0	0	Default
Level 5	0	0	0	0	0	0	0	0	Default
Level 6	0	0	0	0	0	0	0	0	Default
Level 7	0	0	0	0	0	0	0	0	Default

The following figure shows how the AGC on the SM9221-V4 works. For a very small signal input in the receiver, both AGC pins are off. As soon as the input signal reaches the threshold, the AGC1 pin is asserted to reduce the signal on the ADC input. As the signal becomes larger again, the AGC0 pin is asserted while de-asserting the AGC1 pin. But when the signal becomes too large, both AGC pins assert to further reduce the signal seen by the ADC.

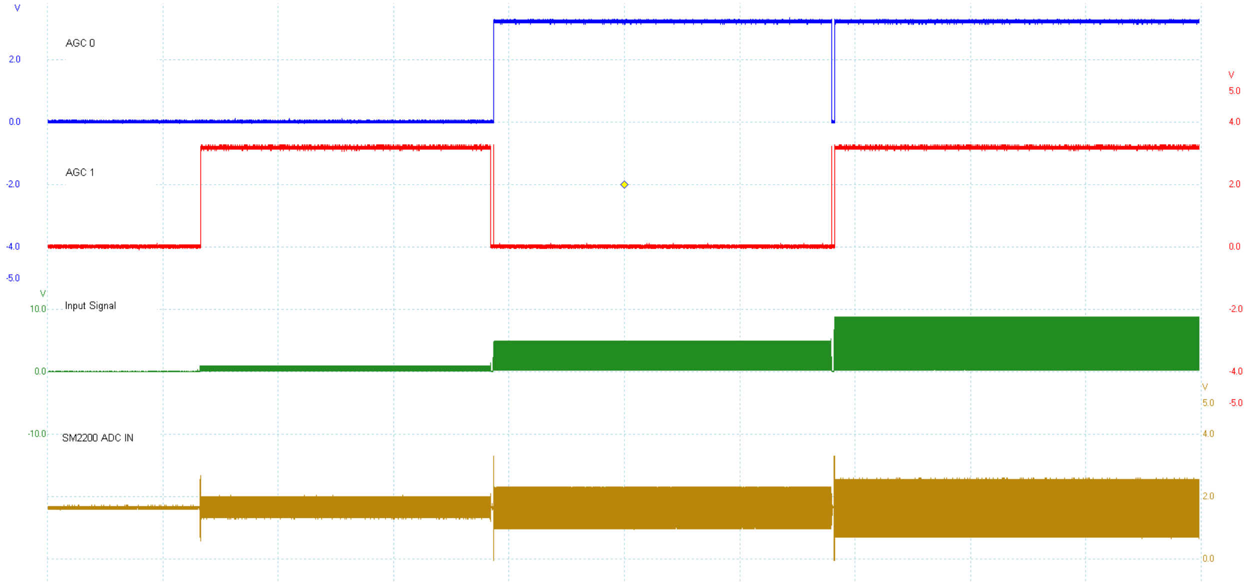


Figure 25 -- SM2200 AGC

The AGC Levels Array register is also used as the AGC PWM word register. There are 8 PWM word registers (one for each level). These registers are accessed using the Index Select Register register.

In PWM mode AGC(1)/Pin 16 of the SM2200 can be used to output a Pulse Width Modulated (PWM) waveform to a voltage controlled amplifier. Each AGC level register is used as a corresponding PWM word to produce a corresponding waveform of varying duty cycle on pin 16. In PWM mode with all bits of a level register turned off Pin 16 will be outputting a signal that is high 1/256 of the time. Bit0 turn on will add another 1/256 “time high” slice, Bit1 - 1/128, Bit2 – 1/64 up to Bit7 which will add 1/2 “time high” slice. With all bits turned on pin 16 will be static at logic “1” (since 1/2+ 1/4+ 1/8 + 1/16 + 1/32 + 1/64 + 1/128 + 1/256 + 1/256 = 1).

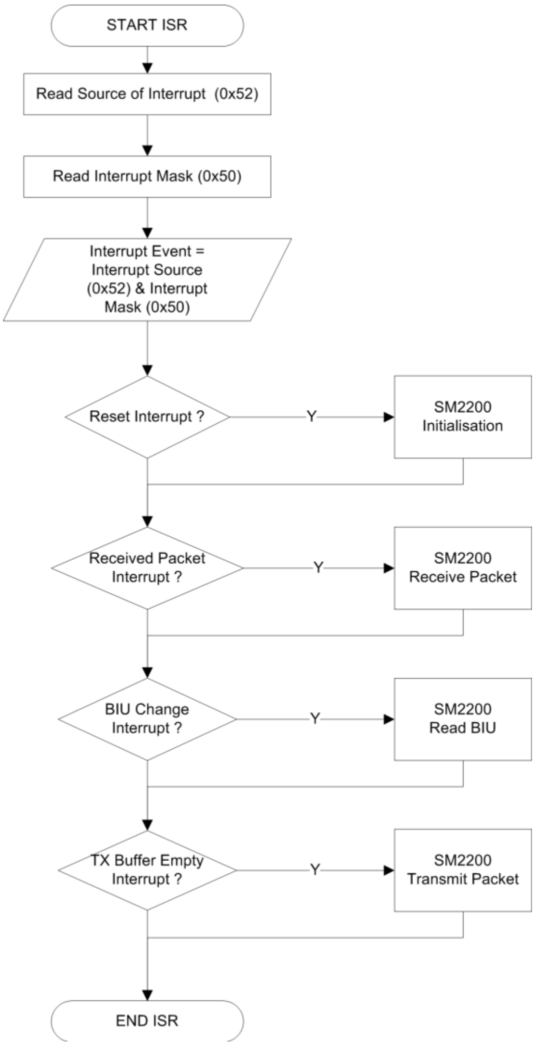
3.4 SM2200 Software Processes

There are four main software processes or functions that need to be implemented when using the SM2200: interrupt service routine, initialization, packet transmit and packet receive. Each function is explained below and a sample flowchart is presented for reference.

3.4.1 SM2200 Interrupt Service Routine

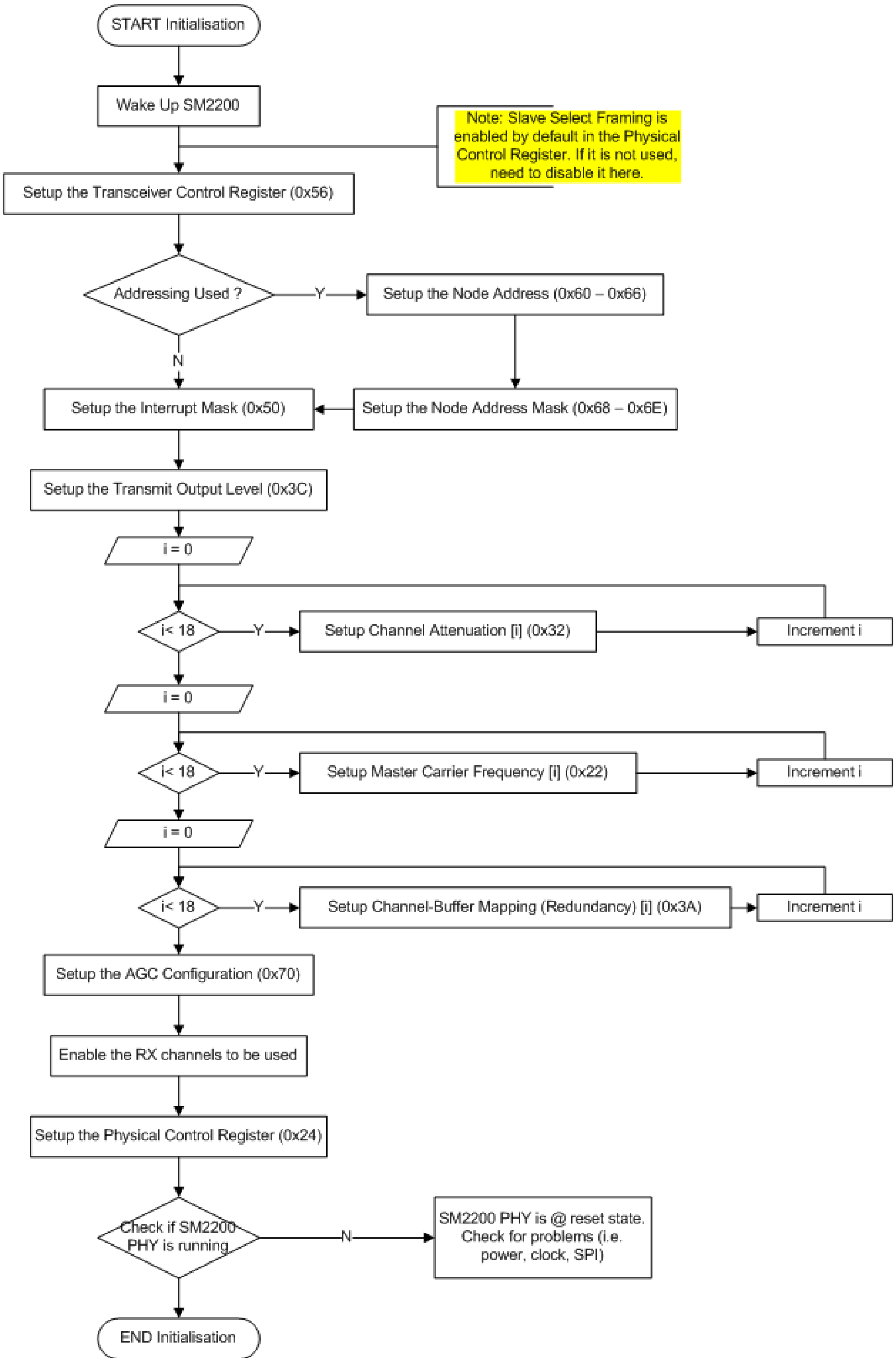
There are two ways for the host microcontroller to know whether the SM2200 needs some attention: polling and interrupts. In the polling mode, the host microcontroller needs to talk to SM2200 every time to check if it needs servicing. This wastes a lot of microcontroller time because the host microcontroller talks to the SM2200 even though it does not need any action. Using interrupt is the preferred method since the SM200 asserts the interrupt pin only when it needs attention, allowing the host microcontroller to sleep or do other tasks when the SM2200 is idle.

The flowchart below shows an interrupt service routine used when processing the SM2200 interrupt. At power on and reset, the SM2200 would always assert the interrupt pin to indicate a reset interrupt event. When the SM2200 comes out of reset, all register settings are set to their default values and the host should initialize the settings accordingly. A sample code is provided in the *SM2200_ISR.c.*


START ISR
Read Source of Interrupt (0x52)
Read Interrupt Mask (0x50)
Interrupt Event = Interrupt Source (0x52) & Interrupt Mask (0x50)
Reset Interrupt ?
Y
SM2200 Initialisation
Received Packet Interrupt ?
Y
SM2200 Receive Packet
BIU Change Interrupt ?
Y
SM2200 Read BIU
TX Buffer Empty Interrupt ?
Y
SM2200 Transmit Packet
END ISR

3.4.2 SM2200 Initialization

During initialization of the SM2200, the SM2200 PHY is enabled and the registers are set including the interrupt masks, transceiver, frequency, transmitter and receiver settings. Below is an example of the initialization procedure. A sample code is provided in the *SM2200_Init.c.*


START Initialisation
Wake Up SM2200
Note: Slave Select Framing is enabled by default in the Physical Control Register. If it is not used, need to disable it here.
Setup the Transceiver Control Register (0x56)
Addressing Used ?
Y
Setup the Node Address (0x60 – 0x66)
N
Setup the Interrupt Mask (0x50)
Setup the Node Address Mask (0x68 – 0x6E)
Setup the Transmit Output Level (0x3C)
i = 0
i< 18
Y
Setup Channel Attenuation [i] (0x32)
Increment i
i = 0
i< 18
Y
Setup Master Carrier Frequency [i] (0x22)
Increment i
i = 0
i< 18
Y
Setup Channel-Buffer Mapping (Redundancy) [i] (0x3A)
Increment i
Setup the AGC Configuration (0x70)
Enable the RX channels to be used
Setup the Physical Control Register (0x24)
Check if SM2200 PHY is running
N
SM2200 PHY is @ reset state. Check for problems (i.e. power, clock, SPI)
END Initialisation

3.4.3 SM2200 Packet Initialization and Transmission

Before sending data over the power line, each packet needs to be configured by deciding on which channel or channels to transmit it on. Because each channel is assigned a frequency, the packet will be sent using those frequencies assigned to the channel. Also, by knowing which channel(s) to transmit the packet on, the packet needs to be written on the tx buffer where the channel(s) is/are assigned on when the redundancy was setup.

Each transmit buffer then needs to be told how big the packet is to be sent and what type of modulation it needs to use. Note that during packet initialization and transmission, the user needs to be aware of the packet redundancy being used. Below is an example of the transmit initialization and transmission for a redundancy 1 setting. For a redundancy 18 setting, only one transmit buffer (TX Buffer [0]) is used and is mapped to all the 18 channels. In this case, only TX Buffer[0] needs to be initialized, but all 18 channels needs to assert its transmit flag (Send Packet Register 0x36).

Note that due to the flexibility of transmitting only the channels that the user wants and buffer-channel redundancy, the user needs to monitor which TX buffer has been sent; otherwise, the TX buffer may be overwritten with new data even though it has not been sent yet.

If the TX buffer empty interrupt mask is enabled, when all of the data on the TX buffer mapped to the channels sent (via the Send Packet Register 0x36) has been processed or read by the PHY, the interrupt pin will be asserted to indicate TX buffer empty interrupt. Note that other buffers may still have data on them if the send bit on the channels mapped to them were not asserted.

For example, in the following figures, SM2200 is setup to be in redundancy 1 (each TX buffer is mapped to each channel (one-to-one)) and each TX buffer is written to. In the first figure, all 18 channels were told to transmit as can be seen in the spectrum. As expected, the TX buffer empty was asserted when all the buffers has been read. It is asserted before the transmission is finished because the packet length transmitted is more than that of the payload length due to the addition of CRC.

In the second figure, only channel 0 was told to transmit (the send bit on channel 0 was asserted; bit 0 on Send Packet Register 0x36). This will send the data on TX Buffer[0] but not on any of the other buffers as can be seen in the spectrum. After all of the data on TX Buffer [0] has been read, the TX Buffer empty interrupt is still asserted.

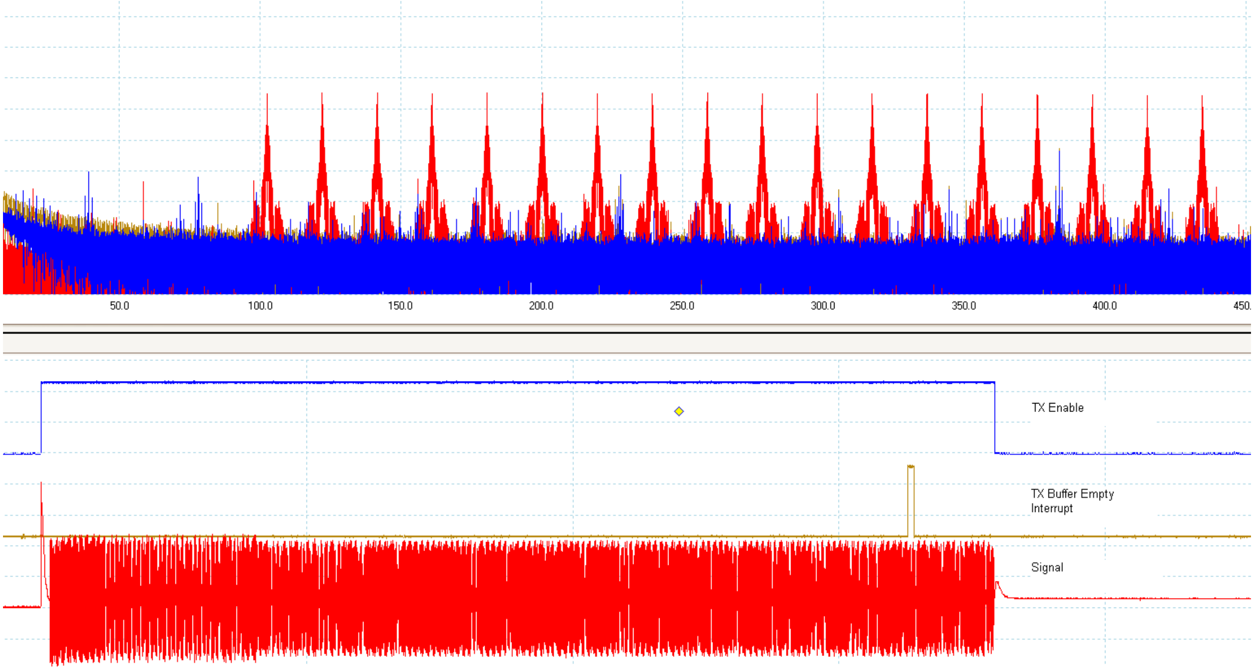


Figure 26 - TX Buffer Empty interrupt on all 18 channels

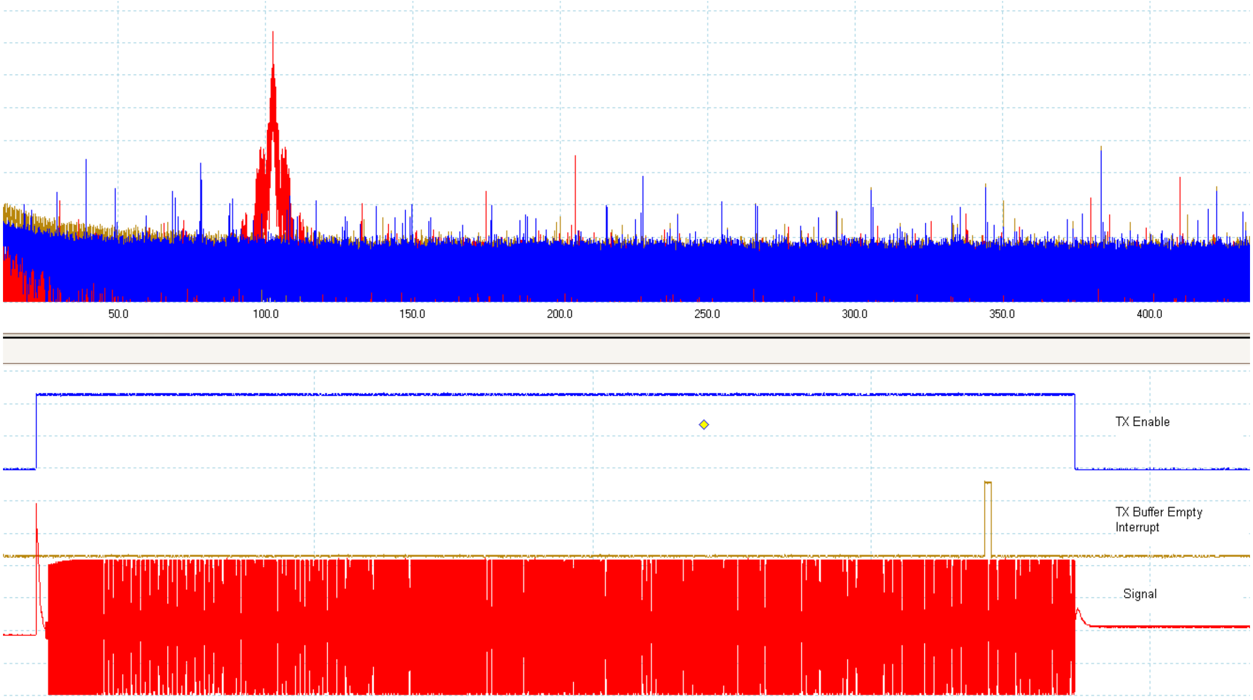
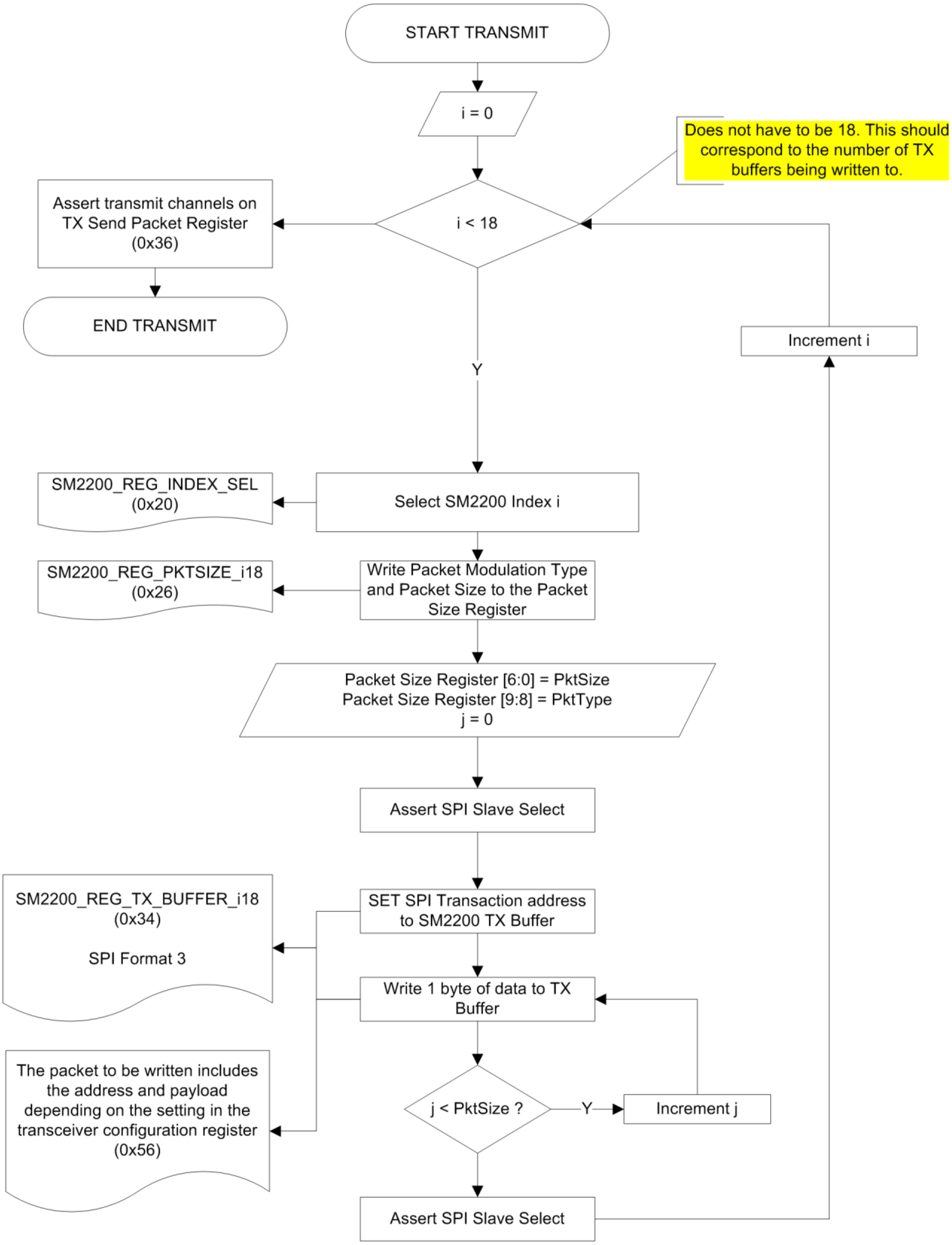


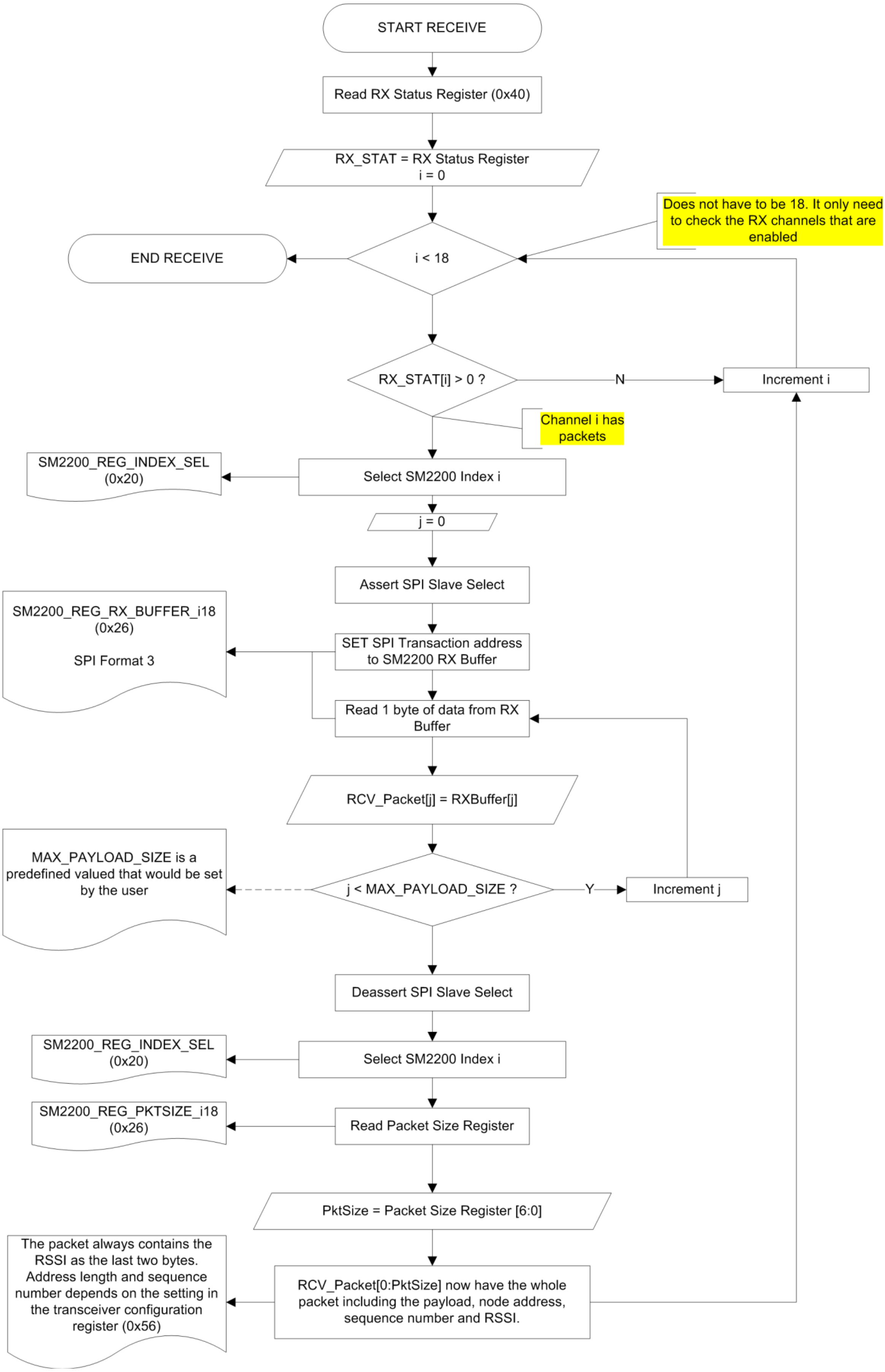
Figure 27 - TX Buffer Empty interrupt on channel 0


START TRANSMIT
i = 0
Does not have to be 18. This should correspond to the number of TX buffers being written to.
i < 18
Assert transmit channels on TX Send Packet Register (0x36)
END TRANSMIT
Increment i
Y
SM2200_REG_INDEX_SEL (0x20)
Select SM2200 Index i
SM2200_REG_PKTSIZE_i18 (0x26)
Write Packet Modulation Type and Packet Size to the Packet Size Register
Packet Size Register [6:0] = PktSize
Packet Size Register [9:8] = PktType
j = 0
Assert SPI Slave Select
SM2200_REG_TX_BUFFER_i18 (0x34)
SPI Format 3
SET SPI Transaction address to SM2200 TX Buffer
Write 1 byte of data to TX Buffer
The packet to be written includes the address and payload depending on the setting in the transceiver configuration register (0x56)
j < PktSize ?
Y
Increment j
Assert SPI Slave Select

3.4.4 SM2200 Packet Reception

SM2200 indicates reception of a new valid packet via the Receive Status Register (0x40). If the Packet Received Interrupt mask is enabled, the interrupt pin will be asserted and the host needs to check the Interrupt Event Register (0x52). If interrupt is not enabled or used, the host needs to poll the Receive Status Register to check for new packet reception. Below is a sample flowchart on how to read the received packet from the SM2200 registers.

In the flowchart below, MAX_PAYLOAD_SIZE is a number set by the user as the maximum data size to transmit and receive. Note that this number cannot exceed 125. Notice that the data in the receive buffer is read first and stored in a temporary variable RCV_Packet[]. The packet size register is then read to determine which of those packets in the RCV_Packet variable are valid data. This is done in order to avoid an incoming packet overwriting the receive buffer since reading the packet size register "unlocks" the receive buffer for the next incoming packet.


START RECEIVE
Read RX Status Register (0x40)
RX_STAT = RX Status Register
i = 0
Does not have to be 18. It only need to check the RX channels that are enabled
END RECEIVE
i < 18
RX_STAT[i] > 0 ?
N
Increment i
Channel i has packets
SM2200_REG_INDEX_SEL (0x20)
Select SM2200 Index i
j = 0
Assert SPI Slave Select
SM2200_REG_RX_BUFFER_i18 (0x26)
SPI Format 3
SET SPI Transaction address to SM2200 RX Buffer
Read 1 byte of data from RX Buffer
RCV_Packet[j] = RXBuffer[j]
MAX_PAYLOAD_SIZE is a predefined valued that would be set by the user
j < MAX_PAYLOAD_SIZE ?
Y
Increment j
Deassert SPI Slave Select
SM2200_REG_INDEX_SEL (0x20)
Select SM2200 Index i
SM2200_REG_PKTSIZE_i18 (0x26)
Read Packet Size Register
PktSize = Packet Size Register [6:0]
The packet always contains the RSSI as the last two bytes. Address length and sequence number depends on the setting in the transceiver configuration register (0x56)
RCV_Packet[0:PktSize] now have the whole packet including the payload, node address, sequence number and RSSI.

3.5 Transmission Voltage Control and Regulation

The SM2200 was designed with transmit voltage regulation control. The transmit voltage regulation is enabled when the TXREGENb bit in the transmission configuration register (i.e. 0x56[2]) is set to ‘0’.
The transmit voltage has a different impact on the signal output with TX regulation on/off.

NOTE: All of the voltages mentioned in this section are at the DAC output.

3.5.1 Constant Carrier Amplitude Mode (TXREG OFF)

One well documented drawback of OFDM is the additive effect of multiple carriers. High voltage peaks are produced when many carriers are added together but the overall average is low. This peak to average ratio (PAR) is used to determine the effectiveness of peak reduction algorithms. If the peak is high and the overall average is low (high PAR) then significant headroom needs to be added to the transmission amplifier in order to transmit enough amplitude into the power line. This increases complexity and cost of both the power supply and transmission amplifier.

The SM2200 is able to keep the PAR below 3 for most configurations. OFDM without PAR reduction will have a PAR of over 30. Figure 28 shows when carriers are added the peak voltage is increased and the overall peak to average ratio is also increased. It also demonstrates that the SM2200 significantly reduces the peak as well as PAR. When using the constant carrier amplitude mode it is important to understand how the output signal is affected by changing the amount of carriers that are transmitting. Figure 29 shows a detailed graph of the increase of the peak voltage when compared to a single carrier. This is provided as a reference.

A good understanding of the peak voltage will help to achieve the best communications performance. Two aspects must be considered when choosing an output voltage. Clipping or distortion due to either the output voltage exceeding an amplifier range or the DAC’s output scale being exceeded. If the output voltage is less than the capacity of the amplifier and DAC, there will be less signal injected into the power line and therefore communications distance is less than the optimum.

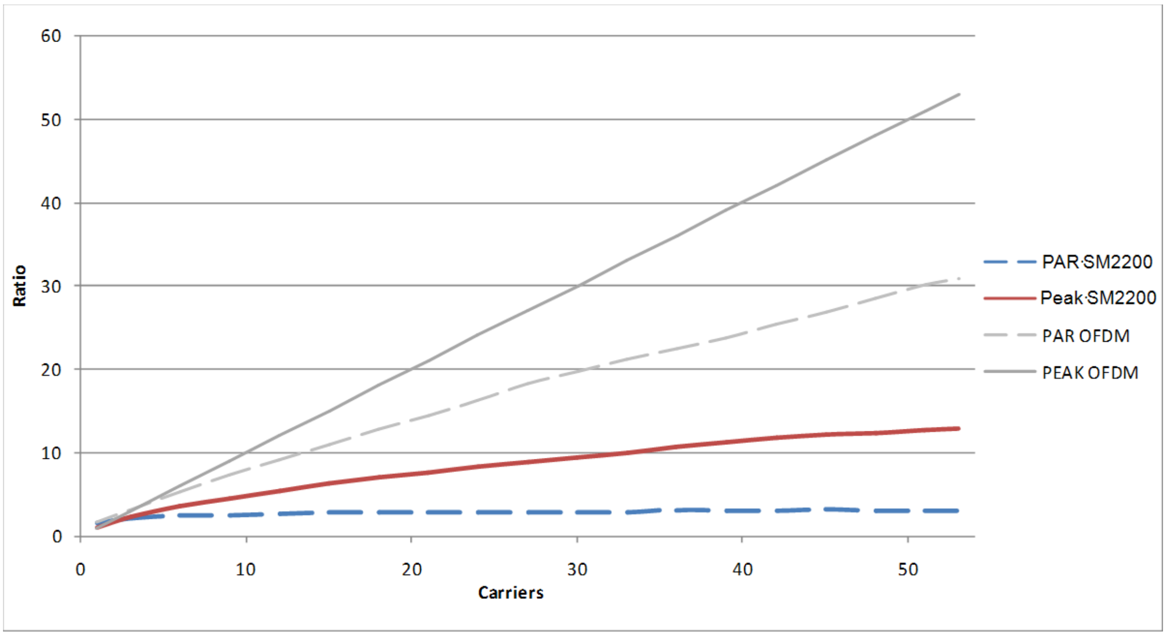


Figure 28 - The effect of adding carriers on the PAR and peak voltage

When the Tx regulation is off (i.e. 0x56[2] = ’1’), the output of the transmitter is unregulated. The SM2200 is in the constant carrier amplitude mode. The number of carriers transmitted and levels directly affect the output voltage.

The peak to peak output voltage for a single carrier can be calculated using the following formula.

$V_{carrier\ p\text{-}p}$ = 0.003* TXV
Where TXV is the value in the TX Level register (0x3C)

For example, if the TX Level register (0x3C) is set to 256, then the output voltage will be 768mV peak to peak.

When using TXREG OFF mode it is important to understand how the output signal is affected by changing the amount of carriers that are transmitting. Figure 29 shows a detailed graph of the increase of the peak voltage when compared to a single carrier. This is provided as reference.

The maximum TXV can roughly calculated based on $V_{carrier\ p\text{-}p}$ = 0.003* TXV and peak ratio shown in Figure 29. Take the case of 6 channels at 1BPSK (i.e. 6 carriers) for example. From Figure 29, the peak ratio for 6 carriers is 3.5, therefore the peak to peak output voltage for 6 carrier $V_{carrier\ p\text{-}p}$ is $V_{MAX\ total\ p\text{-}p}$/peak ratio = 1.53V/3.5 = 0.437V. The maximum transmit power for 6 carriers TXV = $V_{carrier\ p\text{-}p}$/0.003 = 0.437/0.003 = 146.

When Tx regulation is off, TXV value has to be carefully taken to ensure that the output of the DAC does not exceed the maximum range and the amplifier does not enter a non-linear region otherwise communication reliability will be affected. The recommended maximum TXV value vs. number of carriers can be found in Table 8.

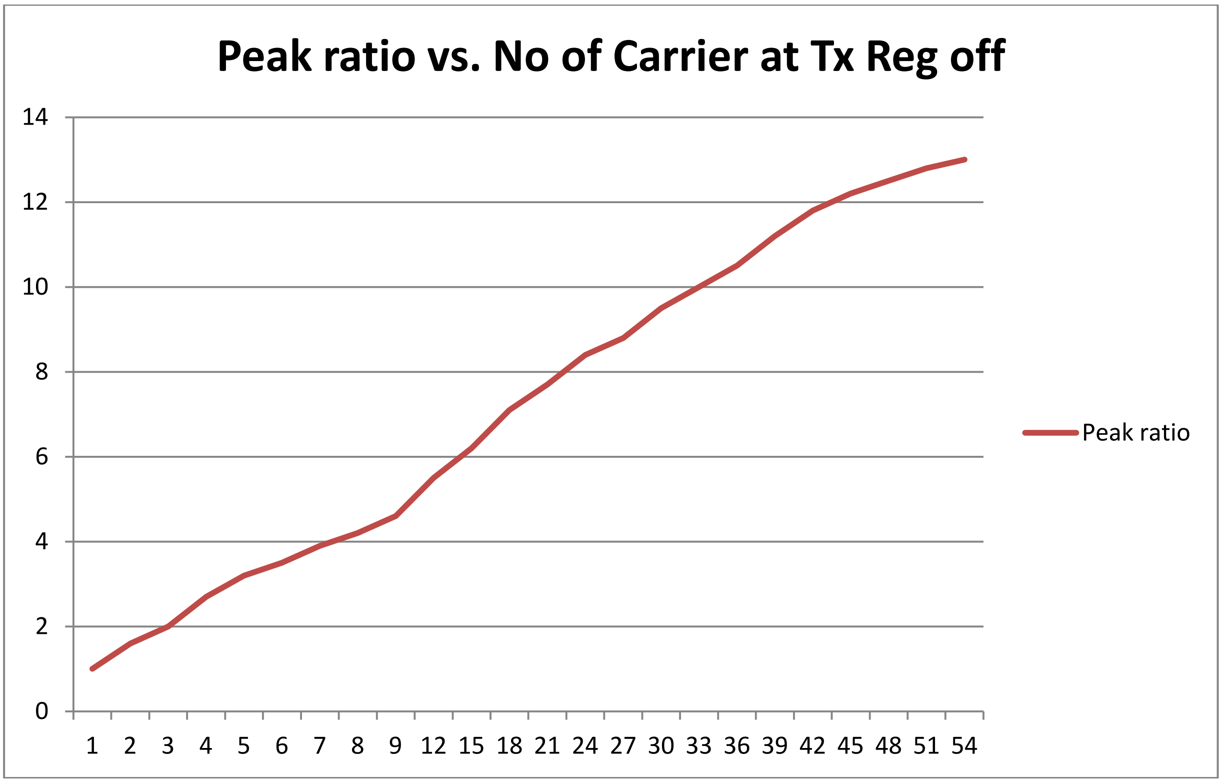


Figure 29 - Detailed graph showing the peak ratio increase compared to a single carrier

Table 8 Recommended TXV value vs. number of carriers when Tx Reg is off

No of Carriers	1	2	3	4	5	6	7	8	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54
Peak ratio	1	1.6	2	2.7	3.2	3.5	3.9	4.2	4.6	5.5	6.2	7.1	7.7	8.4	8.8	9.5	10	10.5	11.2	11.8	12.2	12.5	12.8	13.0
Total Vpp	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53
Voltage per carrier	1.53	0.96	0.77	0.57	0.48	0.44	0.39	0.36	0.33	0.28	0.25	0.22	0.2	0.18	0.17	0.16	0.15	0.15	0.14	0.13	0.13	0.12	0.12	0.12
Tx Power	510	319	255	189	159	146	131	121	111	92.7	82.3	71.8	66.2	60.7	58	53.7	51	48.6	45.5	43.2	41.8	40.8	39.8	39.2

3.5.2 Constant Transmission Voltage Mode (TXREG ON)

Unlike the TXREG OFF mode the TXREG ON (i.e. Constant Transmission Voltage) mode tries to regulate the output voltage so that a constant output voltage peak is always transmitted. This mode means that the output voltage does not need to be calculated before the packet is transmitted. Same as the TXREG OFF mode the output voltage can be adjusted using the TXV[8:0] bits in the TX Level register (0x3C). The scale in which it operates is the same to that of the previous mode. The output voltage can be calculated using the following formula.

$V_{out\ p\text{-}p} = 0.003 * \text{TXV}$

The regulation affects the amplitude of each carrier as the number of carrier is changed. This is also the case when the individual carrier levels are changed. Figure 30 shows the reduction in carrier amplitude as the number of carriers is increased.

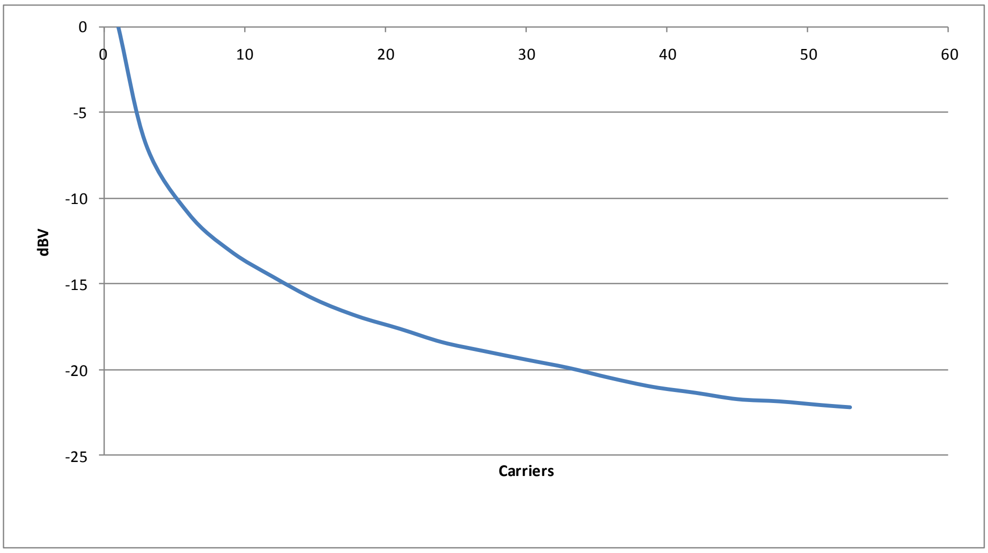


Figure 30 - Carrier amplitude reduction in dBV vs. number of carriers

For example, the maximum total output signal $V_{MAX\ total\ p\text{-}p} = 0.003 * \text{TXV}_{MAX} = 0.003 * 511 = 1.53\text{V}$.

- For 1 carrier (1 channel at 1BPSK)
  From Figure 29, the peak ratio for 1 carrier is 1, therefore the peak to peak output voltage for 1 carrier $V_{carrier\ p\text{-}p} = V_{MAX\ total\ p\text{-}p}$ /peak ratio = 1.53 /1 = 1.53V,

- For 54 carriers (18 channels at 3BPSK or 3QPSK)
  the peak ratio for 54 carriers is 11.3, therefore the peak to peak output voltage for 54 carrier $V_{carrier\ p\text{-}p} = V_{MAX\ total\ p\text{-}p}$ /peak ratio = 1.53 /11.3 = 0.14V.

Amplitude reduction must be taken into account when transmitting data. To achieve the maximum communications distances a minimum of carriers must be used.

When Tx regulation is on, it is recommended to disable the bad channels and use only clean channels for communication. By using less channels, there is more energy (higher transmit level) available per active channel. Figure 31 shows an example of how the spectrum and amplitude changes as the number of carriers is increased from 1 to 54.

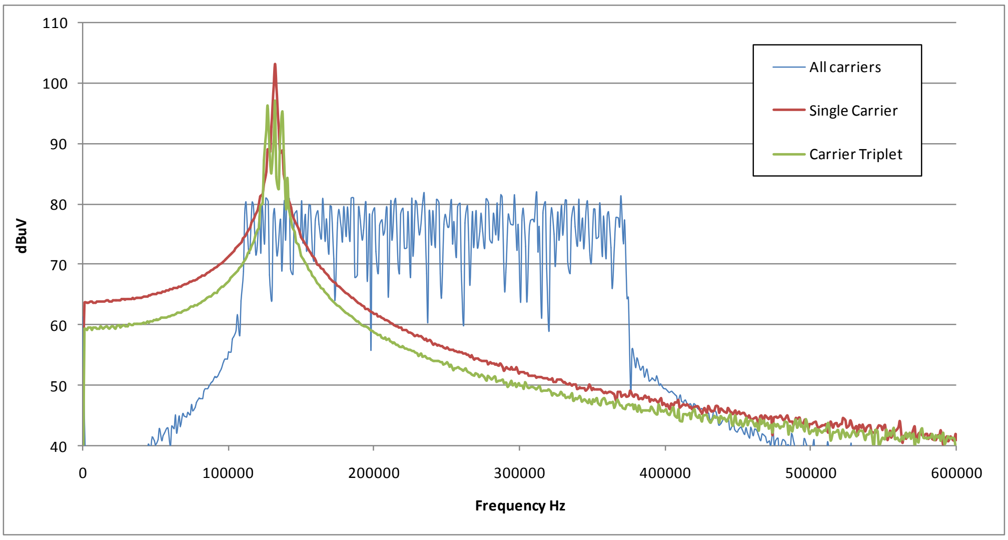


Figure 31 - Example of carrier amplitude for 1, 3 and 54 carriers enabled.

The carrier peak to peak output voltage vs. number of carriers is shown in Table 9.

Table 9 Carrier voltage vs. number of carriers when Tx Reg is on

No of Carriers	**1**	**3**	**6**	**9**	**12**	**15**	**18**	**21**	**24**	**27**	**30**	**33**	**36**	**39**	**42**	**45**	**48**	**51**	**54**
Peak ratio	1	1.6	2	2.7	3.2	3.5	3.9	4.2	4.6	5.5	6.2	7.1	7.7	8.4	8.8	9.5	10	10.5	11.2
Total Vpp	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53
Voltage per carrier	1.53	0.96	0.77	0.57	0.48	0.44	0.39	0.37	0.33	0.28	0.25	0.22	0.2	0.18	0.17	0.16	0.15	0.15	0.14
Tx Power	511	511	511	511	511	511	511	511	511	511	511	511	511	511	511	511	511	511	511

3.6 Data Bandwidth

SM2200 can operate on two frequency bands: 250 kHz and 500 kHz. The frequency band of operation is determined by BWSEL pin (pin 26) as shown in the table below. When the SM2200 operates in the 250 kHz band, the internal clock is halved resulting in a reduction of power consumption and throughput.

BWSEL	Frequency Band
‘0’	500 kHz
‘1’	250 kHz

The BWSEL pin must be set before de-asserting the reset pin.

3.7 Addressing

Often on a shared medium network there are a lot of packets that are not intended for a particular node. In order to address this, SM2200 provides an address filtering to reduce the load on the host processor and save power from processing packets that are not intended for the node.

There are four registers used to store the node address, i.e. Node Address Registers 0, 1, 2 and 3. These registers are all 16 bits giving a maximum addressing range of 64 bits. There are also four address mask registers, i.e. Node Address Mask register 0, 1, 2 and 3. The address mask registers allow the SM2200 to receive a range of address to facilitate routing.

The address filtering implements a TCP-IP style address mask. Example: If we have a 16bit address we set the ALEN = 0x02 (two bytes). We can have a section of the address as a domain i.e.:

16 bit address	
8bit Domain	8bit Node Address
0x30	0xAA

Therefore we could have a domain of 0x30 and a node address of 0xAA. If we only want to receive packets for the address 0x30AA, we set the mask to 0xFFFF. If we want to turn the node into a router for the domain, we can set the mask to 0xFF00. This means we will receive any packet addressed to the domain.

If you do not want to use the address filtering, you can set the ALEN = 0. If address filtering is used, the address has to be at the start of the packet as shown below:

Address	Payload

3.8 Band In Use

The band in use indication can be used to detect if the channel is suitable to be used for transmission. This is primarily used for implementing some parts of medium access control or MAC sub layer. When the channel reaches the band in use threshold set in the BIU Threshold register, the BIU Status register for that particular channel will be set. Noise can also trigger the BIU status and the BIU Qualification on the Physical Control Register can be used to add additional filtering for BIU detection.

3.9 Noise Measurement

SM2200 is capable of estimating the noise present on each channel. This noise reading is averaged heavily due to the unpredictable fluctuations of noise on the power line. The reading is averaged over time and is accessible on the Channel Noise register. A detailed sample code of measuring local noise is shown in the *SM2200_Receive_Packet.c.*

3.10 RSSI Measurement

Each packet that the SM2200 receive includes an RSSI measurement. This measurement is 2 bytes long and is always appended at the end of the packet. There is no specific register associated with the RSSI since it is always present within the packet. Although the RSSI is an indication of how “strong” the signal is, it should be used along with the noise measurement to determine signal quality.

3.11 Current Consumption

The SM2200 transceiver chip and its associated power amplifier circuitry are powered by user-supplied +12 to +15VDC power supplies. The current consumption of SM2200 at 3.3V depends on the settings of the SM2200. Following are the typical current consumption of SM2200 at some settings.

Mode	**Device Current @3.3V (mA)**	**Note**
Shutdown	12.08	Shutdown clock and analogue inputs
Reset	16.66	
Receive Mode	37.47	- Only Channel 0 enabled - BWSEL = 1
Transmit Mode	49.70	- Transmits on all channels every few seconds - Transmission Voltage Level @ 0x80 - All 18 channels enabled - Channel frequency set from 53KHz to 317KHz - Carrier voltage level un-attenuated - Buffer mode is one-to-one - Packet Type : QPSK - Packet Size: 20bytes - BWSEL = 0

Notes:

- Shutdown mode is actually PLL/Oscillator disable mode. The shutdown/PLL disable sequence is asserting the reset, and then asserting shutdown.
- Putting SM2200 in Shutdown mode periodically or receiving on less channels with BWSEL=1 to save the power at idle state.
- Figures based on testing from the SM9222 Evaluation and Development board
- Figures above are for SM2200 device only. The current consumed by the external circuitry is not included.

4. Contact Information

For more information regarding the *SM2200* including technical data sheets, application notes, sample enquiries, demonstration modules, pricing and ordering please contact:

Semitech Semiconductor Pte Ltd
www.semitechsemi.com
sales@semitechsemi.com

5. Revision (22-08)

Version	Description	Date
0.01	Draft	03/08/2011
1.01	Initial release	08/08/2011
1.03	Updated format	16/08/2011
1.04	Updated format	12/09/2011
1.05	Updated format	30/01/2012
1.06	Updated the heading numbers and fixed the broken links; removed some internal info to avoid confusion	08/02/2012
1.07	Added read mode for 0x26 register	13/02/2012
1.08	Clarified Local LB (i.e. Analog LB) and Internal LB (i.e. Digital LB)	24/02/2012
1.09	Fixed the error links	27/06/2012
1.10	Added more details to SPI registers	19/07/2012
2.01	Major updates	26/08/2012
2.02	Added packet size details. Changed SM2200 Init flowchart	05/09/2012
2.03	Add SPI header and current consumption info	24/05/2013
2.04	Matched the names on Page 30	11/09/2013