CC112X Low-Power High Performance Sub-1 GHz RF Transceivers

User's Guide

This product shall not be used in any of the following products or systems without prior express written permission from Texas Instruments:

(i)	*implantable cardiac rhythm management systems, including without limitation pacemakers, defibrillators and cardiac resynchronization devices,*
(ii)	*external cardiac rhythm management systems that communicate directly with one or more implantable medical devices; or*
(iii)	*other devices used to monitor or treat cardiac function, including without limitation pressure sensors, biochemical sensors and neurostimulators.*

Please contact lpw-medical-approval@list.ti.com if your application might fall within the category described above.

Abbreviations

Abbreviations used in this data sheet are described below.

2-FSK	Binary Frequency Shift Keying	MCU	Microcontroller Unit
4-FSK	Quaternary Frequency Shift Keying	MSB	Most Significant Bit
ACP	Adjacent Channel Power	MSK	Minimum Shift Keying
ADC	Analog to Digital Converter	OOK	On-Off Keying
AFC	Automatic Frequency Compensation	PA	Power Amplifier
AGC	Automatic Gain Control	PD	Power Down
ASK	Amplitude Shift Keying	PER	Packet Error Rate
BT	Bandwidth-Time Product	PLL	Phase Locked Loop
CCA	Clear Channel Assessment	POR	Power-On Reset
CRC	Cyclic Redundancy Check	PQI	Preamble Quality Indicator
CS	Carrier Sense	PQT	Preamble Quality Threshold
DC	Direct Current	QPSK	Quadrature Phase Shift Keying
ESR	Equivalent Series Resistance	RC	Resistor-Capacitor
FCC	Federal Communications Commission	RF	Radio Frequency
FIFO	First-In-First-Out	RSSI	Received Signal Strength Indicator
FHSS	Frequency Hopping Spread Spectrum	RX	Receive, Receive Mode
FS	Frequency Synthesizer	SPI	Serial Peripheral Interface
GFSK	Gaussian shaped Frequency Shift Keying	SRD	Short Range Devices
IF	Intermediate Frequency	TBD	To Be Defined
I/Q	In-Phase/Quadrature	TX	Transmit, Transmit Mode
ISM	Industrial, Scientific, Medical	VCO	Voltage Controlled Oscillator
kbps	kilo bit per second	eWOR	Enhanced Wake on Radio
ksps	Kilo symbol per second	XOSC	Crystal Oscillator
LNA	Low Noise Amplifier	XTAL	Crystal
LO	Local Oscillator		
LSB	Least Significant Bit		
LQI	Link Quality Indicator		

Table of Contents

ABBREVIATIONS 2
TABLE OF CONTENTS 3
1 OVERVIEW 5
2 CONFIGURATION SOFTWARE 6
3 MICROCONTROLLER INTERFACE 6
3.1 CONFIGURATION 6
3.2 REGISTER ACCESS 8
3.3 OPTIONAL PIN CTRL RADIO CONTROL FEATURE 15
3.4 GENERAL PURPOSE / TEST OUTPUT CONTROL PINS 16
4 COMMON RECEIVE AND TRANSMIT CONFIGURATION 20
4.1 MODULATION FORMATS 20
4.2 DATA RATE PROGRAMMING 23
5 RECEIVE CONFIGURATION 24
5.1 RECEIVER CHANNEL FILTER BANDWIDTH 24
5.2 DC OFFSET REMOVAL 25
5.3 AUTOMATIC GAIN CONTROL 25
5.4 DEMODULATOR, SYMBOL SYNCHRONIZER, AND DATA DECISION 26
5.5 FREQUENCY OFFSET COMPENSATION 26
5.6 IMAGE COMPENSATION 27
5.7 BIT SYNCHRONIZATION 27
5.8 BYTE SYNCHRONIZATION, SYNC WORD DETECTION 28
5.9 RECEIVED SIGNAL QUALIFIERS AND LINK QUALITY INFORMATION 28
5.10 SYNC WORD QUALIFIER 28
5.11 PREAMBLE DETECTION 28
5.12 COLLISION DETECTOR 29
5.13 RSSI 29
5.14 CARRIER SENSE (CS) 33
5.15 CLEAR CHANNEL ASSESSMENT (CCA) 33
5.16 LISTEN BEFORE TALK (LBT) 34
5.17 LINK QUALITY INDICATOR (LQI) 34
6 TRANSMIT CONFIGURATION 35
6.1 PA OUTPUT POWER PROGRAMMING 35
6.2 OOK/ASK BIT SHAPING 35
7 PACKET HANDLING HARDWARE SUPPORT 36
7.1 STANDARD PACKET FORMAT 36
7.2 PACKET FILTERING IN RECEIVE MODE 40
7.3 PACKET HANDLING IN TRANSMIT MODE 41
7.4 PACKET HANDLING IN RECEIVE MODE 41
7.5 PACKET HANDLING IN FIRMWARE 42
7.6 TX FIFO AND RX FIFO 42
7.7 TRANSPARENT AND SYNCHRONOUS SERIAL OPERATION 43
8 RADIO CONTROL 44
8.1 POWER-ON START-UP SEQUENCE 45
8.2 CRYSTAL CONTROL 45
8.3 VOLTAGE REGULATOR CONTROL 45
8.4 ACTIVE MODES 46
8.5 RX TERMINATION 46
8.6 ENHANCED WAKE ON RADIO (EWOR) 47
8.7 RC OSCILLATOR CALIBRATION 49
8.8 MCU WAKEUP 49
8.9 RX SNIFF MODE 51
8.10 ANTENNA DIVERSITY 52
9 FREQUENCY SYNTHESIZER CONFIGURATION 53
9.1 RF PROGRAMMING 53
9.2 IF PROGRAMMING 54
9.3 FS CALIBRATION 54

9.4 FS OUT OF LOCK DETECTION 55

10 SYSTEM CONSIDERATIONS AND GUIDELINES 55

10.1 VOLTAGE REGULATORS 55

10.2 SRD REGULATIONS 55

10.3 FREQUENCY HOPPING AND MULTI-CHANNEL SYSTEMS 55

10.4 WIDEBAND MODULATION WHEN NOT USING SPREAD SPECTRUM 56

10.5 CONTINUOUS TRANSMISSIONS 56

10.6 BATTERY OPERATED SYSTEMS 56

11 SOLDERING INFORMATION 57

12 DEVELOPMENT KIT ORDERING INFORMATION 57

13 REFERENCES 58

14 GENERAL INFORMATION 59

14.1 DOCUMENT HISTORY 59

1 Overview

CC112X is a family of high performance low power RF transceivers designed for operation with a companion MCU. This user guide describes configurations and functionality to implement a wireless system. **CC112X** automates all common RF related task, greatly offloading the MCU of time critical and processing intensive tasks. Below is the block diagram showing the different parts of the transceiver, divided in an RF related part and a part for digital support functionality.

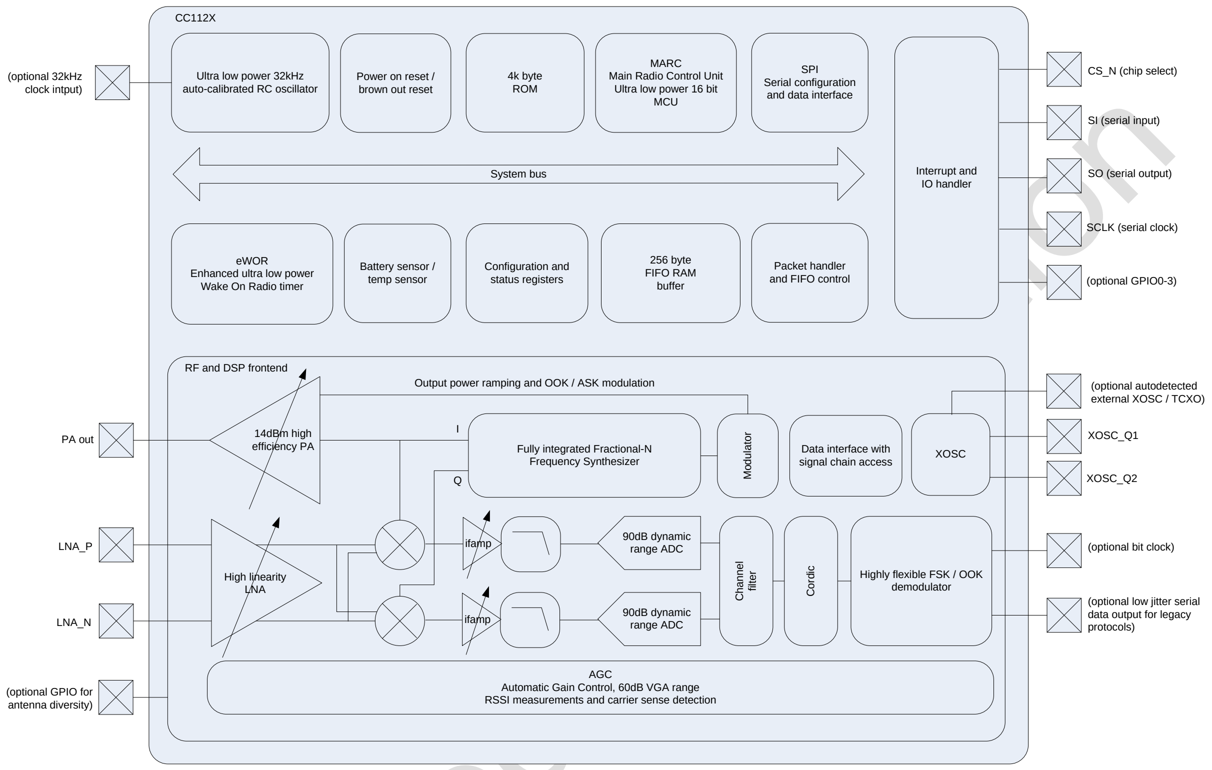

CC112X
(optional 32kHz clock intput)
Ultra low power 32kHz auto-calibrated RC oscillator
Power on reset / brown out reset
4k byte ROM
MARC Main Radio Control Unit Ultra low power 16 bit MCU
SPI Serial configuration and data interface
Interrupt and IO handler
CS_N (chip select)
SI (serial input)
SO (serial output)
SCLK (serial clock)
(optional GPIO0-3)
System bus
eWOR Enhanced ultra low power Wake On Radio timer
Battery sensor / temp sensor
Configuration and status registers
256 byte FIFO RAM buffer
Packet handler and FIFO control
RF and DSP frontend
Output power ramping and OOK / ASK modulation
(optional autodetected external XOSC / TCXO)
PA out
14dBm high efficiency PA
I
Q
Fully integrated Fractional-N Frequency Synthesizer
Modulator
Data interface with signal chain access
XOSC
XOSC_Q1
XOSC_Q2
LNA_P
LNA_N
High linearity LNA
ifamp
ifamp
90dB dynamic range ADC
90dB dynamic range ADC
Channel filter
Cordic
Highly flexible FSK / OOK demodulator
(optional bit clock)
(optional low jitter serial data output for legacy protocols)
(optional GPIO for antenna diversity)
AGC Automatic Gain Control, 60dB VGA range RSSI measurements and carrier sense detection

Figure 1: CC112X Block Diagram

CC112X can be configured to achieve optimum performance for many different applications. Configuration is done using the SPI interface. See Section 3.1.1 for more description of the SPI interface. The following key parameters can be programmed:

Power-down / power up mode
Crystal oscillator power-up / power-down
Receive / transmit mode
RF channel selection
Data rate
Modulation format
RX channel filter bandwidth
RF output power
Data buffering with separate 128-byte receive and transmit FIFOs
Packet radio hardware support
Data whitening
Enhanced Wake-On-Radio (eWOR)

Figure 2 shows a simplified state diagram. For detailed information on controlling the **CC112X** state machine and a complete state diagram, see Section 8.

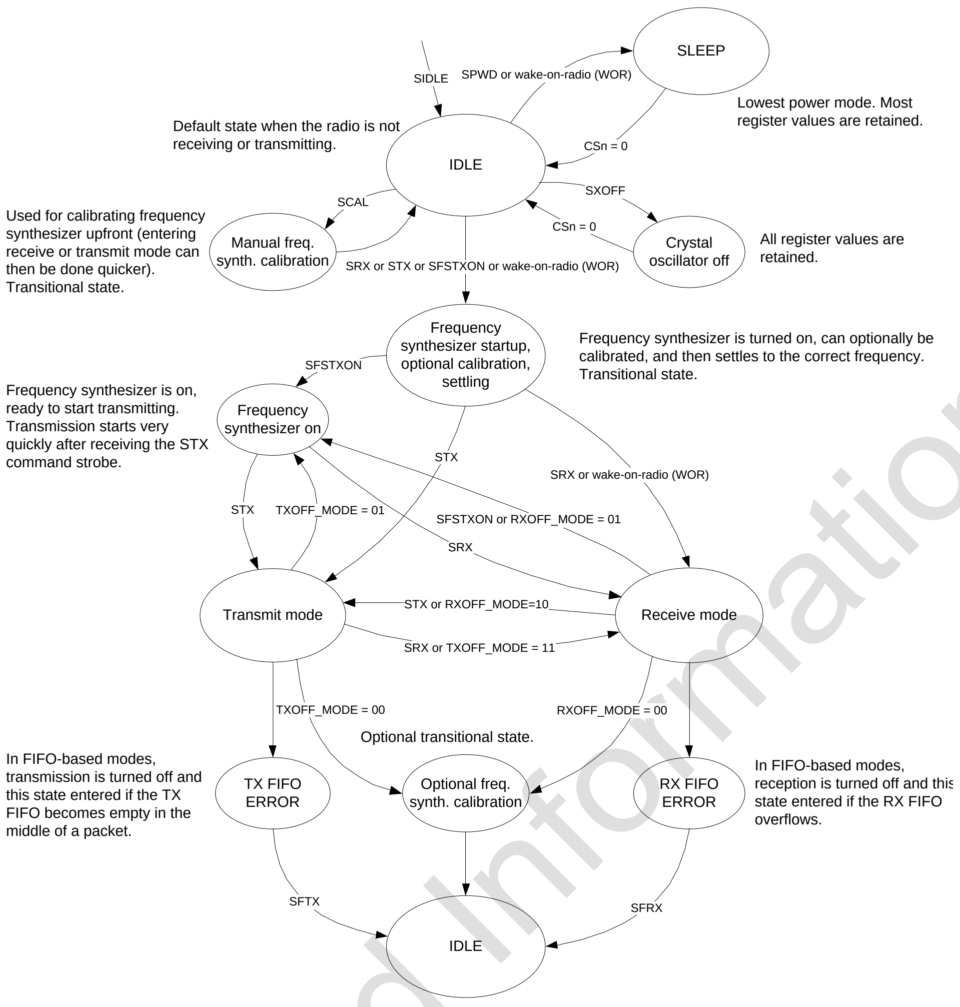

SLEEP
SIDLE
SPWD or wake-on-radio (WOR)
Lowest power mode. Most register values are retained.
Default state when the radio is not receiving or transmitting.
CSn = 0
IDLE
SXOFF
SCAL
Used for calibrating frequency synthesizer upfront (entering receive or transmit mode can then be done quicker). Transitional state.
Manual freq. synth. calibration
CSn = 0
Crystal oscillator off
All register values are retained.
SRX or STX or SFSTXON or wake-on-radio (WOR)
Frequency synthesizer startup, optional calibration, settling
Frequency synthesizer is turned on, can optionally be calibrated, and then settles to the correct frequency. Transitional state.
SFSTXON
Frequency synthesizer is on, ready to start transmitting. Transmission starts very quickly after receiving the STX command strobe.
Frequency synthesizer on
STX
SRX or wake-on-radio (WOR)
STX
TXOFF_MODE = 01
SFSTXON or RXOFF_MODE = 01
SRX
Transmit mode
STX or RXOFF_MODE=10
Receive mode
SRX or TXOFF_MODE = 11
TXOFF_MODE = 00
RXOFF_MODE = 00
Optional transitional state.
In FIFO-based modes, transmission is turned off and this state entered if the TX FIFO becomes empty in the middle of a packet.
TX FIFO ERROR
Optional freq. synth. calibration
RX FIFO ERROR
In FIFO-based modes, reception is turned off and this state entered if the RX FIFO overflows.
SFTX
SFRX
IDLE

Figure 2: Simplified State Diagram

2 Configuration Software

CC112X can be configured using the SmartRF™ Studio software. SmartRF Studio is highly recommended for obtaining optimum register settings, and for evaluating performance and functionality.

After chip reset, all registers have default values and these might differ from the optimum register setting. It is therefore necessary to configure/reconfigure the radio through the SPI interface after the chip has been reset. SmartRF Studio provides a code export function making it easy to implement this in firmware.

3 Microcontroller Interface

3.1 Configuration

In a typical system, CC112X will interface to a microcontroller. This microcontroller must be able to:

- Program CC112X into different modes
- Read and write buffered data
- Read back status information via the 4-wire SPI-bus configuration interface (SI, SO, SCLK and CSn)

3.1.1 4-wire Serial Configuration and Data Interface

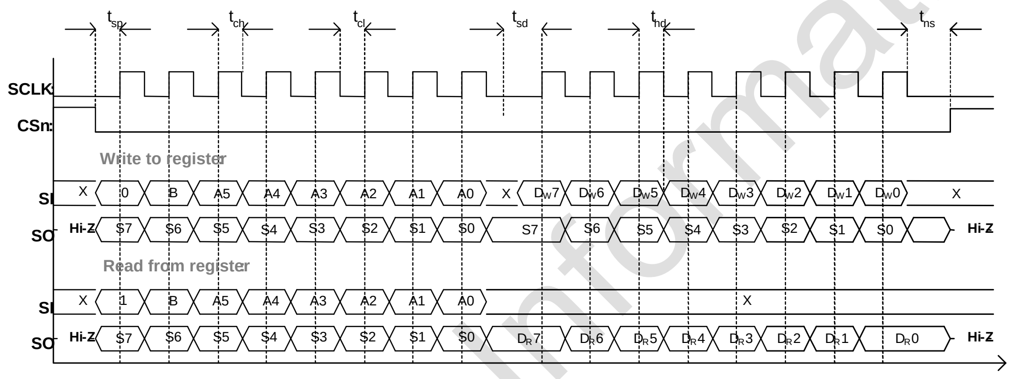
CC112X is configured via a simple 4-wire SPI-compatible interface (SI, SO, SCLK and CSn) where the CC112X is the slave. This interface is also used to read and write buffered data. All transfers on the SPI interface are done most significant bit first.

All transactions on the SPI interface start with a header byte containing a R/$\overline{W}$ bit, a burst access bit (B), and a 6-bit address (A_5 - A_0).

The CSn pin must be kept low during transfers on the SPI bus. If CSn goes high during the transfer of a header byte or during read/write from/to a register, the transfer will be cancelled. The timing for the address and data transfer on the SPI interface is shown in Figure 3 with reference to Table 1.

When CSn is pulled low, the MCU must wait until CC112X SO pin goes low before starting to transfer the header byte. This indicates that the crystal is running. Unless the chip was just reset or was in SLEEP or XOFF state, or the XOSC configuration has been altered, the SO pin will always go low immediately after taking CSn low.

Table 6 gives an overview of the different SPI access types possible.



t_sp
t_ch
t_cl
t_sd
t_hd
t_ns
SCLK
CSn
Write to register
SI
SO
Hi-Z
Read from register
X 0 B A5 A4 A3 A2 A1 A0 X DW7 DW6 DW5 DW4 DW3 DW2 DW1 DW0 X
S7 S6 S5 S4 S3 S2 S1 S0
X 1 B A5 A4 A3 A2 A1 A0 X
DR7 DR6 DR5 DR4 DR3 DR2 DR1 DR0

Figure 3: Configuration Registers Write and Read Operations

Parameter	Description	Min	Max	Units
f_{SCLK}	SCLK frequency read/write access Note: 100 ns delay between consecutive data bytes must be added during burst write access to the configuration registers	-	10	MHz
	SCLK frequency read access extended memory	-	7.7	
t_{sp}	CSn low to positive edge on SCLK	50	-	ns
t_{ch}	Clock high	50	-	ns
t_{cl}	Clock low	50	-	ns
t_{rise}	Clock rise time	-	40	ns
t_{fall}	Clock fall time	-	40	ns
t_{sd}	Setup data before a positive edge on SCLK	10	-	ns
t_{hd}	Hold data after positive edge on SCLK	10	-	ns
t_{ns}	Negative edge on SCLK to CSn high.	200	-	ns
	CSn high time, time from CSn has been pulled high until it can be pulled low again	50		ns

Table 1: SPI Interface Timing Requirements

3.1.2 Chip Status Byte

When the header byte, data byte, or command strobe is sent on the SPI interface, the chip status byte is sent by the CC112X on the SO pin. The status byte contains key status signals, useful for the MCU. The first bit, s7, is the `CHIP_RDYn` signal and this signal must go low before the first positive edge of SCLK. The `CHIP_RDYn` signal indicates that the crystal is running.

Bits 6, 5, and 4 comprise the `STATE` value. This value reflects the state of the chip. The XOSC and power to the digital core are on in the IDLE state, but all other modules are in power down. The frequency and channel configuration should only be updated when the chip is in this state.

The last four bits (3:0) in the status byte contains `FIFO_BYTES_AVAILABLE`. For read operations (the R/$\overline{W}$ bit in the header byte is set to 1), the `FIFO_BYTES_AVAILABLE` field contains the number of bytes available for reading from the RX FIFO. For write operations (the R/$\overline{W}$ bit in the header byte is set to 0), the `FIFO_BYTES_AVAILABLE` field contains the number of bytes that can be written to the TX FIFO. When `FIFO_BYTES_AVAILABLE = 15`, 15 or more bytes are available/free. Table 2 gives a status byte summary.

Bits	Name	Description		
7	CHIP_RDYn	Stays high until power and crystal have stabilized. Should always be low when using the SPI interface.		
6:4	STATE[2:0]	Indicates the current main state machine mode		
		Value	State	Description
		000	IDLE	IDLE state
		001	RX	Receive mode
		010	TX	Transmit mode
		011	FSTXON	Fast TX ready
		100	CALIBRATE	Frequency synthesizer calibration is running
		101	SETTLING	PLL is settling
		110	RXFIFO ERROR	RX FIFO has over / underflowed. Read out any useful data, then flush the FIFO with `SFRX` strobe
		111	TXFIFO ERROR	TX FIFO has over / underflowed. Flush with `SFTX` strobe
3:0	FIFO_BYTES_AVAILABLE[3:0]	The number of bytes available in the RX FIFO or free bytes in the TX FIFO		

Table 2: Status Byte Summary

3.2 Register Access

The configuration registers on the CC112X are located on SPI addresses from 0x00 to 0x2E with address extension command at address 0x2F to access the extended register space. It is highly recommended to use SmartRF Studio to generate optimum register settings. All configuration registers can be both written to and read. The R/$\overline{W}$ bit controls if the register should be written to or read. When writing to registers, the status byte is sent on the SO pin each time a header byte or data byte is transmitted on the SI pin. When reading from registers, the status byte is sent on the SO pin each time a header byte is transmitted on the SI pin.

Registers with consecutive addresses can be accessed in an efficient way by setting the burst bit (B) in the header byte. The address bits (A_5 - A_0) set the start address in an internal address counter. This counter is incremented by one each new byte (every 8 clock pulses). The burst access is either a read or a write access and must be terminated by setting CSn high.

If a single register shall be accessed multiple times (e.g. soft RX / TX data register for external modulation / demodulation on MCU / DSP), the `EXT_CTRL.BURST_ADDR_INCR_EN` bit can be set to 0. In this mode the address counter will not increment in burst mode, and it is possible to read / write the same register repeatedly without address overhead.

Figure 4 shows the SPI memory map, Table 3 shows the SPI address space, and Table 4 shows the extended register space mapping and direct FIOF access mapping.

Note that all registers in register space (address 0x00 - 0x2E) have retention. The FIFOs and the status registers located in extended register space do not have retention (please see Table 4 for details).

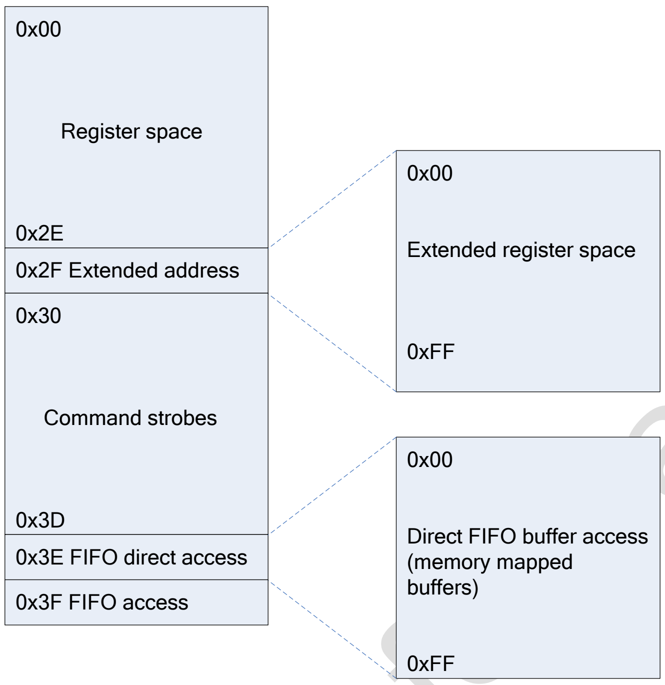


Figure 4: SPI Memory Map

	Write		Read		
	Single Byte	Burst	Single Byte	Burst	
	+0x00	+0x40	+0x80	+0xC0	
0x00	IOCFG3				R/W configuration registers, burst access possible
0x01	IOCFG2				
0x02	IOCFG1				
0x03	IOCFG0				
0x04	SYNC3				
0x05	SYNC2				
0x06	SYNC1				
0x07	SYNC0				
0x08	SYNC_CFG1				
0x09	SYNC_CFG0				
0x0A	DEVIATION_M				
0x0B	MODCFG_DEV_E				
0x0C	DCFILT_CFG				
0x0D	PREAMBLE_CFG1				
0x0E	PREAMBLE_CFG0				
0x0F	FREQ_IF_CFG				
0x10	IQIC				
0x11	CHAN_BW				
0x12	MDMCFG1				
0x13	MDMCFG0				
0x14	DRATE2				
0x15	DRATE1				
0x16	DRATE0				
0x17	AGC_REF				
0x18	AGC_CS_THR				
0x19	AGC_GAIN_ADJUST				
0x1A	AGC_CFG3				
0x1B	AGC_CFG2				
0x1C	AGC_CFG1				
0x1D	AGC_CFG0				
0x1E	FIFO_CFG				
0x1F	DEV_ADDR				
0x20	SETTLING_CFG				
0x21	FS_CFG				
0x22	WOR_CFG1				
0x23	WOR_CFG0				
0x24	WOR_EVENT0_MSB				
0x25	WOR_EVENT0_LSB				
0x26	PKT_CFG2				
0x27	PKT_CFG1				
0x28	PKT_CFG0				
0x29	RFEND_CFG1				
0x2A	RFEND_CFG0				
0x2B	PA_CFG2				
0x2C	PA_CFG1				
0x2D	PA_CFG0				
0x2E	PKT_LEN				
0x2F	EXTENDED MEMORY ACCESS				
0x30	SRES		SRES		Command Strobes
0x31	SFSTXON		SFSTXON		
0x32	SXOFF		SXOFF		
0x33	SCAL		SCAL		
0x34	SRX		SRX		
0x35	STX		STX		
0x36	SIDLE		SIDLE		
0x37	SAFC		SAFC		
0x38	SWOR		SWOR		
0x39	SPWD		SPWD		
0x3A	SFRX		SFRX		
0x3B	SFTX		SFTX		
0x3C	SWORRST		SWORRST		
0x3D	SNOP		SNOP		
0x3E	DIRECT FIFO ACCESS				
0x3F	TX FIFO	TX FIFO	RX FIFO	RX FIFO	

Table 3: SPI Address Space

Extended Register Space (0x00 - 0x29)		Retention
0x00	IF_MIX_CFG	Yes
0x01	FREQOFF_CFG	Yes
0x02	TOC_CFG	Yes
0x03	MARC_SPARE	Yes
0x04	ECG_CFG	Yes
0x05	SOFT_TX_DATA_CFG	Yes
0x06	EXT_CTRL	Yes
0x07	RCCAL_FINE	Yes
0x08	RCCAL_COARSE	Yes
0x09	RCCAL_OFFSET	Yes
0x0A	FREQOFF1	Yes
0x0B	FREQOFF0	Yes
0x0C	FREQ2	Yes
0x0D	FREQ1	Yes
0x0E	FREQ0	Yes
0x0F	IF_ADC2	Yes
0x10	IF_ADC1	Yes
0x11	IF_ADC0	Yes
0x12	FS_DIG1	Yes
0x13	FS_DIG0	Yes
0x14	FS_CAL3	Yes
0x15	FS_CAL2	Yes
0x16	FS_CAL1	Yes
0x17	FS_CAL0	Yes
0x18	FS_CHP	Yes
0x19	FS_DIVTWO	Yes
0x1A	FS_DSM1	Yes
0x1B	FS_DSM0	Yes
0x1C	FS_DVC1	Yes
0x1D	FS_DVC0	Yes
0x1E	FS_LBI	Yes
0x1F	FS_PFD	Yes
0x20	FS_PRE	Yes
0x21	FS_REG_DIV_CML	Yes
0x22	FS_SPARE	Yes
0x23	FS_VCO4	Yes
0x24	FS_VCO3	Yes
0x25	FS_VCO2	Yes
0x26	FS_VCO1	Yes
0x27	FS_VCO0	Yes
0x28	GBIAS6	Yes
0x29	GBIAS5	Yes

Extended Register Space (0x2A - 0x75)		Retention
0x2A	GBIAS4	Yes
0x2B	GBIAS3	Yes
0x2C	GBIAS2	Yes
0x2D	GBIAS1	Yes
0x2E	GBIAS0	Yes
0x2F	IFAMP	Yes
0x30	LNA	Yes
0x31	RXMIX	Yes
0x32	XOSC5	Yes
0x33	XOSC4	Yes
0x34	XOSC3	Yes
0x35	XOSC2	Yes
0x36	XOSC1	Yes
0x37	XOSC0	Yes
0x38	ANALOG_SPARE	Yes
0x39	PA_CFG3	Yes
0x3A	Not Used	
0x3B	Not Used	
0x3C	Not Used	
0x3D	Not Used	
0x3E	Not Used	
0x3F	IRQ0M	Yes
0x40	IRQ0F	Yes
…	Not Used	
0x64	WOR_TIME1	No
0x65	WOR_TIME0	No
0x66	WOR_CAPTURE1	No
0x67	WOR_CAPTURE0	No
0x68	BIST	No
0x69	DCFILTOFFSET_I1	No
0x6A	DCFILTOFFSET_I0	No
0x6B	DCFILTOFFSET_Q1	No
0x6C	DCFILTOFFSET_Q0	No
0x6D	IQIE_I1	No
0x6E	IQIE_I0	No
0x6F	IQIE_Q1	No
0x70	IQIE_Q0	No
0x71	RSSI1	No
0x72	RSSI0	No
0x73	MARCSTATE	No
0x74	LQI_VAL	No
0x75	PQT_SYNC_ERR	No

Extended Register Space (0x76 - 0x91)		Retention
0x76	DEM_STATUS	No
0x77	FREQOFF_EST1	No
0x78	FREQOFF_EST0	No
0x79	AGC_GAIN3	No
0x7A	AGC_GAIN2	No
0x7B	AGC_GAIN1	No
0x7C	AGC_GAIN0	No
0x7D	SOFT_RX_DATA_OUT	No
0x7E	SOFT_TX_DATA_IN	No
0x7F	ASK_SOFT_RX_DATA	No
0x80	RNDGEN	No
0x81	MAGN2	No
0x82	MAGN1	No
0x83	MAGN0	No
0x84	ANG1	No
0x85	ANG0	No
0x86	CHFILT_I2	No
0x87	CHFILT_I1	No
0x88	CHFILT_I0	No
0x89	CHFILT_Q2	No
0x8A	CHFILT_Q1	No
0x8B	CHFILT_Q0	No
0x8C	GPIO_STATUS	No
0x8D	FSCAL_CTRL	No
0x8E	PHASE_ADJUST	No
0x8F	PARTNUMBER	No
0x90	PARTVERSION	No
0x91	SERIAL_STATUS	No

Extended Register Space (0x92 - 0xD9)		Retention
0x92	RX_STATUS	No
0x93	TX_STATUS	No
0x94	MARC_STATUS1	No
0x95	MARC_STATUS0	No
0x96	PA_IFAMP_TEST	No
0x97	FSRF_TEST	No
0x98	PRE_TEST	No
0x99	PRE_OVR	No
0x9A	ADC_TEST	No
0x9B	DVC_TEST	No
0x9C	ATEST	No
0x9D	ATEST_LVDS	No
0x9E	ATEST_MODE	No
0x9F	XOSC_TEST1	No
0xA0	XOSC_TEST0	No
…	Not Used	
0xD2	RXFIRST	No
0xD3	TXFIRST	No
0xD4	RXLAST	No
0xD5	TXLAST	No
0xD6	NUM_TXBYTES	No
0xD7	NUM_RXBYTES	No
0xD8	FIFO_NUM_TXBYTES	No
0xD9	FIFO_NUM_RXBYTES	No
Direct FIFO Access Mapping		
0x00 - 0x7F	TXFIFO	Yes
0x80 - 0xFF	RXFIFO	Yes

Table 4: Extended Register Space Mapping and Direct FIFO Access Mapping

3.2.1 Command Strobes

Command Strobes may be viewed as single byte instructions to **CC112X**. By addressing a command strobe register, internal sequences will be started. These commands are used to enable receive mode, enable eWOR, disable the crystal oscillator, etc. The command strobes are listed in Table 5.

Note: An SIDLE strobe will clear all pending command strobes until IDLE state is reached. This means that if for example an SIDLE strobe is issued while the radio is in RX state, any other command strobes issued before the radio reaches IDLE state will be ignored.

The command strobe registers are accessed by transferring a single header byte (no data is being transferred). That is, only the R/$\overline{W}$ bit, the burst access bit (set to 0), and the six address bits (in the range 0x30 through 0x3D) are written. The R/$\overline{W}$ bit can be either one or zero and will determine how the `FIFO_BYTES_AVAILABLE` field in the status byte should be interpreted (see Section 3.1.2). When writing command strobes, the status byte is sent on the SO pin.

A command strobe may be followed by any other SPI access without pulling CSn high, and the command strobes are executed immediately. This applies for all command strobes except `SRES`, `SPWD`, `SWOR`, and the `SXOFF` strobe.

When a `SRES` strobe is issued the CSn pin must be kept low and wait for SO to go low again before the next header byte can be issued, as shown in Figure 5:.

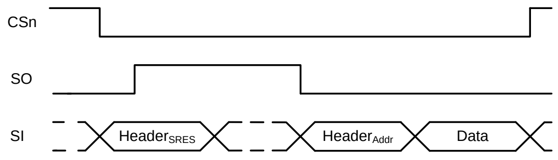


Figure 5: SRES Command Strobe

The `SPWD`, `SWOR`, and the `SXOFF` command strobes are not executed before the CSn goes high.

Address	Strobe Name	Description
0x30	SRES	Reset chip.
0x31	SFSTXON	Enable and calibrate frequency synthesizer (if `SETTLING_CFG.FS_AUTOCAL = 1`). If in RX (with CCA): Go to a wait state where only the synthesizer is running (for quick RX / TX turnaround).
0x32	SXOFF	Turn off crystal oscillator.
0x33	SCAL	Calibrate frequency synthesizer and turn it off. `SCAL` can be strobed from IDLE mode without setting manual calibration mode (`SETTLING_CFG.FS_AUTOCAL = 0`)
0x34	SRX	Enable RX. Perform calibration first if coming from IDLE and `SETTLING_CFG.FS_AUTOCAL = 1.`
0x35	STX	In IDLE state: Enable TX. Perform calibration first if `SETTLING_CFG.FS_AUTOCAL = 1.` If in RX state and CCA is enabled: Only go to TX if channel is clear.
0x36	SIDLE	Exit RX / TX, turn off frequency synthesizer and exit eWOR mode if applicable.
0x37	SAFC	Automatic Frequency Compensation.
0x38	SWOR	Start automatic RX polling sequence (eWOR) as described in Section 8.6 if `WOR_CFG0.RC_PD = 0.`
0x39	SPWD	Enter SLEEP mode when CSn goes high.
0x3A	SFRX	Flush the RX FIFO. Only issue `SFRX` in IDLE or RXFIFO_ERROR states.
0x3B	SFTX	Flush the TX FIFO. Only issue `SFTX` in IDLE or TXFIFO_ERROR states.
0x3C	SWORRST	Reset real time clock to Event1 value.
0x3D	SNOP	No operation. May be used to get access to the chip status byte.

Table 5: Command Strobes

3.2.2 FIFO Access

The 128-byte TX FIFO and the 128-byte RX FIFO are accessed through the 0x3F address. When the R/$\overline{W}$ bit is zero, the TX FIFO is accessed, and the RX FIFO is accessed when the R/$\overline{W}$ bit is one.

Using the standard FIFO push / pop interface, the TX FIFO is write-only, while the RX FIFO is read-only. However, the complete RX and TX FIFOs with associated pointers are mapped in the register space for FIFO manipulation and SW debug purposes. Both FIFO data and pointers are read / writeable to enable e.g. re-transmissions, partial flush, partial readouts, changing only sequence number before re-transmission, etc.

The burst bit is used to determine if the FIFO access is a single byte access or a burst access. The single byte access method expects a header byte with the burst bit set to zero and one data byte. After the data byte, a new header byte is expected; hence, CSn can remain low. The burst access method expects one header byte and then consecutive data bytes until terminating the access by setting CSn high.

Note that the `FIFO_BYTES_AVAILABLE` during a TX FIFO write contains the number of bytes free before writing the byte in progress to the TX FIFO.

The TX FIFO may be flushed by issuing a `SFTX` command strobe. Similarly, a `SFRX` command strobe will flush the RX FIFO. A `SFTX` or `SFRX` command strobe can only be issued in the IDLE, TXFIFO_ERROR, or RXFIFO_ERROR states. Both FIFOs are flushed when going to the SLEEP state. Figure 6 gives a brief overview of different register access types possible.

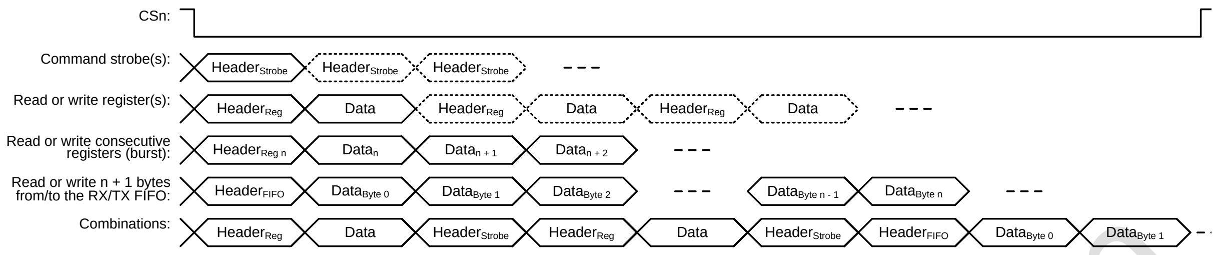


Figure 6: Register Access Types

Access type	Command/Address byte	Description
Single Register Access (register space)	**Address**: R/$\overline{W}$ 0 A_5 A_4 A_3 A_2 A_1 A_0 (A_{5-0} < 0x2F)	The R/$\overline{W}$ bit determines whether the operation is a read (1) or a write (0) operation The register accessed is determined by the address in A_{5-0} Exactly one data byte is expected after the address byte The chip status byte is returned on the SO line both when the address is sent on the SI line as well as when data are written
Burst Register Access (register space)	**Address**: R/$\overline{W}$ 1 A_5 A_4 A_3 A_2 A_1 A_0 (A_{5-0} < 0x2F)	The R/$\overline{W}$ bit determines whether the operation is a read (1) or a write (0) operation The address in A_{5-0} determines the first register accessed, after which an internal address counter is incremented for each new data byte following the address byte Consecutive bytes are expected after the address byte and the burst access is terminated by setting CSn high The chip status byte is returned on the SO line both when the address is sent on the SI line as well as when data is written If the internal address counter reached address 0x2E (last byte in register space) the counter will not increment anymore and the same address will be read / written until the burst access is being terminated
Single Register Access (extended register space)	**Command**: R/$\overline{W}$ 0 1 0 1 1 1 1 **Address**: A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0 (A_{7-0}: See Table 4)	This access mode starts with a specific command (0x2F) The first byte following this command is interpreted as the extended address Exactly one data byte is expected after the extended address byte When the extended address is sent on the SI line, SO will return all zeros. The chip status byte is returned on the SO line when the command is transmitted as well as when data are written to the extended address

Access type	Command/Address byte	Description
Burst Register Access (extended register space)	**Command**: R/$\overline{W}$ 1 1 0 1 1 1 1 **Address**: A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0 (A_{7-0}: See Table 4)	This access mode starts with a specific command (0x2F) The first byte following this command is interpreted as the extended address Consecutive bytes are expected after the extended address byte and the burst access is terminated by setting CSn high When the extended address is sent on the SI line, SO will return all zeros. The chip status byte is returned on the SO line when the command is transmitted as well as when data are written to the extended address. If the internal address counter reached address 0xFF (last byte in extended register space) the counter will wrap around to 0x00 Registers not listed in Table 4 can be part of a burst access
Command Strobe Access	**Address**: R/$\overline{W}$ 0 A_5 A_4 A_3 A_2 A_1 A_0 ($0x30 \leq A_{5-0} \leq 0x3D$)	Accessing one of the command strobe registers triggers an event determined by the address in A_{5-0}, e.g. resetting the device, enabling the crystal oscillator, entering TX, etc. No data byte is expected. The chip status byte is returned on the SO line when a command strobe is sent on the SI line
Direct FIFO Access	**Command**: R/$\overline{W}$ B 1 1 1 1 1 0 **Address**: A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0 $A_{7-0} < 0x80$: TX FIFO $0x80 \leq A_{7-0} \leq 0xFF$: RX FIFO	This access mode starts with a specific command (0x3E) which makes it possible to access the FIFOs directly through memory operations without affecting the FIFO pointers. The first byte following this command is interpreted as the FIFO address. The byte following is read / written to this address. If burst is enabled, consecutive bytes will be read / written by incrementing the address. FIFO pointers are available in extended register space for debug purposes.
Standard FIFO Access	**Address**: R/$\overline{W}$ B 1 1 1 1 1 1	The R/$\overline{W}$ bit determines whether the operation is a read (1) operation from the RX FIFO or a write (0) operation to the TX FIFO. If the burst bit B is 1, all bytes following the address byte are treated as data bytes until CSn goes high. If the burst bit B is 0, the FIFOs are accessed byte-wise as a normal register.

Table 6: SPI Access Types

3.3 Optional PIN CTRL Radio Control Feature

The CC112X has an optional way of controlling the radio by reusing SI, SCLK, and CSn from the SPI interface. This feature allows for a simple three-pin control of the major states of the radio: SLEEP, IDLE, RX, and TX. This optional functionality is enabled with the `EXT_CTRL.PIN_CTRL_EN` configuration bit.

State changes are commanded as follows:

- If CSn is high, the SI and SCLK are set to the desired state according to Table 7.
- WhenIf CSn goes low, the state of SI and SCLK is latched and a command strobe is generated internally according to the pin configuration.

If the device is in the TX state and the TX command is issued, it will be ignored. For RX state an RX command will restart RX. When CSn is low the SI and SCLK have normal SPI functionality.

All pin control command strobes are executed immediately, except the `SPWD` strobe. The `SPWD` strobe is delayed until CSn goes high.

Pin control is useful to get precise timing on RX / TX strobes.

CSn	SCLK	SI	Function
1	X	X	Chip unaffected by SCLK/SI
↓	0	0	Generates SPWD strobe
↓	0	1	Generates STX strobe
↓	1	0	Generates SIDLE strobe
↓	1	1	Generates SRX strobe
0	SPI mode	SPI mode	SPI mode (wakes up into IDLE if in SLEEP/XOFF)

Table 7: Optional Pin Control Coding

3.4 General Purpose / Test Output Control Pins

The four digital input / output pins GPIO0, GPIO1, GPIO2 and GPIO3 are general control pins configured with IOCFG0/1/2/3.GPIO0/1/2/3_CFG. Table 8 shows the different signals that can be monitored on the GPIO pins.

GPIO1 is shared with the SO pin in the SPI interface. The default setting for GPIO1/SO is tri-state output, which is useful when the SPI interface is shared with other devices. By selecting any other of the programming options, the GPIO1/SO pin will become a generic pin when CSn is high and function as SO when CSn is low.

When the IOCFGx.GPIOx_CFG setting is less than 48 and IOCFGx_GPIOx_INV is 0 (1), the GPIO0, GPIO2 and GPIO3 pins will be hardwired to 0 (1), and the GPIO1 pin will be hardwired to 1 (0) in the SLEEP state. These signals will be hardwired until the CHIP_RDYn signal goes low. If the IOCFGx.GPIOx_CFG setting is 48 or higher, the GPIO pins will work as programmed also in SLEEP state.

The GPIOs can also be used as inputs (see Table 10). In this mode the IOCFGx.GPIOx_CFG must be set to HIGHZ (48).

There are three main methods that can be used as inputs / interrupts to the MCU

1. MARC MCU WAKEUP

   MARC (Main Radio Control) handles all RF related operations to greatly reduce the load on the MCU. If the MCU is required to intervene, this signal is pulsed and the MCU can read the MARC_STATUS1.MARC_STATUS_OUT to find cause of wake up event and take appropriate action. MARC MCU wakeup is available through the GPIO pins (IOCFGx.GPIOx_CFG = MARC_MCU_WAKEUP (20)). See Section 8.8 for more details.

2. EVENT IRQ

   CC112X has an interrupt register IRQ0F, which can be masked with the IRQ0M register, which is connected to a single interrupt line. This interrupts signal is available through the GPIO pins (IOCFGx.GPIOx_CFG = EVENT_IRQ (10)). Table 9 shows which interrupts are available through the IRQ0M register.

3. GPIO signals.

   CC112X has a wide range of options to fit various applications. Which GPIO signal that is used can also be changed during operation to suit different needs

Signals in the table below can be read as follows:

One GPIO map name: The signal can be routed out to any of the four GPIO pins for full flexibility

Four GPIO map names: The signal can only be routed out on the GPIO designated in the table. These signals are mainly for debug purposes and not intended for use in normal applications

GPIOx_CFG	Signal Name	Description
0	RXFIFO_THR	Associated to the RX FIFO: Asserts when RX FIFO is filled above `FIFO_CFG.FIFO_THR`. De-asserts when RX FIFO is drained below/equal to `FIFO_THR`.
1	RXFIFO_THR_PKT	Associated to the RX FIFO: Asserts when RX FIFO is filled above `FIFO_THR` or the end of packet is reached. De-asserts when RX FIFO is empty.
2	TXFIFO_THR	Associated to the TX FIFO: Asserts when the TX FIFO is filled above/or equal (127 - `FIFO_THR`). De-asserts when the TX FIFO is below (127 - `FIFO_THR`)
3	TXFIFO_THR_PKT	Associated to the TX FIFO: Asserts when TX FIFO is full. De-asserts when the TX FIFO is drained below (127 - `FIFO_THR`)
4	RXFIFO_OVERFLOW	Asserts when the RX FIFO has overflowed. De-asserts when the RX FIFO has been flushed.
5	TXFIFO_UNDERFLOW	Asserts when the TX FIFO has underflowed. De-asserts when the TX FIFO is flushed.
6	PKT_SYNC_RXTX	For RX: Asserts when sync word has been received, and de-asserts at the end of the packet. Will de-assert when the optional address or length check fails or the RX FIFO overflows/underflows For TX: Asserts when sync word has been sent, and de-asserts at the end of the packet (when the last bit has been sent). Will de-assert if the TX FIFO underflows/overflows.
7	PKT_CRC_OK	Asserts when a packet has been received with OK CRC. De-asserts when the first byte is read from the RX FIFO.
8	SERIAL_CLK	Serial data clock, RX and TX Synchronous to the serial data in synchronous serial mode. Data is set up on the falling edge in RX and is captured on the rising edge of the serial clock in TX.
9	SERIAL_RX	Serial data, RX mode Synchronous serial mode. RX: Data is set up on the falling edge of SERIAL_CLK. Transparent mode: No timing recovery, hard limited signal, 2-ary modulation only
10	EVENT_IRQ	General purpose level interrupt See Section 3.4 and Table 9
11	PQT_REACHED	Preamble Quality Reached. Asserts when the PQI is above the programmed PQT value (sticky)
12	PQT_VALID	Preamble quality valid, asserts when PQT logic has received sufficient number of bits
13	RSSI_VALID	RSSI calculation is valid. (sticky)
14	RSSI Signals 3: RSSI_UPDATE 2: RSSI_UPDATE 1: AGC_HOLD 0: AGC_UPDATE	 3: Pulse 2: Pulse 1: AGC waits for gain settling 0: Pulse

15	Clear Channel Assessment 3: CCA_STATUS 2: TXONCCA_DONE 1: CCA_STATUS 0: TXONCCA_FAILED	 3: Current CCA status 2: Pulse 1: Current CCA status 0: TX on CCA falied
16	CARRIER_SENSE_VALID	
17	CARRIER_SENSE	Carrier sense. High if RSSI level is above threshold.
18	3: DSSS_CLK 2: DSSS_DATA0 1: DSSS_CLK 0: DSSS_DATA1	DSSS (Direct Sequence Spread Spectrum) signals
19	PKT_CRC_OK	Packet received with OK CRC
20	MARC_MCU_WAKEUP	MCU wake up signal generated by MARC. Associated with status register to find cause of wake up event.
21	SYNC_LOW0_HIGH1	Sync detect on high / low part of SFD
22	3: MARC_GPIO_STATE[3] 2: MARC_GPIO_STATE[2] 1: MARC_GPIO_STATE[1] 0: MARC_GPIO_STATE[0]	MARC general purpose IO
23	LNA_PA_REG_PD	Common regulator control for PA and LNA. Indicates RF operation
24	LNA_PD	Control external LNA
25	PA_PD	Control external PA
26	RX0TX1_CFG	Indicates whether RF operation is in RX or TX
27		Reserved - Used for test
28	IMAGE_FOUND	Image detected by image rejection calibration algorithm
29	DEM_CLKEN_SOFT	Data clock for demodulator soft data
30	MOD_SOFT_TX_DATA_CLK	Data clock for modulator soft data
31 - 32		Reserved - Used for test
33	RSSI_STEP_FOUND	Collision indication: RSSI step detected after SYNC found. (Sticky)
34	RSSI_STEP_EVENT	RSSI step detected. (Pulse)
35		Reserved - Used for test
36	ANTENNA_SELECT	Antenna diversity control. Can be used to control external antenna switch. If differential signal is needed, two GPIOs can be used with one set to invert
37	MARC_2PIN_STATUS[1]	Partial FSM state signal, can be muxed on GPIOs pins for easily getting MARC state. 00:SETTLING 01:TX 10:IDLE 11:RX
38	MARC_2PIN_STATUS[0]	Partial FSM state signal, can be muxed on GPIOs pins for easily getting MARC state. 00:SETTLING 01:TX 10:IDLE 11:RX

39	3: 2: TXFIFO_OVERFLOW 1: 0: RXFIFO_UNDERFLOW	3: Reserved - Used for test 2: TX FIFO overflow, MCU writes too much data 1: Reserved - Used for test 0: RX FIFO underflow, MCU reads too much data
40	3: MAGN_VALID 2: CHFILT_VALID 1: RCC_CAL_VALID 0: CHFILT_STARTUP_VALID	3: New CORDIC magnitude sample 2: New channel filter sample 1: RCOSC calibration has been performed at least once 0: Channel filter has settled
41	3: COLLISION_FOUND 2: SYNC_EVENT 1: COLLISION_FOUND 0: COLLISION_EVENT	3: High when SUW detected during receive mode (Sticky) 2: Sync detect (pulse) 1: Same as 3 0: Sync detected during receive (pulse)
42	PA_RAMP_UP	Signal that ramping is started (for compliance testing)
43	3: CRC_FAILED 2: LENGTH_FAILED 1: ADDR_FAILED 0: UART_FRAMING_ERROR	3: Packet CRC error 2: Packet Length error 1: Packet Address error 0: Packet UART framing error
44	AGC_STABLE_GAIN	AGC has settled to a gain
45	AGC_UPDATE	AGC update (Pulse)
46	ADC_DATA	3: ADC clock 2: ADC Q data sample 1: ADC clock 0: ADC I data sample
47	DEM_SERIAL_DEBUG	Demdoulator debug data
48	HIGHZ	High impedance (tri state)
49	EXT_CLOCK	External clock, divided crystal clock. Division factor controlled by register
50	CHIP_RDY_N	Chip ready, XOSC and regulator settled, ready to receive commands.
51	HW0	HW to 0 (HW1 achieved with _INV signal)
52 - 53		Reserved - Used for test
54	CLOCK_32K	32 kHz clock output from internal RC oscillator
55	IOC_WOR_EVENT0	WOR EVENT0 (non maskable)
56	IOC_WOR_EVENT1	WOR EVENT1 (non maskable)
57	IOC_WOR_EVENT2	WOR_EVENT2 (non maskable)
58		Reserved - Used for test
59	XOSC_STABLE	Internal XOSC has finished settling. Debug only
60	EXT_OSC_EN	External oscillator enable, to control e.g TCXO
61 - 63		Reserved - Used for test

Table 8: GPIO Output Pin Mapping

IRQ0_MASK	Interrupt Name	Description
0	PQT_EVENT	Preamble Quality Threshold reached, single cycle pulse when preamble is detected
1	SYNC_EVENT	Single cycle pulse when sync is detected
2	CLKEN_SOFT	Pulse when new soft data is available
3	COLLISION_EVENT	Collision detected. Sync word detected during reception.
4	CRC_OK	Packet received with correct CRC
5	RSSI_UPDATE	Received Signal Strength algorithm updated
6	MAGN_VALID	Magnitude from CORDIC is valid, debug / special use only
7	MARC_MCU_WAKEUP	Main Radio Control Unit requires MCU interaction, read MARC status register to find cause of wake up event.

Table 9: Interrupt Mapping

GPIO Pin	Signal Name	Description
0	IOC_SERIAL_TX	Serial data input, connects to modulator, synchronized in IOC
1	NOT USED	SPI SO signal
2	IOC_SYNC_FOUND	In blind mode sync is done externally, used for gearshift of demodulator logic, synchronized in IOC
3	EXT_32K_CLOCK	External 32 kHz clock signal

Table 10: GPIO Input Pin Mapping

4 Common Receive and Transmit Configuration

4.1 Modulation Formats

CC112X supports amplitude and frequency shift modulation formats. The desired modulation format is set in the `MODCFG_DEV_E.MOD_FORMAT` register.

Optionally, the data stream can be Manchester coded by the modulator and decoded by the demodulator. This option is enabled by setting `MDMCFG1.MANCHESTER_EN = 1`.

4.1.1 Frequency Shift Keying

CC112X supports both 2-FSK and 4-FSK modulation. Both can optionally be shaped by a Gaussian filter with BT = 0.5, producing a GFSK modulated signal. This spectrum-shaping feature improves adjacent channel power (ACP) and occupied bandwidth.

In 'true' 2-FSK systems with abrupt frequency shifting, the spectrum is inherently broad. By making the frequency shift 'softer', the spectrum can be made significantly narrower. Thus, higher data rates can be transmitted in the same bandwidth using GFSK.

When 2-(G)FSK/4-(G)FSK modulation is used, the `DEVIATION_M` and `MOD_CFG_DEV_E.DEV_E` register specifies the expected frequency deviation of incoming signals in RX and should be the same as the TX deviation for demodulation to be performed reliably and robustly.

The frequency deviation is programmed with the `DEV_M` and `DEV_E` values in the `DEVIATION_M` and `MOD_CFG_DEV_E.DEV_E` register. The value has an exponent/mantissa form, and the resultant deviation is given by:

<u>`DEVIATION_E` >0:</u>

$$f_{dev} = \frac{f_{xosc}}{2^{24}} \cdot (256 + DEV\_M) \cdot 2^{DEV\_E}$$

<u>`DEVIATION_E` = 0:</u>

$$f_{dev} = \frac{f_{xosc}}{2^{23}} \cdot DEV\_M$$

The default symbol encoding is shown in Table 11. The symbol encoding can be configured through the `SOFT_TX_DATA_CFG.SYMBOL_MAP_CFG` register field.

Format	Symbol	Coding
2-(G)FSK	'0'	– Deviation
	'1'	+ Deviation
4-(G)FSK	'01'	– Deviation
	'00'	– 1/3·Deviation
	'10'	+1/3·Deviation
	'11'	+ Deviation

Table 11: Symbol Encoding for 2-(G)FSK and 4-(G)FSK Modulation

4.1.2 Minimum Shift Keying

When using MSK[1], the complete transmission (preamble, sync word, and payload) will be MSK modulated.

MSK modulation is configured by `MODCFG_DEV_E.MOD_FORMAT` set to 2-FSK modulation and frequency deviation set to ¼ of the data rate. Phase shifts are then performed with a constant transition time.

Data Rate [ksps]	Frequency Deviation [kHz]	Actual Modulation Index
1.0	0.25	0.49999
1.2	0.3	0.50000
2.4	0.6	0.50000
4.8	1.2	0.50041
9.6	2.4	0.49958
19.6	4.8	0.50010
38.4	9.6	0.49963
50	12.5	0.50048
76.8	19.2	0.49966
100	25.0	0.50048
125	31.25	0.50047

Table 12: MSK Parameters

4.1.3 Amplitude Modulation (ASK) and On-Off Keying (OOK)

CC112X supports two different forms of amplitude modulation: On-Off Keying (OOK) and Amplitude Shift Keying (ASK).

OOK modulation simply turns the PA on or off to modulate ones and zeros respectively.

ASK in ***CC112X*** allows programming of the modulation depth (the difference between 1 and 0), and shaping of the pulse amplitude. Pulse shaping produces a more bandwidth constrained output spectrum (see Figure 7 for shaped / unshaped spectrum).

[1] Identical to offset QPSK with half-sine shaping (data coding may differ)

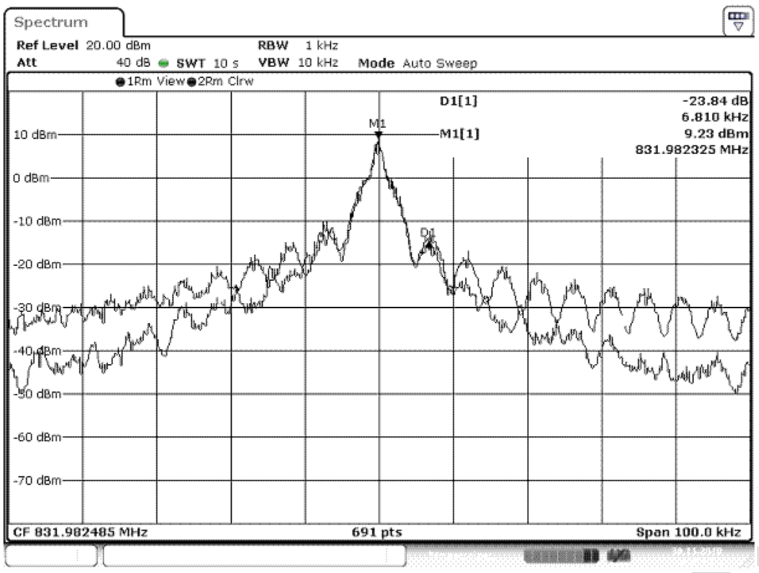


Figure 7: OOK with and without Shaping

4.1.4 Custom Frequency Modulation / Analog FM

CC112X supports a simple scheme to do arbitrary frequency modulation/analog FM (e.g. communication with analog legacy voice devices, N-FSK systems). This feature utilizes the high resolution PLL and lets the user in a simple way directly control/read the instantaneous frequency without SPI overhead. Custom frequency modulation is enabled by setting `SOFT_TX_DATA_CFG.SOFT_TX_DATA_EN = 1`.

A one byte register can be used to set the carrier frequency (`SOFT_TX_DATA_IN`) and a one byte register can be used to read the instantaneous frequency (`SOFT_RX_DATA_OUT`). The two registers have the same format to simplify SW control in both TX and RX. This feature is best combined with the use of SPI burst access with `EXT_CTRL.BURST_ADDR_INCR_EN = 0` to continuously write/read to the same address without any SPI address overhead.

The `SOFT_TX_DATA_IN` represents a range normalized to $\pm f_{dev}/64$. Legal range for `SOFT_TX_DATA_IN` is −64 to +64 giving 129 valid values/frequency steps between $-f_{dev}$ and $+f_{dev}$. Values outside these are considered to be reserved. The frequency deviation is programmed through the `DEV_M` and `DEV_E` values in the `DEVIATION_M` and `MOD_CFG_DEV_E.DEV_E` registers).

Internally, the modulator operates with a normalized representation of the frequency deviation, where $\pm 512 = \pm f_{dev}$. That allows 1024 intermediate steps between the two extreme values to account for any possible shape (for instance, a Gaussian curve). The actual instantaneous deviation is then found as the product of the shape and f_{dev}. The use of symmetric 512 values allows for a more precise multiplication and rounding.

When using costume frequency modulation, the external 8-bit value (in the allowed range of −64 to +64) is directly multiplied by 8 to reach the internal ±512 normalized symbols.

The modulator writes values to the PLL at 16x the programmed data rate. Between the modulator and the PLL there is an optional linear upsampler with a configurable upsampler factor of 1, 2, 4, 8, 16, 32, or 64 configured through the `PA_CFG0.UPSAMPLER_P` register field.

The signal MOD_SOFT_TX_DATA_CLK can be set on a GPIO and used as interrupt on the MCU to synchronize the SPI data to the internal modulation rate. The signal runs at 16x the programmed data rate.

The signal DEM_CLKEN_SOFT can be set on a GPIO and used as a trigger to read the SOFT_RX_DATA_IN samples.

4.1.5 DSSS PN Mode

CC112X supports DSSS PN Mode for application requiring high sensitivity. Preamble and sync word is unchanged, but the payload data bit is spread with a fixed PN Gold sequence. The spreading factor is set to 4, thus the programmed data rate is reduced by 4.

DSSS PN mode is enabled by setting `MODCFG_DEV_E.MODEM_MODE = 10b`.

Note that the sync detection logic is disabled in this mode to force the receiver to continuously calculate channel power.

4.1.6 DSSS Repeat Mode

CC112X supports DSSS Repeat Mode for application requiring high sensitivity. In DSSS Repeat mode preamble is unchanged. Payload data bit is spread using the sync word. Only sync word length 11 and 16 is supported, and the PN gold sequence is initialized at the beginning of each packet.

DSSS Repeat Mode is enabled by setting the `MODCFG_DEV_E.MODEM_MODE = 01`. Note that infinite packet length mode must be enabled when using this mode (`PKT_CFG0.LENGTH_CONFIG = 10b`).The decoded DSSS Repeat Mode signals (DSSS_CLK, DSSS_DATA1, and DSS_DATA0) are available on the GDOx pins by setting `IOCFGx.GPIOx_CFG = 18`. Normal mode/FIFO mode (`PKT_CFG2.PKT_FORMAT = 00`) is not supported when using DSSS Repeat Mode.

4.2 Data Rate Programming

The data rate used in transmit and the data rate expected in receive is programmed by the `DATARATE_M` and the `DATARATE_E` configuration settings. The data rate is given by the formula below. As the formula shows, the programmed data rate depends on the crystal frequency.

Note that `DATARATE_M` is 20 bits wide and consists of the register fields `DATARATE_M_19_16`, `DATARATE_M_15_8` and `DATARATE_M_7_0` found in `DRATE2`, `DRATE1`, and `DRATE0` respectively.

<u>`DATARATE_E` > 0:</u>

$$R_{DATA} = \frac{(2^{20} + DATARATE\_M)\cdot 2^{DATARATE\_E}}{2^{39}} \cdot f_{XOSC}$$

<u>`DATARATE_E` = 0:</u>

$$R_{DATA} = \frac{DATARATE\_M}{2^{38}} \cdot f_{XOSC}$$

The following approach can be used to find suitable values for a given data rate:

$$DATARATE\_E = \left\lfloor \log_2\left(\frac{R_{DATA}\cdot 2^{39}}{f_{XOSC}}\right) - 20 \right\rfloor$$

$$DATARATE\_M = \frac{R_{DATA}\cdot 2^{39}}{f_{XOSC}\cdot 2^{DATARATE\_E}} - 2^{20}$$

If `DATARATE_M` is rounded to the nearest integer and becomes 2^{20}, one should increment `DATARATE_E` and use `DATARATE_M = 0` instead.

The data rate can be set from 0.6 ksps to 100 ksps with the minimum step size according to Table 13.

Min Data Rate [ksps]	Typical Data Rate [ksps]	Max Data Rate [ksps]	Data Rate Step Size [ksps]
0	0.04	0.15	0.00014
0.15	0.25	0.3	0.00014
0.61	1.2	1.22	0.0006
1.22	2.4	2.44	0.001
2.44	4.8	4.88	0.002
4.88	9.6	9.76	0.005
19.5	25	39.0	0.018
39.0	50	78.1	0.037
78.1	100	125	0.074

Table 13: Data Rate Step Size

The data rate is programmed as symbol rate meaning that for 4-(G)FSK, DSSS mode, or Manchester Coding, the bit rate and symbol rate is not equal. Table 14 shows the relationship between bit rate and symbol rate.

Modulation Format	Bit Rate / Symbol Rate Ratio
2-(G)FSK / OOK / ASK	$\frac{R_{Bit}}{R_{Symbol}} = 1$
Manchester Coding	$\frac{R_{Bit}}{R_{Symbol}} = \frac{1}{2}$
4-(G)FSK	$\frac{R_{Bit}}{R_{Symbol}} = 2$
DSSS Mode	$\frac{R_{Bit}}{R_{Symbol}} = \frac{1}{SpreadingFactor}$

Table 14: Bit Rate vs. Symbol Rate

5 Receive Configuration

5.1 Receiver Channel Filter Bandwidth

In order to meet different channel width requirements, the receiver channel filter is programmable. The `CHAN_BW.ADC_CIC_DECFACT` and `CHAN_BW.BB_CIC_DECFACT` register fields control the receiver channel filter bandwidth, which scales with the crystal oscillator frequency. When `CHAN_BW.ADC_CIC_DECFACT = 0` a value of 20 should be used in the equation below, and when `CHAN_BW.ADC_CIC_DECFACT = 1` a value of 32 should be used.

The following formula gives the relation between the register settings and the channel filter bandwidth:

$$CBW = \frac{f_{xosc}}{ADC\_CIC\_DECFACT \cdot BB\_CIC\_DECFACT \cdot 8}$$

Table 15 and Table 16 list some channel filter bandwidths configurations supported by the CC112X.

ADC_CIC_DECFACT = 0		ADC_CIC_DECFACT = 1	
BB_CIC_DECFACT	CBW [kHz]	BB_CIC_DECFACT	CBW [kHz]
0x01	250	0x01	250
0x03	83	0x03	83
0x05	50	0x05	50
0x0A	25	0x0A	25
0x14	12.5	0x14	12.5
0x28	6.25	0x28	6.25

Table 15: Channel Bandwidth Examples with 40 MHz XOSC (CC1125)

ADC_CIC_DECFACT = 0		ADC_CIC_DECFACT = 1	
BB_CIC_DECFACT	CBW [kHz]	BB_CIC_DECFACT	CBW [kHz]
0x01	200	0x01	125
0x02	100	0x02	62.5
0x04	50	0x03	42
0x08	25	0x05	25
0x10	12.5	0x0A	12

Table 16: Channel Bandwidth Examples with 32 MHz XOSC (CC1120 and CC1125)

By compensating for a frequency offset between the transmitter and the receiver, the filter bandwidth can be reduced and the sensitivity can be improved.

The `CHAN_BW.ADC_CIC_DECFACT` sets the bandwidth into the digital low-IF mixer, see Figure 8. A narrow bandwidth use less power consumption and gives better sensitivity at the cost of more accurate RF crystals.

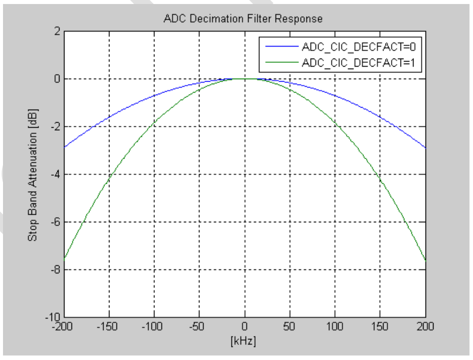


Figure 8: ADC Decimation Filter Response

5.2 DC Offset Removal

CC112X support Low-IF and Zero-IF receiver architecture, which is set by the `FREQ_IF_CFG.FREQ_IF` register field. For more information see section 9. For Zero-IF (and receiver architecture where DC offset component is not blocked by channel filter) the DC offset removal must be enabled by setting `DCFILT_CFG.DCFILT_FREEZE_COEFF = 0`. The `DCFILT_CFG` configures the DC filter bandwidth, both during settling period and during tracking. There is a tradeoff between bandwidth and settle time. Narrower DC filter bandwidth requires longer settling time. It is strongly recommended to use SmartRF Studio to generate DC removal parameter settings.

5.3 Automatic Gain Control

CC112X contains an Automatic Gain Control (AGC) for adjusting the input signal level to the demodulator.

The AGC behavior depends on the following register fields:

- `AGC_CFG2.FE_PERFORMANCE_MODE`
- `AGC_REF.AGC_REFERENCE`
- `AGC_CFG1.AGC_SYNC_BEHAVIOUR`
- `AGC_CFG1.AGC_WIN_SIZE`
- `AGC_CFG1.AGC_SETTLE_WAIT`
- `AGC_CFG2.AGC_MAX_GAIN`
- `AGC_CFG3.AGC_MIN_GAIN`

The `AGC_CFG2.FE_PERFORMANCE_MODE` sets the correct gain tables that shall be used for a given operation mode.

The `AGC_REFERENCE` sets the reference value for the AGC and the setting is a compromise between blocker tolerance/selectivity and sensitivity. The value sets the desired signal level in the channel into the demodulator and it must be larger than the required minimum signal-to-noise ratio for the demodulator with a wanted margin. Typically the minimum signal-to-noise ratio varies from 5 dB to 15 dB. Increasing this value reduces the headroom for blockers, and therefore close-in selectivity.

The `AGC_CFG1.AGC_SYNC_BEHAVIOUR` sets the AGC behavior and RSSI update behavior after sync word is found.

The `AGC_CFG1.AGC_WIN_SIZE` sets the AGC integration window size for each level adjustment. The AGC update rate refers to the channel filter sampling frequency, which is programmed to be 4 times the desired channel bandwidth. The `AGC_CFG1.AGC_SETTLE_WAIT` is a table index for selecting the time to wait for the AGC before a new measurement starts. The `AGC_CFG2.AGC_MAX_GAIN` and `AGC_CFG3.AGC_MIN_GAIN` sets the maximum/minimum gain. The resolution is 1/3 dB. Note that 0 refers to maximum gain while 17 refers to minimum gain.

The `AGC_CFG2.FE_PERFORMANCE_MODE` table selection will default select the max/min gain settings, but changing max/min gain setting will override the predefined table entries. A lower maximum gain will reduce power consumption in the receiver front end, since the highest gain settings are avoided. Limiting max gain also improves worst case linearity in the front-end, this is particularly useful when using external LNA.

See Figure 9 for detailed timing information on different AGC signals.

5.4 Demodulator, Symbol Synchronizer, and Data Decision

CC112X contains an advanced and highly configurable demodulator. Channel filtering and frequency offset compensation is performed digitally. By doing filtering and compensation using digital logic, RF performance parameters are less vulnerable to variations due to shifts in process, temperature, and voltage levels.

To generate the RSSI level (see Section 5.13 for more information), the signal level in the channel is estimated. Data filtering is also included for enhanced performance.

5.5 Frequency Offset Compensation

The ***CC112X*** has very fine frequency resolution. This can be used to compensate for initial crystal frequency tolerance in system production test. If the temperature sensor is also used, it is possible to do crystal temperature drift compensation in SW to reduce the cost of the RF crystal.

When using 2-(G)FSK, 4-(G)FSK, or MSK modulation, the demodulator will compensate for the offset between the transmitter and receiver frequency within certain limits, by estimating the centre frequency of the received signal. By compensating for a large frequency offset between the transmitter and the receiver, the sensitivity can be improved.

The ***CC112X*** receiver supports frequency offset compensation both in the frequency synthesizer and after the channel filter.

The FREQOFF_CFG.FOC_CFG register selects the frequency offset compensation algorithm. If frequency offset compensation in the frequency synthesizer is selected, the receiver will use a feedback loop to the frequency synthesizer. CC112X will then track frequency offset and center the channel filter around the wanted signal. Then, after the sync word is found, the frequency offset compensation value is frozen and the frequency compensation during packet reception is done after the channel filter.

The tracking loop for frequency offset compensation after the channel filter has a single gain factor which affect noise sensitivity of the algorithm.

FREQOFF_CFG.FOC_KI_FACTOR sets the tracking loop gain.

The tracking loop for frequency compensation in the frequency synthesizer has a single gain factor, which affect the settling time and sensitivity. FREQOFF_CFG.FOC_CFG activates frequency tracking and sets the tracking loop gain. The tracking range of the algorithm is selectable as fractions of the channel bandwidth with the FOC_LIMIT configuration register.

If the MDMCFG1.CARRIER_SENSE_GATE bit is set, the offset compensator will not start until carrier sense is asserted. This may be useful for tracking large offsets, since the algorithm will likely drift to the boundaries when trying to track noise.

The estimated frequency offset value is available in the FREQOFF_EST status register. This can be used for permanent frequency offset compensation. By writing the value from FREQOFF_EST into the FREQOFF register (or use the SAFC strobe), the frequency synthesizer will automatically be adjusted according to the estimated frequency offset.

Frequency offset compensation in frequency synthesizer increases frequency tracking range of the receiver from channel filter bandwidth to channel filter bandwidth + 2*FOC_LIMIT. This compensation algorithm requires 1 - 4 byte preamble for settling depending on the tracking range.

Note: Frequency offset compensation is not supported for ASK or OOK modulation.

5.6 Image Compensation

CC112X has support for the ImageExtinct image compensation algorithm that digitally compensates for I/Q mismatch. ImageExtinct removes the image component completely, removing any issues at the system image frequency. ImageExtinct removes the need for time consuming image calibration steps in system production test, reducing both test time and test cost. This feature is enabled by setting IQIC.IQIC_EN = 1.

5.7 Bit Synchronization

The bit synchronization algorithm extracts the clock from the incoming symbols. The algorithm requires that the expected data rate is programmed as described in Section 4.2. Re-synchronization is performed continuously to adjust for error in the incoming symbol rate.

It is possible to select between two different bit synchronization algorithms.

TOC_CFG.TOC_LIMIT sets the bit synchronization algorithm and Table 17 below shows the properties of the bit synchronization algorithms.

TOC_LIMIT	Data Rate Offset Tolerance	Required Preamble Length
0	< 2000 ppm	0.5 byte (only for gain adjustment)
1	< 2%	Approximately 4 bytes
2	Reserved	
3	< 12%	Approximately 4 bytes (CS needed)

Table 17: Bit Synchronization Property

Using the low tolerance setting, the novel InstantSync capture logic is enabled. The InstantSync algorithm does not need any preamble bits for bit synchronization or frequency offset compensation, greatly reducing system settling times and system power consumption (0.5 preamble byte needed for AGC settling).

5.8 Byte Synchronization, Sync Word Detection

Byte synchronization is achieved by a continuous sync word search. The sync word can be programmed to be 11, 16, 18, 24 or 32 (through the `SYNC_CFG0.SYNC_MODE` register field) unique bits that are automatically inserted at the start of the packet by the modulator in transmit mode. The MSB in the sync word is sent first. The demodulator uses the sync word to find the start of the incoming packet.

The sync word will also function as a system identifier, since only packets with the correct predefined sync word will be received.

The ***CC112X*** will continuously calculate a sync word qualifier value to distinguish the sync word (or start of packet) from background noise. This value is available in the `PQT_SYNC_ERR.SYNC_ERROR` register field. If the sync word qualifier is less than the programmed sync threshold (`SYNC_CFG1.SYNC_THR)` the demodulator starts to demodulate the packet.

The received sync word can be further qualified using the bit check on sync word enabled through `SYNC_CFG0.SUW_NUM_ERROR` and/or a carrier sense condition (`MDMCFG1.CARRIER_SENSE_GATE`). The sync word is configured through the `SYNC0/1/2/3` registers.

The ***CC112X*** supports DualSync search which makes it possible to concurrently search for 2 different 16 bit sync word. DualSync search is enabled by settings `SYNC_CFG0.SYNC_MODE = 111b`.

5.9 Received Signal Qualifiers and Link Quality Information

CC112X has several qualifiers that can be used to increase the likelihood that a valid sync word is detected:

- Sync Word Qualifier
- Preamble Quality Threshold
- RSSI
- Carrier Sense
- Clear Channel Assessment
- Link Quality Indicator

5.10 Sync Word Qualifier

If sync word detection in RX is enabled in the `SYNC_CFG0.SYNC_MODE` register, the ***CC112X*** will not start filling the RX FIFO and perform the packet filtering described in Section 7.2 before a valid sync word has been detected (i.e. the continuously calculated sync word qualifier value is less than programmed sync threshold (`SYNC_CFG1.SYNC_THR)`).

5.11 Preamble Detection

CC112X has a high performance preamble detector which can be turned on by configuration register `PREAMBLE_CFG0.PQT_EN = 1`.

A preamble is detected if preamble with a quality above the programmed threshold (PQT) is present. Then a “Preamble Quality Reached” signal can be observed on one of the GPIO pins by setting `IOCFGx.GPIOx_CFG = SERIAL_CLK` (8). It is also possible to determine if preamble quality is reached by checking the `PQT_REACHED` bit in the `RX_STATUS` register. This signal / bit asserts when the received signal exceeds the PQT.

The preamble quality estimator uses a correlation filter to find a valid preamble. The threshold is configured with the register field `PREAMBLE_CFG0.PQT`.

Another use of the preamble quality threshold is as a qualifier for the optional RX termination timer (see Section 8.5.1 for more details).

There is also a PQT startup timer available. Preamble search will not be gated before `PQT_VALID` is asserted. The startup timer period is programmable (through `PREAMBLE_CFG0.PQT_VALID_TIMEOUT`) to enable a tradeoff between speed and accuracy.

5.12 Collision Detector

A collision is here referred to as the following

a) A packet is currently being received (data is put into the RX FIFO) when

b) a new packet appears on the air (with higher signal energy that blocks the packet currently being received)

This situation can be handled in different ways:

- Not handled. The current packet received will have data errors and can be discarded. This method will work well in most systems as RF protocols have capabilities for re-transmissions to solve these issues.
- RX can be terminated and resumed later.
- RX can be restarted to receive the new packet. For this to be successful, the new packet must have signal energy that is sufficiently higher than the current packet to allow correct demodulation. This scenario is mainly for high throughput protocols where nodes communicate with several nodes at various distances.

CC112X has several powerful features to cover the above scenarios. Two examples are given here based on RSSI and PQT

5.12.1 RSSI Based Detection

During packet reception a collision can be detected as a step in RSSI. When the first packet is detected, the carrier sense level can be changed to a value higher than the RSSI for the current packet. If a new packet with higher signal power appears on the air, a carrier sense interrupt will tell the MCU to restart RX. The SRX strobe can be used to immediately restart the demodulator to catch the incoming packet. Since an `SFRX` strobe cannot be issued from RX state one should read the `NUM_RXBYTES` register to find out how many bytes belong to the first packet.

5.12.2 Preamble Based Detection

A more robust indicator is to use the preamble detector. If a new message is incoming it will start with a preamble. A PQT interrupt can then be used to indicate a collision. If the data can contain a 010101 preamble sequence, a check for a step in RSSI should also be done before running the SRX strobe command. Also in this case one should read the `NUM_RXBYTES` register before strobing SRX over again.

5.13 RSSI

The AGC module returns an estimate on the received in-band signal power at the antenna called Received Signal Strength Indicator (RSSI), and the valid range of values is [-127, 128] dBm with 0.0625 dBm resolution. A value of -128 dBm indicates that the RSSI is invalid.

The RSSI value is output from a configurable moving average filter in order to reduce uncertainty in the RSSI estimates. It is as such possible to trade RSSI computation speed / update rate against RSSI accuracy. This trade-off is determined by configuring the `AGC_CFG0.RSSI_VALID_CNT` register. The number of new input samples to the moving average filter (internal RSSI estimates) that are required before the next update of the RSSI value is made is given by [$2^{\text{AGC\_CFG0.RSSI\_VALID\_CNT}}$ `+ 1]`. The `RSSI0.RSSI_VALID` register field will be asserted from the first RSSI update. RSSI computation speed is prioritized when the number of required input samples is low, while RSSI accuracy is prioritized when the number of input samples required is high. Carrier

Sense (CS) indication will also be affected by the setting of `AGC_CFG0.RSSI_VALID_CNT`, since CS is evaluated each time the RSSI is updated or the programmable CS threshold is changed.

An external MCU has several ways to obtain information from the chip that is RSSI specific:

- The RSSI value, which is a 2's complement number, can always be read by polling the `RSSI1/RSSI0` registers. These two registers constitute the RSSI value in Figure 9.
- The validness of the current RSSI value (obtained by polling the `RSSI0.RSSI_VALID` register field) is depending on the configuration of the `AGC_CFG0.RSSI_VALID_CNT` register. It can also be routed out on a GPIO pin by configuring `IOCFGX.GPIOX_CFG = RSSI_VALID` (13). The behavior of `RSSI0.RSSI_VALID` is equal to the behavior of `RSSI0.AGC_CARRIER_SENSE_VALID`. The latter is shown in Figure 9.
- By setting the `IOCFG0.GPIO0_CFG = AGC_UPDATE` (14), a pulse of $[f_{XOSC}/2]^{-1}$duration will occur on GPIO0 each time the frontend gain has been adjusted. The observed signal is labeled AGC_UPDATE in Figure 9 and in Table 8.
- By setting the `3.GPIO3_CFG` or `IOCFG2.GPIO2_CFG = RSSI_UPDATE` (14), a pulse of $[f_{XOSC}/2]^{-1}$duration will occur on GPIO3 or GPIO2 each time the RSSI value is updated. This observed signal is labeled RSSI_UPDATE in Figure 9 and in Table 8.
- By setting the `IOCFG2.GPIO2_CFG = AGC_STABLE_GAIN` (44), GPIO2 will be asserted when the AGC gain is stable. The AGC gain is reported stable whenever the current gain setting is equal to the previous gain setting. This condition is evaluated each time a new internal RSSI estimate is computed and the result is labeled AGC_STABLE_GAIN in Figure 9 and in Table 8.
- By setting the `IOCFG1.GPIO1_CFG = AGC_HOLD` (14), GPIO1 will be high whenever the AGC is waiting for signals to stabilize after a gain adjustment. The observed signal is labeled AGC_HOLD in Figure 9 and in Table 8. CSn must be high when observing this signal.

Figure 9 shows an example of the behavior of RSSI specific signals given two different values for the `AGC_CFG0.RSSI_VALID_CNT` register value.

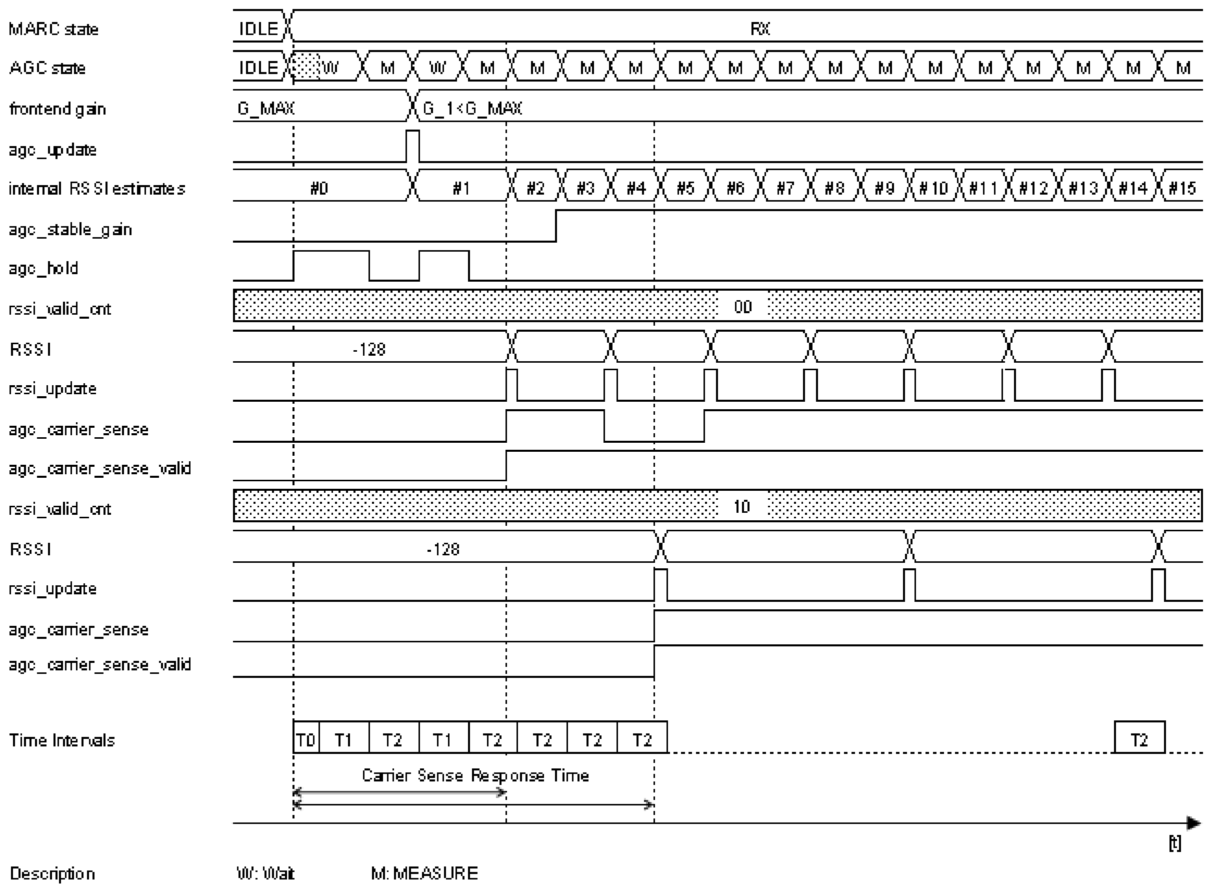


Figure 9: RSSI Timing Diagram

T2 is the time the AGC uses to measure the signal strength and potentially adjust the gain while T1 is the time the AGC waits after adjusting the frontend gain to allow signal transients to decay before the next signal strength measurement can take place. Figure 9 also shows a start-up delay before RSSI measurements can commence, T0. This delay is dependent on demodulator settings. The CS response time indicates the maximum amount of time the radio will be in RX when RSSI-based RX termination is enabled. More information on this topic is located in section 8.5. As an example, the AGC_CFG0.RSSI_VALID_CNT value of 0x00 in Figure 9 makes the averaging window short and leads to a failing CS indication on the second RSSI update, while this is not observed for the greater value of AGC_CFG0.RSSI_VALID_CNT.

The following provide means to calculate the maximum CS response time given certain configuration register settings.

Important: Use a value of 1 for CHAN_BW.BB_CIC_DECFACT if CHAN_BW.BB_CIC_DECFACT is configured to 0 or 1. CHAN_BW.ADC_CIC_DECFACT equals 20 or 32 according to the configuration register setting. The AGC_SETTLE_WAIT and AGC_WIN_SIZE register fields are found in the AGC_CFG1 register. The duration of T1 and T2 are as follows:

$$T1 = \frac{[16 \cdot AGC\_SETTLE\_WAIT + 32] : \cdot BB\_CIC\_DECFACT \cdot ADC\_CIC\_DECFACT}{f_{XOSC}}$$

$$T2 \le \frac{2^{AGC\_WIN\_SIZE+4} \cdot BB\_CIC\_DECFACT \cdot ADC\_CIC\_DECFACT + 46}{f_{XOSC}}$$

The duration of T0 is found using Table 18 and the following equations.

Configuration Register Fields / Conditions			**Delay**					
`(MODCFG_DEV_E.MODEM_MODE != 0x11) && (CHAN_BW.BB_CIC_DECFACT > 0x01)`	`DCFILT_CFG. DCFILT_FREEZE_COEFF`	`MDMCFG1. CARRIER_SENSE_GATE`	**D0**	**D1**	**D2**	**D3**	**D4**	**D5**
0	0	0	1			1		1
0	0	1	1		1			1
0	1	0	1			1	1	
0	1	1	1		1		1	
1	0	0	1	1		1	1	
1	0	1	1	1	1		1	
1	1	X	1	1			1	

Table 18: T0 Matrix

$$D0 = \frac{17 \cdot ADC\_CIC\_DECFACT}{f_{XOSC}}$$

$$D1 = \frac{30 \cdot BB\_CIC\_DECFACT \cdot ADC\_CIC\_DECFACT}{f_{XOSC}}$$

$$Exp = DCFILT\_CFG.DCFILT\_BW\_SETTLE \; when$$
$$DCFILT\_CFG.DCFILT\_BW\_SETTLE \; < 7 \; else \; 6$$

$$D2 = \frac{[29 + 2^{(5+Exp)}] \cdot 2 \cdot ADC\_CIC\_DECFACT}{f_{XOSC}}$$

$$D3 = \frac{60 \cdot ADC\_CIC\_DECFACT}{f_{XOSC}}$$

$$D4 = \frac{70 \cdot BB\_CIC\_DECFACT \cdot ADC\_CIC\_DACFACT}{f_{XOSC}}$$

$$D5 = \frac{72 \cdot BB\_CIC\_DECFACT \cdot ADC\_CIC\_DECFACT}{f_{XOSC}}$$

$$T0 \le \sum Applicable \; Delays \,|_{CURRENT \; CONFIGURATION}$$

The maximum Carrier Sense Response Time is found by:

$$\text{CS Response Time} \leq T0 + (T1 + T2) \cdot (2^{RSSI\_VALID\_CNT} + 1)$$

This time can be reduced if knowledge of the maximum measured RSSI is present. In this case it is possible to find the maximum number of gain reductions, GAIN_ADJ_MAX, performed before the first RSSI update. This number can be found by routing RSSI related signals to GPIO as described above. When this number is available, the CS response time is given by:

$$\text{CS Response Time} \leq T0 + T1 \cdot GAIN\_ADJ\_MAX + T2 \cdot (2^{RSSI\_VALID\_CNT} + 1)$$

5.14 Carrier Sense (CS)

Carrier Sense (CS) is asserted when the RSSI is above a programmable CS threshold, `AGC_CS_THR`, and de-asserted when RSSI is below the same threshold. The CS threshold should be set high so that CS is de-asserted when only background noise is present, and low enough that CS is asserted when wanted carrier signal is present, i.e. if threshold is set high, only strong signals are wanted. Different usage of CS includes:

- Sync word qualifier: A sync word cannot be valid if no CS. This is configured by enabling the function in the `MDMCFG1.CARRIER_SENSE_GATE` register.
- Clear Channel Assessment (CCA)
- TX on CCA
- Avoid interference from other RF sources in the ISM band
- RX termination

The latter is described in section 8.5. By setting the `IOCFGX.GPIOX_CFG = CARRIER_SENSE` (17), GPIOx will indicate if a carrier is present. The observed signal is labelled AGC_CARRIER_SENSE in Figure 9 and in Table 8 (CARRIER_SENSE). It is evaluated each time a new internal RSSI estimate is computed. The AGC_CARRIER_SENSE signal must only be interpreted when it is valid, as indicated by AGC_CARRIER_SENSE_VALID. This signal can be routed to GPIOX by setting `IOCFGX.GPIOX_CFG = CARRIER_SENSE_VALID` (16) to help evaluate this in real-time. The two signals can also be read from the `RSSI0` register.

5.15 Clear Channel Assessment (CCA)

The Clear Channel Assessment (CCA) is used to indicate if the current channel is free or busy. The current CCA state is viewable on any of the GPIO pins. `PKT_CFG2.CCA_MODE` selects the mode to use when determining CCA.

When the `STX` or `SFSTXON` command strobe is given while ***CC112X*** is in the RX state, the TX or FSTXON state is only entered if the clear channel requirements are fulfilled. Otherwise, the chip will remain in RX. If the channel then becomes available, the radio will not enter TX or FSTXON state before a new strobe command is sent on the SPI interface. This feature is called TX on CCA. Five CCA requirements can be programmed:

- Always (CCA disabled, always goes to TX)
- If RSSI is below threshold
- Unless currently receiving a packet
- Both the above (RSSI below threshold and not currently receiving a packet)
- If RSSI is below threshold and ETSI LBT [1] requirements are met.

5.16 Listen Before Talk (LBT)

ETSI EN 300 220-1 V2.3.1 [1] has specific requirements for LBT. To simplify compliance **CC112X** has built in HW support to automate the LBT algorithm, including random back-offs. The requirements are taken from the ETSI specifications, and a summary is shown below.

5.16.1 LBT Minimum Listening Time

"The minimum listening time is defined as the minimum time that the equipment listens for a received signal at or above the LBT threshold level (…..) immediately prior to transmission to determine whether the intended channel is available for use.

The listening time shall consist of the "minimum fixed listening time" and an additional pseudo random part. If during the listening mode another user is detected on the intended channel, the listening time shall commence from the instant that the intended channel is free again. Alternatively, the equipment may select another channel and again start the listen time before transmission."

Changing channel is not supported in HW and must be performed by the MCU.

5.16.2 Limit for Minimum Listening Time

"The total listen time, t_L, consists of a fixed part, t_F, and a pseudo random part, t_{PS}, as the following:

$t_L = t_F + t_{PS}$

a) The fixed part of the minimum listening time, t_F , shall be 5 ms.

b) The pseudo random listening time t_{PS} shall be randomly varied between 0 ms and a value of 5 ms or more in equal steps of approximately 0,5 ms as the following:

- *If the channel is free from traffic at the beginning of the listen time, t_L, and remains free throughout the fixed part of the listen time, t_F, then the pseudo random part, t_{PS}, is automatically set to zero by the equipment itself.*
- *If the channel is occupied by traffic when the equipment either starts to listen or during the listen period, then the listen time commences from the instant that the intended channel is free. In this situation the total listen time t_L shall comprise t_F and the pseudo random part, t_{PS}.*

The limit for total listen time for the receiver consists of the sum of a) and b) together."

If LBT is enabled (`PKT_CFG2.CCA_MODE = 100b`) the **CC112X** will run the algorithm until successful transmission.

5.17 Link Quality Indicator (LQI)

The Link Quality Indicator is a metric of the current quality of the received signal. If `PKT_CFG1.APPEND_STATUS` is enabled, the value is automatically added to the last byte appended after the payload. The value can also be read from the `LQI_VAL` register. The LQI gives an estimate of how easily a received signal can be demodulated. LQI is best used as a relative measurement of the link quality (a low value indicates a better link than what a high value does), since the value is dependent on the modulation format.

6 Transmit Configuration

6.1 PA Output Power Programming

PA power ramping is used to improve spectral efficiency of the system by reducing the out of band signal energy created by abrupt changes in the output power. PA output power ramping is used when starting / ending a transmission. The power ramping is very flexible and can be controlled by a configurable, piecewise linear function.

The RF output power level from the device is programmed with the `PA_CFG2.PA_POWER_RAMP` register field. The power level resolution is 0.5 dB. The shaped power ramping is always enabled.

$$OutputPower = \frac{PA\_POWER\_RAMP + 1}{2} - 18$$

Where 3 ≤ `PA_POWER_RAMP` ≤ 64

The shaped power ramping is controlled by the `PA_CFG1.PA_RAMP_CTRL` register. The shaped power ramp up curve passes through two intermediate power levels from off-state to programmed output power level (`PA_CFG2.PA_POWER_RAMP`). The intermediate power levels and total ramp time can be configured. For the shaped ramp up the output power level is split into 16 sections (i.e. 16 steps from 0 to 1, where 1 equals the output power level). The two intermediate power levels are defined using these 16 sections. The first intermediate power level can be programmed within the power level range 0 - 0.5 through `PA_CFG1.PA_RAMP_CTRL[7:5]`. The second intermediate power level can be programmed within power range of 0.5 - 1.0 through `PA_CFG1.PA_RAMP_CTRL[4:2]`.

The ramp down time equals ramp up time.

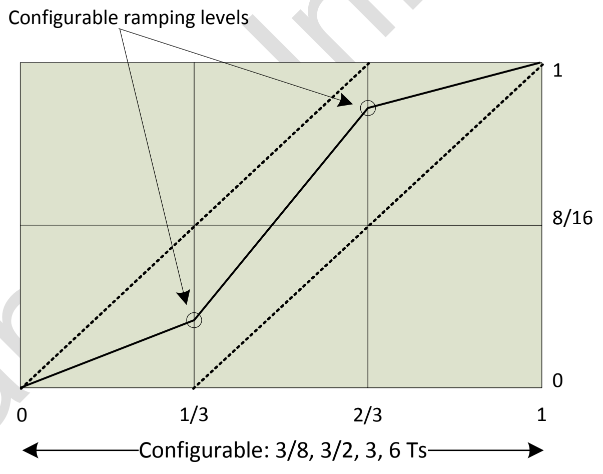


Figure 10: PA Power Ramping Control (configurable in area between dotted lines)

The total PA ramp time is $^3/_8$, $^3/_2$, 3, or 6 symbols and is configured through `PA_CFG1.PA_RAMP_CTRL[1:0]`.

6.2 OOK/ASK Bit Shaping

OOK/ASK bit shaping is always enabled. The OOK/ASK shape length is $^1/_{32}$, $^1/_{16}$, $^1/_8$, or $^1/_4$ symbols and is set through `PA_CFG1.PA_RAMP_CTRL[1:0]`.

7 Packet Handling Hardware Support

The CC112X has built-in hardware support for packet oriented radio protocols.

In transmit mode, the packet handler can be configured to add the following elements to the packet stored in the TX FIFO:

- A programmable number of preamble bytes
- An 11, 16, 18, 24 or 32 bit sync word
- A 2 byte CRC checksum computed over the data field
- Whitening of the data with a PN9 sequence

In receive mode, the packet handling support will de-construct the data packet by implementing the following (if enabled):

- Preamble detection
- Sync word detection
- CRC computation and CRC check
- One byte address check
- Packet length check (length byte checked against a programmable maximum length)
- De-whitening

Optionally, two status bytes (see Table 19 and Table 20) with RSSI value, LQI, and CRC status can be appended in the RX FIFO.

Bit	Field Name	Description
7:0	RSSI	RSSI value

Table 19: Received Packet Status Byte 1 (first byte appended after the data)

Bit	Field Name	Description
7	CRC_OK	1: CRC for received data OK (or CRC disabled or TX mode) 0: CRC error in received data
6:0	LQI	Indicating the link quality

Table 20: Received Packet Status Byte 2 (second byte appended after the data)

Note: Register fields that control the packet handling features should only be altered when CC112X is in the IDLE state unless stated otherwise.

7.1 Standard Packet Format

The format of the data packet can be configured and consists of the following items (see Figure 11):

- Preamble
- Synchronization word
- Optional length byte
- Optional address byte
- Payload
- Optional 2 byte CRC

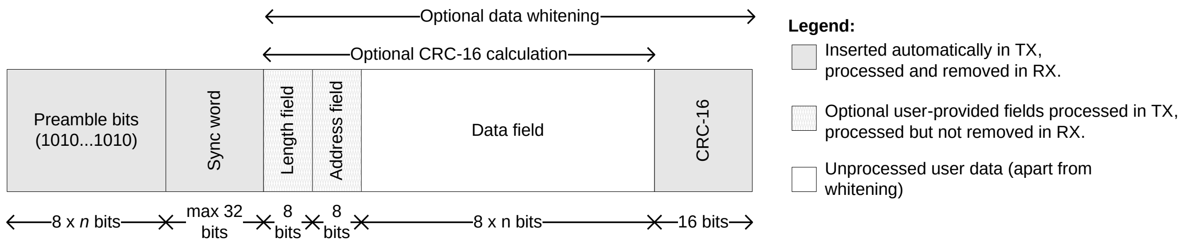

Optional data whitening
Optional CRC-16 calculation
Preamble bits (1010...1010)
Sync word
Length field
Address field
Data field
CRC-16
8 x n bits
max 32 bits
8 bits
8 bits
8 x n bits
16 bits
Legend:
Inserted automatically in TX, processed and removed in RX.
Optional user-provided fields processed in TX, processed but not removed in RX.
Unprocessed user data (apart from whitening)

Figure 11: Packet Format

The preamble pattern is an alternating sequence of ones and zeros (1010··· / 0101··· / 00110011··· / 11001100···) programmable through the `PREAMBLE_CFG1.PREAMBLE_WORD` register field. The minimum length of the preamble is programmable through the `PREAMBLE_CFG1.NUM_PREAMBLE` register field. When strobing TX, the modulator will start transmitting a preamble. When the programmed number of preamble bytes has been transmitted, the modulator will send the sync word and then data from the TX FIFO. If the TX FIFO is empty, the modulator will continue to send preamble bytes until the first byte is written to the TX FIFO. The modulator will then send the sync word and then the data bytes.

The sync word is set in the `SYNC3/2/1/0` registers. The sync word provides byte synchronization of the incoming packet. Non-supported sync word lengths can be emulated by using parts of the preamble pattern in the `SYNC` registers.

CC112X supports both fixed packet length protocols and variable packet length protocols. Variable or fixed packet length mode can be used for packets up to 255 bytes. For longer packets, infinite packet length mode must be used. The packet length is defined as the payload data, and the optional address byte, excluding the optional length byte, the optional CRC, and the optional append status.

7.1.1 Fixed Packet Length

Fixed packet length mode is selected by setting `PKT_CFG0.LENGTH_CONFIG = 00`. The desired packet length is set by the `PKT_LEN` register.

To support non-byte oriented protocols, fixed packet length mode supports packet lengths of n bytes + m bits, where n is programmed through the `PKT_LEN` register and m is programmed through `PKT_CFG0.PKT_BIT_LEN`. If m ≠ 0, only m bits of the last byte read from FIFO is transmitted. This is very useful in low power systems where it is important not to stay in RX / TX longer than necessary. CRC is not supported when `PKT_CFG0.PKT_BIT_LEN ≠ 0`. In RX, you will read zero's from the (8 − m) LSBs in the last byte in the RX FIFO.

Note that n = 0, which gives 256 byte packet length in normal aligned mode is interpreted as zero bytes with only bits transmission when the number of m bits is not zero.

7.1.2 Variable Packet Length

In variable packet length mode, `PKT_CFG0.LENGTH_CONFIG = 01`, the packet length is configured by the first byte after the sync word. The packet length is defined as the payload data, excluding the length byte and the optional CRC. The `PKT_LEN` register is used to set the maximum packet length allowed in RX. Any packet received with a length byte with a value greater than `PKT_LEN` will be discarded.

For the IO Home Control protocol, only the 5 LSB of the length byte is used for length configuration while the 3 MSB are treated as normal data (protocol control bits). This is supported by setting `PKT_CFG0.LENGTH_CONFIG = 11b`. Maximum packet length is hence 32 bytes. The only difference from standard variable length mode is the masking of 3 MSB bits of the received packet length byte, configured through `LENGTH_CONFIG.`

7.1.3 Infinite Packet Length

With `PKT_CFG0.LENGTH_CONFIG = 10b`, the packet length is set to infinite and transmission and reception will continue until turned off manually. As described in the next section, this can be used to support packet formats with different length configuration than natively supported by **CC112X**.

Note: The minimum packet length supported (excluding the optional length byte and CRC) is one byte of payload data.

7.1.4 Arbitrary Length Field Configuration

The packet length register, `PKT_LEN`, can be reprogrammed during receive and transmit. In combination with fixed packet length mode (`PKT_CFG0.LENGTH_CONFIG = 00`), this opens the possibility to have a different length field configuration than supported for variable length packets (in variable packet length mode the length byte is the first byte after the sync word). At the start of reception, the packet length is set to a large value. The MCU reads out enough bytes to interpret the length field in the packet. Then the `PKT_LEN` value is set according to this value. The end of packet will occur when the byte counter in the packet handler is equal to the `PKT_LEN` register. Thus, the MCU must be able to program the correct length before the internal counter reaches the packet length.

7.1.5 Packet Length > 255

The packet automation control register, `PKT_CFG0` can be reprogrammed during TX and RX. This opens the possibility to transmit and receive packets that are longer than 256 bytes and still be able to use the packet handling hardware support. At the start of the packet, the infinite packet length mode (`PKT_CFG0.LENGTH_CONFIG = 10b`) must be active. On the TX side, the `PKT_LEN` register is set to mod(length, 256). On the RX side the MCU reads out enough bytes to interpret the length field in the packet and sets the `PKT_LEN` register to mod(length, 256). When less than 256 bytes remains of the packet, the MCU disables infinite packet length mode and activates fixed packet length mode (`PKT_CFG0.LENGTH_CONFIG = 00`). When the internal byte counter reaches the `PKT_LEN` value, the transmission or reception ends (the radio enters the state determined by `RFEND_CFG0.TXOFF_MODE` or `RFEND_CFG1.RXOFF_MODE`). Automatic CRC appending/checking can also be used (by setting `PKT_CFG1.CRC_CFG = 01 or 10b`).

When for example a 600-byte packet is to be transmitted, the MCU should do the following.

- Set `PKT_CFG0.LENGTH_CONFIG = 2`
- Pre-program the `PKT_LEN` register to mod(600, 256) = 88
- Transmit at least 345 bytes (600 − 255), for example by filling the 128-byte TX FIFO three times (384 bytes transmitted)
- Set `PKT_CFG0.LENGTH_CONFIG = 0`
- The transmission ends when the packet counter reaches 88. A total of 600 bytes are transmitted.

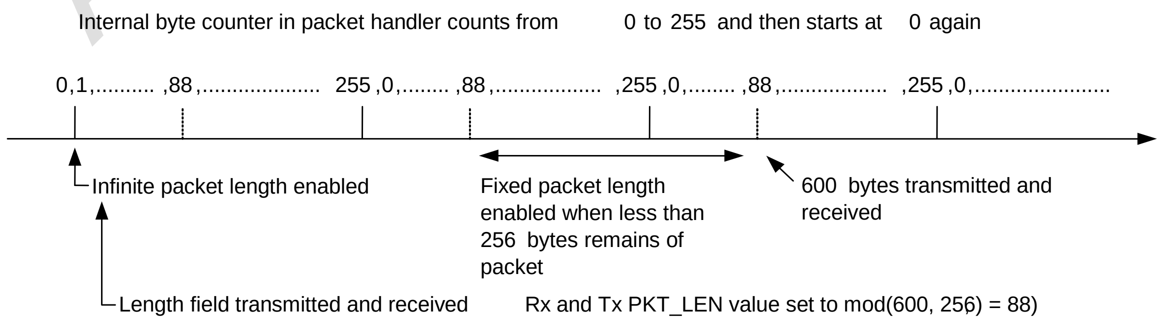

Internal byte counter in packet handler counts from 0 to 255 and then starts at 0 again
0,1,.......... ,88,.................... 255,0,........ ,88,.................. ,255,0,........ ,88,.................. ,255,0,.......................
Infinite packet length enabled
Fixed packet length enabled when less than 256 bytes remains of packet
600 bytes transmitted and received
Length field transmitted and received
Rx and Tx PKT_LEN value set to mod(600, 256) = 88)

Figure 12: Packet Length > 255

7.1.6 Data Whitening

From a radio perspective, the ideal over the air data are random and DC free. This results in the smoothest power distribution over the occupied bandwidth. This also gives the regulation loops in the receiver uniform operation conditions (no data dependencies).

Real data often contain long sequences of zeros and ones making it difficult to track the data bits. In these cases, performance can be improved by whitening the data before transmitting, and de-whitening the data in the receiver.

With ***CC112X***, this can be done automatically. By setting `PKT_CFG1.WHITE_DATA = 1`, all data, except the preamble and the sync word will be XOR-ed with a 9-bit pseudo-random (PN9) sequence before being transmitted. This is shown in Figure 13. At the receiver end, the data are XOR-ed with the same pseudo-random sequence. In this way, the whitening is reversed, and the original data appear in the receiver. The PN9 sequence is initialized to all 1's.

If ***CC112X*** is set up to transmit random data, the PN9 whitening sequence will be transmitted.

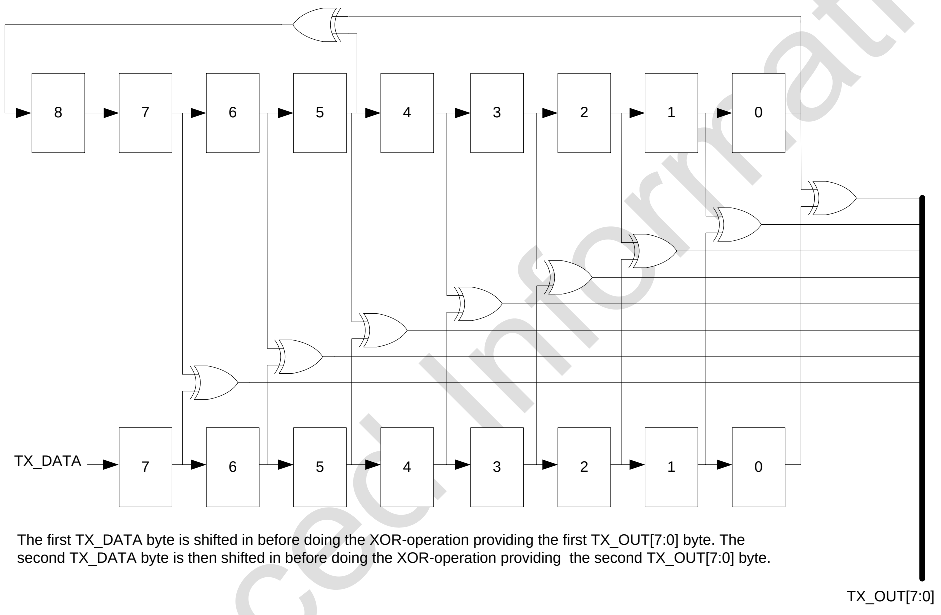

8
7
6
5
4
3
2
1
0
TX_DATA
7
6
5
4
3
2
1
0
The first TX_DATA byte is shifted in before doing the XOR-operation providing the first TX_OUT[7:0] byte. The second TX_DATA byte is then shifted in before doing the XOR-operation providing the second TX_OUT[7:0] byte.
TX_OUT[7:0]

Figure 13: Data Whitening in TX Mode

7.1.7 Data Byte Swap

If the `PKT_CFG1.BYTE_SWAP_EN` register field is set than the bits in each byte are swapped, meaning that bit 0 become 7, bit 1 become bit 6 etc. until bit 7 becomes bit 0.

In TX mode all bytes in the TX FIFO are swapped before optional CRC calculation and whitening are performed.

In RX mode the data byte is swapped after all the processing is done, meaning that de-whitening and CRC calculation are done on the original received data, and the swapping is done right before the actual writing to the RX FIFO. This means that if using address filtering (see Section 7.2.1) is used when `PKT_CFG1.BYTE_SWAP_EN = 1`, the user should swap the address manually in the `DEV_ADDR` register in order to match the received address due to the fact that the packet engine compares the address register to the received address before the swapping is done.

7.1.8 UART Mode

If UART mode is enabled (`PKT_CFG0.UART_MODE_EN = 1`), the packet engine inserts and removes start/stop bits automatically. In this mode the packet engine will emulate UART back-to-back transmissions typically done over an asynchronous RF interface, to enable communication with simple RF devices without packet support.

The start and stop bits are not handled as data, only inserted/removed in the data stream to/from the modem, hence all packet features are available in this mode. The value of the start/stop bits are configurable through the `PKT_CFG0.UART_SWAP_EN` register field.

If whitening is enabled, only the data bits are affected, not the start/stop bits.

A "framing error" occurs in RX mode when the designated "start" and "stop" bits are not received as expected. As the "start" bit is used to identify the beginning of an incoming byte it acts as a reference for the remaining bits. If the "start" and "stop" bits are not in their expected value, it means that the data is not in line also and a Framing Error will occur.
The framing error is different from other errors, it behave more as indication and will not stop the ongoing reception. The framing error can go up and down several times at the same packet.

7.2 Packet Filtering in Receive Mode

CC112X supports three different types of packet-filtering; address filtering, maximum length filtering, and CRC filtering.

7.2.1 Address Filtering

Setting `PKT_CFG1.ADDR_CHECK_CFG` to any other value than zero enables the packet address filtering. The packet handler engine will compare the destination address byte in the packet with the programmed node address in the `DEV_ADDR` register and the 0x00 broadcast address when `PKT_CFG1.ADDR_CHECK_CFG = 10b` or both the 0x00 and 0xFF broadcast addresses when `PKT_CFG1.ADDR_CHECK_CFG = 11b`. If the received address matches a valid address, the packet is received and written into the RX FIFO. If the address match fails, the packet is discarded and the radio controller will either restart RX or go to IDLE dependent on the `RFEND_CFG0.TERM_ON_BAD_PKT_EN` setting (the `RFEND_CFG1.RXOFF_MODE` setting is ignored).

7.2.2 Maximum Length Filtering

In variable packet length mode, `PKT_CFG0.LENGTH_CONFIG = 01`, the `PKT_LEN` register value is used to set the maximum allowed packet length. If the received length byte has a larger value than this, the packet is discarded and the radio controller will either restart RX or go to IDLE dependent on the `RFEND_CFG0.TERM_ON_BAD_PKT_EN` setting (the `RFEND_CFG1.RXOFF_MODE` setting is ignored).

7.2.3 *CRC Filtering*

The filtering of a packet when CRC check fails is enabled by setting FIFO_CFG.CRC_AUTOFLUSH = 1. The CRC auto flush function will only flush the packet received with bad CRC, other packets will remain unchanged in the RX FIFO. After auto flushing the faulty packet, the radio controller will either restart RX or go to IDLE dependent on the RFEND_CFG0.TERM_ON_BAD_PKT_EN setting (the RFEND_CFG1.RXOFF_MODE setting is ignored).

When using the auto flush function, the maximum packet length is 127 bytes in variable packet length mode and 128 bytes in fixed packet length mode. Note that when PKT_CFG1.APPEND_STATUS is enabled, the maximum allowed packet length is reduced by two bytes in order to make room in the RX FIFO for the two status bytes appended at the end of the packet. The MCU must not read from the current packet until the CRC has been checked as OK.

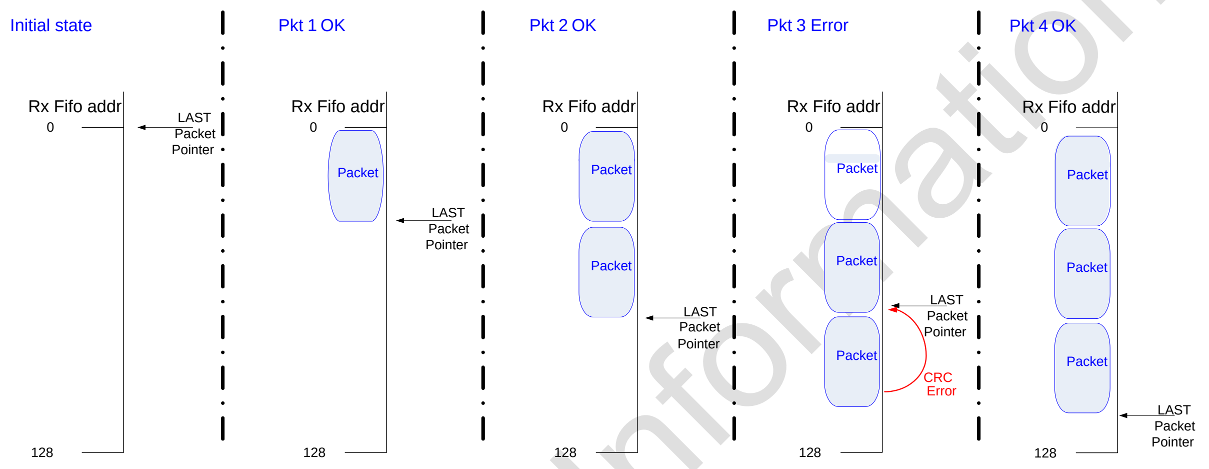


Figure 14: CRC Autoflush of Packet Received with CRC Error

7.3 Packet Handling in Transmit Mode

The payload that is to be transmitted must be written into the TX FIFO. The first byte written must be the length byte when variable packet length is enabled (PKT_CFG0.LENGTH_CONFIG = 01). The length byte has a value equal to the payload of the packet (including the optional address byte). If address recognition is enabled on the receiver (PKT_CFG1.ADDR_CHECK_CFG ≠ 00), the second byte written to the TX FIFO must be the address byte.

If fixed packet length is enabled (PKT_CFG0.LENGTH_CONFIG = 00), the first byte written to the TX FIFO should be the address (assuming the receiver uses address recognition).

The modulator will first send the programmed number of preamble bytes. If data is available in the TX FIFO, the modulator will send the sync word followed by the payload in the TX FIFO. If CRC is enabled, the checksum is calculated over all the data pulled from the TX FIFO, and the result is sent as two extra bytes following the payload data. If the TX FIFO runs empty before the complete packet has been transmitted, the radio will enter TXFIFO_ERROR state. The only way to exit this state is by issuing an SFTX strobe. Writing to the TX FIFO after it has underflowed will not restart TX mode.

If whitening is enabled, everything following the sync words will be whitened. Whitening is enabled by setting PKT_CFG1.WHITE_DATA = 1.

7.4 Packet Handling in Receive Mode

In receive mode, the demodulator and packet handler will search for a valid preamble and sync word. When found, the demodulator has obtained both bit and byte synchronism and will receive the first payload byte.

If whitening is enabled (PKT_CFG1.WHITE_DATA = 1), the data will be de-whitened at this stage.

When variable packet length mode is enabled, the first byte is the length byte. The packet handler stores this value as the packet length and receives the number of bytes indicated by the length byte. If fixed packet length mode is used, the packet handler will accept the programmed number of bytes.

Next, the packet handler optionally checks the address and only continues the reception if the address matches. If automatic CRC check is enabled, the packet handler computes CRC and matches it with the appended CRC checksum.

At the end of the payload, the packet handler will optionally write two extra packet status bytes (see Table 19 and Table 20) that contain CRC status, LQI, and RSSI value.

7.5 Packet Handling in Firmware

When implementing a packet oriented radio protocol in firmware, the MCU needs to know when a packet has been received / transmitted. Additionally, for packets longer than 128 bytes, the RX FIFO needs to be read while in RX and the TX FIFO needs to be refilled while in TX. This means that the MCU needs to know the number of bytes that can be read from or written to the RX FIFO and TX FIFO respectively. There are two possible solutions to get the necessary status information:

a) Interrupt Driven Solution

The GPIO pins can be used in both RX and TX to give an interrupt when a sync word has been received/transmitted or when a complete packet has been received/transmitted by setting `IOCFGx.GPIOx_CFG = PKT_SYNC_RXTX` (6). In addition, there are two configurations for the `IOCFGx.GPIOx_CFG` register that can be used as an interrupt source to provide information on how many bytes are in the RX FIFO and TX FIFO respectively. The `IOCFGx.GPIOx_CFG = RXFIFO_THR` (0) and the `IOCFGx.GPIOx_CFG = RXFIFO_THR_PKT` (1) configurations are associated with the RX FIFO while the `IOCFGx.GPIOx_CFG = TXFIFO_THR` (2) and the `IOCFGx.GPIOx_CFG = TXFIFO_THR_PKT` (3) configurations are associated with the TX FIFO.

b) SPI Polling

The `GPIO_STATUS` register can be polled at a given rate to get information about the current GPIO3, GPIO2, and GPIO0 values. The `NUM_RXBYTES` and `NUM_TXBYTES` registers can be polled at a given rate to get information about the number of bytes in the RX FIFO and TX FIFO respectively. Alternatively, the number of bytes in the RX FIFO and TX FIFO can be read from the chip status byte returned on the MISO line each time a header byte, data byte, or command strobe is sent on the SPI bus (see Section 3.1.2).

7.6 TX FIFO and RX FIFO

The ***CC112X*** contains two 128 byte FIFOs, one for received data and one for transmit data. The SPI interface is used to read from the RX FIFO and write to the TX FIFO. Section 3.2.2 contains details on the SPI FIFO access. The FIFO controller will detect under / overflow in both the RX FIFO and the TX FIFO.

The chip status byte, that is available on the SO pin while transferring the SPI header, contains the fill grade of the FIFOs. Section 3.1.2 contains more details on this. The number of bytes in the RX FIFO and TX FIFO can be read from the status registers `NUM_RXBYTES` and `NUM_TXBYTES` respectively.

If the packet length is larger than 128 bytes, the MCU must determine how many bytes can be read from the RX FIFO. The following software routine can be used:

1. Read `NUM_RXBYTES`; store value in *n*.
2. If *n* < # of bytes remaining in packet, read *n* bytes from the RX FIFO.
3. Repeat steps 1 and 2 until *n* = # of bytes remaining in packet.
4. Read the remaining bytes from the RX FIFO.

A signal will assert when the number of bytes in the FIFO is equal to or higher than a programmable threshold. This signal can be viewed on the GPIO pins and can be used for interrupt driven FIFO routines to avoid polling the `NUM_RXBYTES` register.

The 7-bit `FIFO_CFG.FIFO_THR` setting is used to program threshold points. Table 21 lists the `FIFO_THR` settings and the corresponding thresholds for the RX and TX FIFO. The threshold value is coded in opposite directions for the two FIFOs to give equal margin to the overflow and underflow conditions when the threshold is reached.

`FIFO_THR`	Bytes in TX FIFO	Bytes in RX FIFO
0	127	1
1	126	2
2	125	3
...	...	...
126	1	127
127	0	128

Table 21: `FIFO_THR` Settings and the Corresponding FIFO Thresholds

To simplify debug and advanced FIFO features, the full FIFO buffer is memory mapped and can be accessed directly with SW. Both FIFO content and FIFO data pointers are accessible. This can be used to significantly reduce the SPI traffic, see examples below

1. In a hostile RF environment packets are lost and re-transmissions are often required. Normally the packet data must then be written again over the SPI interface. By using the direct FIFO access feature and changing the `TXFIRST` register to point to the head of the previous message, a re-transmission can be done without writing the packet over the SPI.
2. In many protocols only parts of the message is changed between each transmission (e.g. changing a read value from a sensor, incrementing a transmission counter). Direct FIFO access can then be used to change only the new data (the FIFOs are reached through the 0x3E command, see Table 3), leaving the rest of the data unchanged. FIFO data pointers (`TXFIRST` and `TXLAST`) can then be manipulated to re-transmit the packet with changed data.

7.7 Transparent and Synchronous Serial Operation

Several features and modes of operation have been included in the ***CC112X*** to provide backward compatibility with legacy systems that cannot be supported by the built in packet handling functionality. For new systems, it is recommended to use the built-in packet handling features, as they give more robust communication, significantly offload the microcontroller, and simplify software development.

There are two serial modes and they are described in the two following sections.

7.7.1 Synchronous Serial Mode

In the synchronous serial mode, data is transferred on a two-wire serial interface. The ***CC112X*** provides a clock (`IOCFGx.GPIOx_CFG = SERIAL_CLK` (8)) that is used to set up new data on the data input line or sample data on the data output line. Data timing recovery is done automatically. The data pin is updated on the falling edge of the clock pin at the programmed data rate. Preamble and sync word insertion / detection may or may not be active, depending on the sync mode.

Define a GPIO pin to be serial data clock for both TX and RX operation. For RX operation, another GPIO pin has to be defined as serial data output (`IOCFGx.GPIOx_CFG = SERIAL_RX` (9)). For TX operation, the GPIO0 pin is explicitly used as serial data input. This is automatically done when in TX. In order to avoid internal I/O conflict, GPIO0 should be defined as tri state (`IOCFGx.GPIOx_CFG = HIGHZ` (48)). In synchronous serial mode the TX data must be set up on the falling edge of the data clock output from ***CC112X***.

7.7.2 Transparent Serial Mode

CC112X does not do any timing recovery and just outputs the hard limited baseband signal. In transparent serial mode the data rate programming does not affect operation. When transparent mode is enabled, the device is set up to resemble a legacy purely analog front end device with baseband output to support legacy pulse position modulation, PWM modulated signals etc. In transparent mode the signal is digitally demodulated and output on GPIO pins through a programmable interpolation filter (MDMCFG0.TRANSPARENT_INTFACT). The interpolation filter is used to eliminate jitter on the baseband signal. This is useful when using external DSP demodulators as jitter introduce noise in the demodulation process. The rate on the GPIO is ((4 x receiver bandwidth) x interpolation factor).

When using transparent serial mode make sure the interfacing MCU/DSP does proper oversampling. In transparent serial mode, several of the support mechanisms for the MCU that are included in CC112X will be disabled, such as packet handling hardware, buffering in the FIFO, and so on.

Preamble and sync word insertion/detection is not supported in transparent serial mode.

For TX operation, the GPIO0 pin is explicitly used as serial data input. This is automatically done when in TX. In order to avoid internal I/O conflict, GPIO0 should be defined as tri state (decimal 48) or RX data output.

Data output can be observed on GPIO0/1/2/3. This is set by the IOCFGx.GPIOx_CFG fields.

8 Radio Control

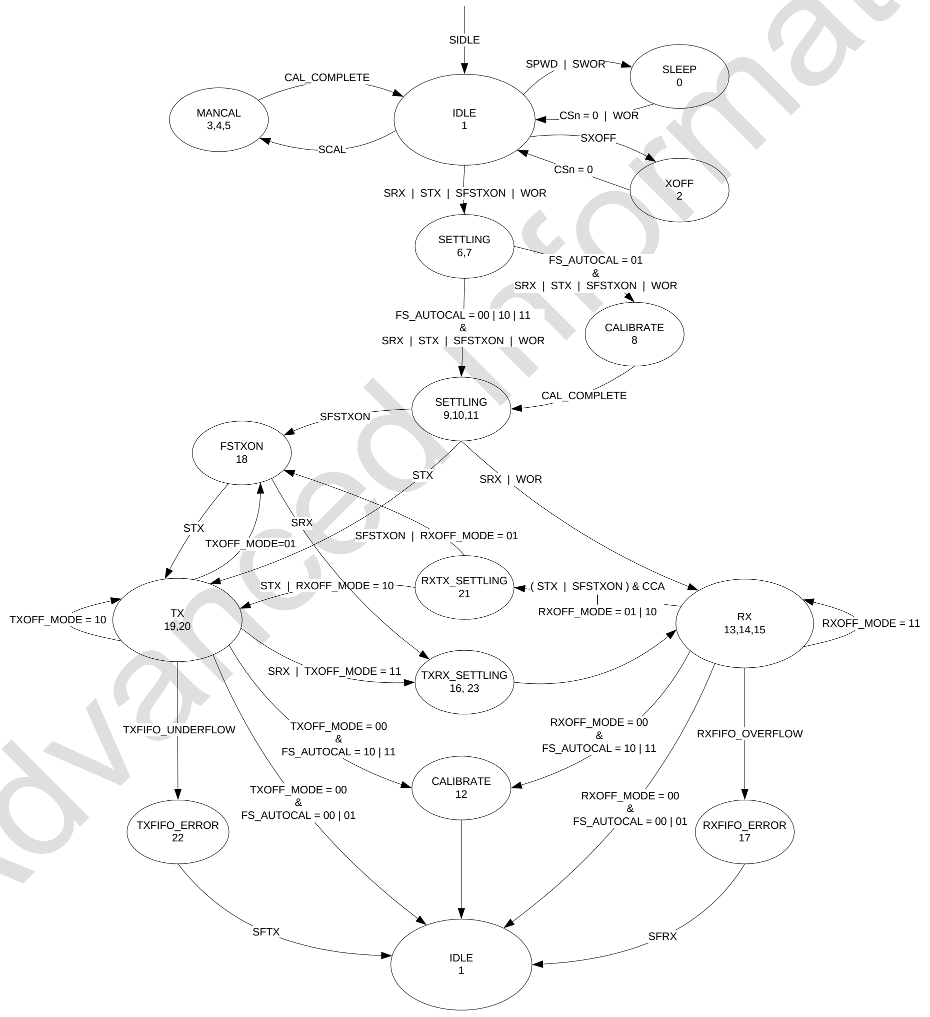


Figure 15: Complete Radio Control State Diagram

CC112X has a built-in state machine that is used to switch between different operational states (modes). The change of state is done either by using command strobes or by internal events such as TX FIFO underflow.

A simplified state diagram is shown in Figure 2 and the complete radio control state diagram is shown in Figure 15. The numbers refer to the state number readable in the MARCSTATE.MARC_STATE register.

8.1 Power-On Start-Up Sequence

When the power supply is turned on, the system must be reset. This is achieved by one of the two sequences described below, i.e. automatic power-on reset (POR) or manual reset.

8.1.1 Automatic POR

A power-on reset circuit is included in the **CC112X**. The internal power-up sequence is completed when `CHIP_RDYn` goes low. `CHIP_RDYn` is observed on the SO pin after CSn is pulled low (See Section 3.1.2 for more details).

When the **CC112X** reset is completed, the chip will be in the IDLE state and the crystal oscillator will be running. If the chip has had sufficient time for the crystal oscillator to stabilize after the power-on-reset, the SO pin will go low immediately after taking CSn low. If CSn is taken low before reset is completed, the SO pin will first go high, indicating that the crystal oscillator is not stabilized, before going low as shown in Figure 16.

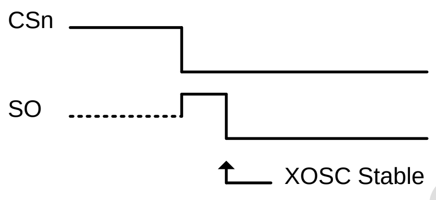


Figure 16: Power-On Reset

8.1.2 Manual Reset

The other reset possibilities on the **CC112X** are issuing using the `SRES` command strobe or using the RESET_N pin. By issuing a manual reset, all internal registers are set to their default values and the radio will enter IDLE state.

8.2 Crystal Control

The crystal oscillator (XOSC) is either automatically controlled or always on, if `XOSC2.XOSC_CORE_PD_OVERRIDE = 1`. If the XOSC is forced on, the crystal will always stay on even in the SLEEP state.

`XOSC2.XOSC_CORE_PD_OVERRIDE = 0`, the XOSC will be turned off if the `SXOFF`, `SPWD`, or `SWOR` command strobes are issued; the state machine then goes to XOFF or SLEEP respectively. This can only be done from the IDLE state. The XOSC will be automatically turned on again when CSn goes low, and the MARC will go to the IDLE state. The SO pin on the SPI interface must be pulled low before the SPI interface is ready to be used as described in Section 3.1.2.

Crystal oscillator start-up time depends on crystal ESR and load capacitances.

8.3 Voltage Regulator Control

The voltage regulator to the digital core is controlled by the radio controller. When the chip enters the SLEEP state which is the state with the lowest current consumption, the voltage regulator is disabled. This occurs after CSn is released when a `SPWD` command strobe has been sent on the SPI interface. The chip is then in the SLEEP state. Setting CSn low again will turn on the regulator and crystal oscillator and make the chip enter the IDLE state.

When enhanced Wake on Radio is enabled, the eWOR module will control the voltage regulator as described in Section 8.6.

8.4 Active Modes

CC112X has two active modes: receive and transmit. These modes are activated directly by the MCU by using the `SRX` and `STX` command strobes, or automatically by eWOR (RX mode).

The MCU can manually change the state from RX to TX and vice versa by using the command strobes. If the radio controller is currently in transmit and the `SRX` strobe is used, the current transmission will be ended and the transition to RX will be done. If the radio controller is in RX when the `STX` or `SFSTXON` command strobes are used, the TX-on-CCA function will be used. If the channel is not clear, the chip will remain in RX. The `PKT_CFG2.CCA_MODE` setting controls the conditions for clear channel assessment.

The `SIDLE` command strobe can always be used to force the radio controller to go to the IDLE state.

8.4.1 RX

When RX is activated, the chip will remain in receive mode until:

- A packet is received
- An `SIDLE, SRX, STX`, or `SFSTXON` command strobe is being issued
- The RX FIFO overflows / underflows
- The RX termination timer expires
- A CS or PQT based termination takes place

When a packet is successfully received, the radio controller goes to the state indicated by the `RFEND_CFG1.RXOFF_MODE` setting, i.e. IDLE, FSTXON, TX or RX.

Note: When `RFEND_CFG1.RXOFF_MODE = 11b` and a packet has been received, it will take some time before RSSI is valid again even if the radio has never exited RX mode. This time is the same as the RSSI response time.

When a bad packet is received (packet length/address/CRC error) the radio controller will either restart RX, or go to IDLE dependent on the `RFEND_CFG0.TERM_ON_BAD_PKT_EN`.

When RX is terminated due to RX termination, the radio controller will go to IDLE through FS calibration dependent on the `SETTLING_CFG.FS_AUTOCAL` setting.

8.4.2 TX

Similarly, when TX is active the chip will remain in the TX state until:

- The current packet has been transmitted
- An `SIDLE` or `SRX` command strobe is being issued
- The TX FIFO overflows / underflows

When a packet is successfully transmitted, the radio controller goes to the state indicated by the `RFEND_CFG0.TXOFF_MODE` setting. The possible destinations are the same as for RX.

8.5 RX Termination

The RX termination timer can be activated with RX terminate on CS or RX terminate on PQT. If RX terminates due to RX termination on RSSI/PQT or RX termination timer, the chip will always go back to IDLE if eWOR is disabled and back to SLEEP if eWOR is enabled.

8.5.1 RX Termination Timer

CC112X has optional functions for automatic termination of RX after a programmable time. The main use for this functionality is Wake on Radio, but it is useful for other applications as it reduces the need for a dedicated MCU timer. The termination timer starts when the chip has entered RX state, and the timeout is programmable with the `RFEND_CFG1.RX_TIME` setting. When the timer expires, the radio controller will check the condition for staying in RX.

The programmable conditions are:

- `RFEND_CFG1.RX_TIME_QUAL = 0`: Continue receive if SYNC word has been found
- `RFEND_CFG1.RX_TIME_QUAL = 1`: Continue receive if SYNC word has been found, or if PQT is reached or CS is asserted.

The following formula can be used to set the RX timeout:

$$RX\_TIMEOUT = MAX\left[1, FLOOR\left[\frac{REG\_EVENT0}{2^{REG\_RX\_TIME+3}}\right]\right] \cdot 2^{4 \cdot WOR\_RES} \cdot \frac{1250}{f_{XOSC}}$$

8.5.2 RX Termination Based on CS

If `RFEND_CFG0.ANT_DIV_RX_TERM_CFG = 001b` the device will use the first RSSI sample to determine if a carrier is present on the channel. If no carrier is present, RX will terminate. The RSSI samples are continually evaluated, and if the RSSI level falls below the threshold RX will terminate if not SYNC is found. This can be used to achieve very low power by reducing the time in active RX to a minimum.

8.5.3 RX Termination Based on PQT

If `RFEND_CFG0.ANT_DIV_RX_TERM_CFG = 100b` the device will use the PQT indication to determine if RX mode should be terminated or not. If no preamble is detected, RX will be terminated. The PQT indication is continually evaluated, and if the PQT level falls below the threshold RX will terminate if no SYNC is found. This can be used to achieve very low power by reducing the time in active RX to a minimum.

8.6 Enhanced Wake on Radio (eWOR)

The optional enhanced Wake on Radio (eWOR) functionality enables **CC112X** to periodically wake up from SLEEP and listen for incoming packets without MCU interaction.

When the `SWOR` strobe command is sent on the SPI interface, the **CC112X** will go to the SLEEP state when CSn is released. The RC oscillator must be enabled by setting `WOR_CFG0.RC_PD = 0` before the `SWOR` strobe can be used, as it is the clock source for the eWOR timer. The on-chip eWOR timer will set **CC112X** into IDLE state and then RX state. After a programmable time in RX, the chip will go back to the SLEEP state, unless a packet is received.

To exit eWOR mode, issue an `SIDLE` strobe command.

The on-chip eWOR timer will be clocked whenever the RC oscillator is running, independent of eWOR mode enabled or not. Hence the timing is preserved, when eWOR mode is exited. The on-chip eWOR timer can be reset to the Event 1 value, by issuing the `SWORRST` strobe command.

CC112X can be set up to signal the MCU that a packet has been received by using the GPIO pins. When the MCU has read the packet, it can put the chip back into SLEEP with the `SWOR` strobe from the IDLE state.

8.6.1 WOR EVENT 0, 1, and 2

The eWOR timer has three events, Event 0, Event 1, and Event 2. In the SLEEP state with eWOR activated, reaching Event 0 will turn on the digital regulator and start the crystal oscillator. Event 1 follows Event 0 after a programmed timeout (t_{Event1}). Figure 17 shows the timing relationship between Event 0 timeout and Event 1 timeout.

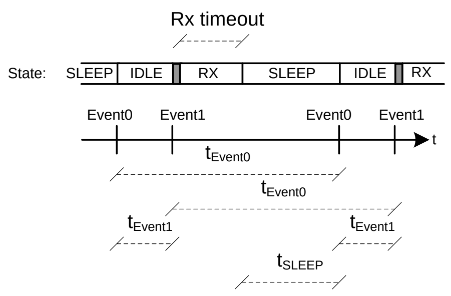


Figure 17: Event 0 and Event 1 Relationship

The time between two consecutive Event 0's is programmed with a mantissa value given by `WOR_EVENT0_MSB` and `WOR_EVENT0_LSB` and an exponent value set by `WOR_CFG1.WOR_RES`. The equation is:

$$t_{Event0} = \frac{1}{f_{RCOSC}} \cdot EVENT0 \cdot 2^{5*WOR\_RES}$$

The Event 1 timeout is programmed with a mantissa value decoded by the `WOR_CFG1.EVENT1` setting.

WOR_CFG1.EVENT1	**WOR_EVENT1**	**t_{EVENT1} [µs] (f_{RCOSC} = 32 kHz)**
000	4	125
001	6	187.5
010	8	250
011	12	375
100	16	500
101	24	750
110	32	1000
111	48	1500

Table 22: Event 1

The equation for the Event 1 timeout is:

$$t_{Event1} = \frac{1}{f_{RCOSC}} \cdot WOR\_EVENT1$$

Event 2 can used to autonomously take the system out of SLEEP at regular intervals to perform RC oscillator calibration. This will improve the accuracy of the timer.

The Event 2 timing is programmed with an exponent value decoded by the `WOR_CFG0.EVENT2_CFG` setting

WOR_CFG0.EVENT2_CFG	**WOR_EVENT2**	**t_{Event2} [s] (f_{RCOSC} = 32 kHz)**
00	Disabled	NA
01	15	1
10	18	8
11	21	65

Table 23: Event 2

t_{Event2} is given by:

$$t_{Event2} = \frac{2^{WOR\_EVENT2}}{f_{RCOSC}}$$

Note that the timing period is reset at Event 0.

8.7 RC Oscillator Calibration

The frequency of the low-power RC oscillator used for the eWOR functionality varies with temperature and supply voltage. In order to keep the frequency as accurate as possible, the RC oscillator should be calibrated whenever possible. Two automatic RC calibration options are available that are controlled by the `WOR_CFG0.RC_MODE` setting:

- RC calibration is enabled when the XOSC is running and the device is not in SLEEP.
- RC calibration is enabled on every 4th time the device is powered up and goes from IDLE to RX/TX

During calibration the eWOR timer will be clocked on a down-divided version of the XOSC clock. When the chip goes to the SLEEP state, the RC oscillator will use the last valid calibration result. The frequency of the RC oscillator is calibrated to the main crystal frequency divided by 1000.

In applications where the radio wakes up very often, typically several times every second, it is possible to do the RC oscillator calibration once and then turn off calibration to reduce the current consumption. If the RC oscillator calibration is turned off, it will have to be manually turned on again to resume calibration. This will be necessary if temperature and supply voltage changes to maintain accuracy.

When the `WOR_CFG0.RC_MODE` setting is altered, an `SIDLE` command strobe must be issued before the new configuration is taken into account.

8.8 MCU Wakeup

The MCU wake up pulse (`IOCFGx.GPIOx_CFG = MARC_MCU_WAKEUP` (20)) should be used to initiate an interrupt to the MCU. The main purpose of the MCU wake up feature is to wake up the MCU from sleeping mode at the right time, i.e. when there is a need of intervention from the MCU side. The following list describes all the case that MARC will initiate MARC Wake up pulse:

1. On any good packet **reception** completion -> Indicates to the MCU that the packet is ready to be read and it is already in the RX FIFO.
2. On any good packet **transmission** completion -> Indicates to the MCU that the packet was transmitted and the device is ready for the next operation.
3. RX timeout or RX terminated based on RSSI/PQT (no eWOR) - Indicates to the MCU that RX mode has been terminated and no packet has been received.
4. RX FIFO error occurred -> Indicate to the MCU that the received packet size exceeded the free space left in the RX FIFO or the MCU read too many bytes from RX FIFO and got it to underflow. The MCU should flush the RX FIFO and restart RX mode.
5. TX FIFO error occurred -> Indicate to the MCU that the packet stored in the TXFIFO was not long enough with respect to the number of bytes needed to be transmitted or the MCU exceeded the free space left in the TX FIFO when writing to it. The MCU should flush the TX FIFO and restart the transmission.
6. One of the packet engine errors occurred (address / length / CRC) and configuration `RFEND_CFG0.TERM_ON_BAD_PKT = 1`. Note: On eWOR Normal & Feedback modes (with an exception of eWOR Legacy mode) the wake up pulse will not be asserted and the Device will go to sleep until next time slot.

7. MARC is set to eWOR Feedback mode (`WOR_CFG1.WOR_MODE[2:0] = 000`) and the bad reception counter reached its maximum (16 slots with no successful reception) -> indicate to the MCU that probably the time slot which was selected is out of sync and it should be re-programmed.
8. While in RX mode, when issuing an STX and TX mode is not entered due to an unclear channel (CCA not asserted) -> indicate to the MCU that TX request failed. In parallel the MARC also asserts TXONCCA_DONE and TXONCCA_FAILED (can be monitored on the GPIO pins or read from `MARC_STATUS0`).

The following list describes all the case that MARC will NOT initiate a MARC Wake up pulse:

1. While in TX mode, MCU strobe for RX and MARC response and switch to RX. User should know in this case that TX is finished and the Wake up will not be asserted.
2. While in RX mode, MCU strobe for TX and MARC response by switching to TX in case of a clear channel (CCA asserted). In this case the Wake up signal will not be asserted. If the user needs to know quickly that MARC switched to TX, the user can monitor the TXONCCA_DONE and TXONCCA_FAILED signals (can be monitored on the GPIO pins or read from `MARC_STATUS0`). Eventually the user will receive a wake up interrupt when TX finishes.
3. One of the packet engine errors occurred (address / length / CRC) and configuration `RFEND_CFG0.TERM_ON_BAD_PKT_EN = 0`. In this case MARC will return to RX without waking up the MCU.
4. One of the packet engine errors occurred (address / length / CRC) and eWOR (Normal & Feedback mode) is enabled. The MARC will return to IDLE and then to power down without pulsing the wake up.
5. RX timeout or RX terminated based on RSSI/PQT occurred (eWOR mode enabled) - The wake up pulse will not be asserted and the device will go to sleep until next time slot.

In all the cases that the MARC is heading directly for IDLE state (not switching RX/TX or calibrating), the MCU wake up pulse will be asserted when MARC is already in IDLE, saving the MCU effort of ensuring IDLE state before next operation.

By setting `RFEND_CFG0.CAL_END_WAKE_UP_EN = 1`, the MCU set the MARC to give additional wake up pulse on each transition to IDLE when calibration has finished (`MARC_STATUS_OUT` will be 0x00).

If the MCU is required to intervene, this signal is pulsed and the MCU can read the `MARC_STATUS1.MARC_STATUS_OUT` to find cause of wake up event and take appropriate action.

MARC_STATUS_OUT	Description
00000000	No failure
00000001	RX timeout occurred (Only valid in RX mode)
00000010	Direct RX Terminated occurred based on RSSI or PQT (Only valid in RX mode)
00000011	eWOR sync lost -16 slots with no successful reception (Only valid in Enhanced eWOR mode)
00000100	Packet engine length failed (Only valid in RX mode)
00000101	Packet engine address failed (Only valid in RX mode)
00000110	Packet engine CRC failed (Only valid in RX mode)
00000111	TX FIFO overflow error occurred
00001000	TX FIFO underflow error occurred
00001001	RX FIFO overflow error occurred
00001010	RX FIFO underflow error occurred
00001011	TX ON CCA failed - STX strobe ignored due to a busy channel
01000000	TX finished successfully
10000000	RX finished successfully

Table 24: MARC_STATUS_OUT

8.9 RX Sniff Mode

- For battery operated systems the active receive current is an important feature for battery lifetime
- To increase battery lifetime a novel Channel Sniff Mode feature has been designed for the CC112X family to **autonomously sniff for RF activity** using a low power algorithm
- The Channel Sniff Mode algorithm **maintains the range, robustness** to interference and coexistence performance of the device
- The average receive current can be reduced by 75% by increasing the length of the preamble
- The ***CC112X*** platform is designed for extremely quick settling time. The short settling time can be used for lower power operation by being able to quickly turn on and off the receiver
- Only turning on and off does not include the full picture, also the ability quickly and reliably detect if there is RF activity or not is a key parameter.
- The ***CC112X*** family of devices use strong DSP logic to detect the sync word without the need for preamble bits for settling. The preamble is only needed for AGC settling, i.e. settling the gain of the frontend
- The ***CC112X*** receiver requires only 4 bit preamble for settling incl. frequency offset compensation (AFC)

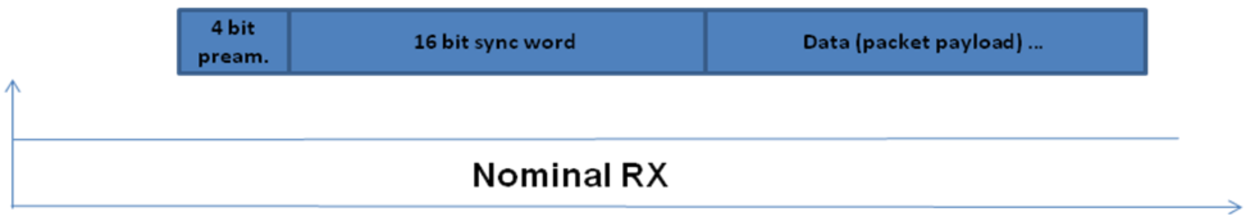


Figure 18: Ordinary RX Mode

- In Advanced Channel Sniff Mode the quick settling is combined with a longer preamble to run automated duty cycling of the receiver
- The duty cycling is **transparent to the user** and does not impact the RF performance (sensitivity, selectivity, robustness etc.)
- Average power consumption depends on data rate and preamble length
    - Example: 1.2 kbps with 4 byte preamble gives 3 mA average receive current

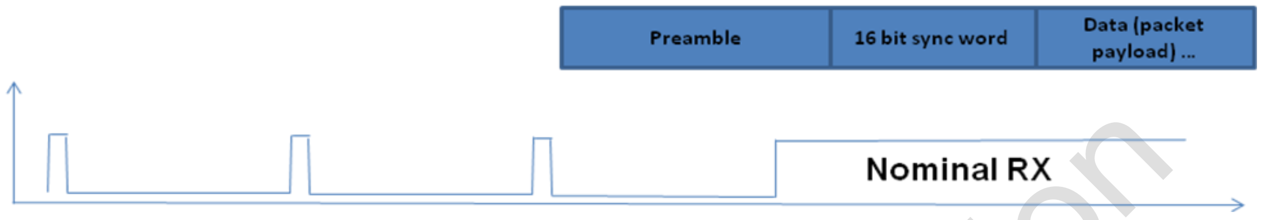


Figure 19: RX Sniff Mode

8.10 Antenna Diversity

CC112X has two different antenna diversity modes: Single-switch mode and continuous-switch mode.

Single switch mode is useful for very low power schemes where the device only checks each antenna once for a signal and directly returns to power down if a signal is not detected (automated using the eWOR feature).If a signal is found on the first antenna checked, it does not check the second antenna.

Continuous mode is useful when staying in RX for longer intervals.

The antenna diversity algorithm can operate based on PQT or CS. The user can configure how the device will act in regards to antenna diversity and RX termination as described in Table 25.

RFEND_CFG0.ANT_DIV_RX_TERM_CFG	Description
000	Antenna diversity and termination based on CS/PQT are disabled
001	RX termination base on CS is enabled (Antenna diversity OFF). See 8.5.2 for details.
010	Single-switch antenna diversity on CS enabled. One or both antenna is CS evaluated once and RX will terminate if CS failed on both antennas.
011	Continuous-switch antenna diversity on CS enabled. Antennas are switched until CS is asserted or RX timeout occurs (if RX timeout is enabled)
100	RX termination base on PQT is enabled (Antenna diversity OFF). See 8.5.3 for details.
101	Single-switch antenna diversity on PQT enabled. One or both antenna is PQT evaluated once and RX will terminate if PQT is not reached on any of the antennas.
110	Continuous-switch antenna diversity on PQT enabled. Antennas are switched until PQT is reached or RX timeout occurs (if RX timeout is enabled)
111	Reserved

Table 25: `RFEND_CFG0.ANT_DIV_RX_TERM_CFG` Setting

8.10.1 Antenna Diversity Features

The device supports antenna diversity by controlling an external RF switch using the ANTENNA_SELECT control signal available on GPIO (`IOCFGx.GPIOx_CFG = ANTENNA_SELECT` (36)).

The device will remember the last antenna used (when not entering SLEEP mode[2]) and use the last antenna for the next RX or TX transaction. Staying with the same antenna will make sure:

- In RX, that the last antenna used for good reception will be the first one to be checked (minimize time for the next reception)
- In TX, the device will transmit acknowledge with the same antenna that received the packet.

9 Frequency Synthesizer Configuration

If any frequency programming registers are altered when the frequency synthesizer is running, the synthesizer may give an undesired response. Hence, the frequency programming should only be updated when the radio is in the IDLE state.

9.1 RF Programming

RF programming in **CC112X** is given by two factors; the VCO frequency programming and the LO divider programming (RF band selection). The relation is given in the equation below.

$$f_{RF} = \frac{f_{VCO}}{LO\_DIV}$$

VCO frequency programming is given by the 24 bit (unsigned) frequency word `FREQ` located in the `FREQ2`, `FREQ1`, and `FREQ0` registers. There is also a possibility to perform VCO frequency offset programming, given by the 16 bit (signed) frequency offset word `FREQOFF` located in the `FREQOFF1` and `FREQOFF0` registers. This is intended to adjust for crystal intolerance or fine adjustments of the RF programming.

$$f_{VCO} = \frac{FREQ}{2^{16}} \cdot f_{XOSC} + \frac{FREQOFF}{2^{18}} \cdot f_{XOSC}$$

Note that the `FREQOFF` programming and `FREQOFF_EST` (found in `FREQOFF_EST1` and `FREQOFF_EST0`) have identical formats hence the frequency estimate can be accumulated directly to the `FREQOFF` programming. This can be done either manually or automatically through the `SAFC` command strobe. When a `SAFC` command strobe is issued, the `FREQOFF` value is accumulated with the `FREQEST` value when the modem is disabled meaning that the `FREQOFF` value can be updated automatically when going from RX to TX.

The band select decoding is displayed in Table 26.

[2] In SLEEP mode the GDIOx pin will be hardwired to 0 or 1 depending on which GDIO pin is used and what the value of `IOCFGx_GPIOx_INV` is. Please see Section 3.4 for more details.

FS_CFG.FSD_BANDSELECT	LO_DIV	RF Band [MHz]
0000 - 0001	-	Not in use
0010	4	820 - 960
0011	-	Not in use
0100	8	410 - 480
0101	-	Not in use
0110	12	273.3 - 320
0111	-	Not in use
1000	16	205 - 240
1001	-	Not in use
1010	20	164 - 192
1011	24	136.7 - 160
1100 - 1111	-	Not in use

Table 26: RF Band Selection Decoding

Since the RF band is determined by the LO divider setting, the different RF bands will also have different frequency resolution. Note that the frequency offset word is related to the VCO frequency programming, and hence any crystal inaccuracy compensation is therefore independent of the selected RF band.

See Table 27 for an overview of the RF resolution.

	RF Programming Resolution [Hz]	
RF Band [MHz]	**XOSC @ 32 MHz**	**XOSC @ 40 MHz**
820 - 960	30.5	38.1
410 - 480	15.3	19.1
273.3 - 320	10.2	12.7
205 - 240	7.6	9.5
164 - 192	6.1	7.6
136.7 - 160	5.1	6.4

Table 27: RF Programming Resolution

9.2 IF Programming

The IF frequency is given by the equation below (`FREQ_IF` is found in the `FREQ_IF_CFG` register).

$$f_{IF} = \frac{FREQ\_IF}{2^{15}} \cdot f_{XOSC}$$

Note that `FREQ_IF` is given in two's complement format, hence both positive and negative IF are supported.

9.3 FS Calibration

The internal on-chip FS characteristics will vary with temperature and supply voltage changes as well as the desired operating frequency. In order to ensure reliable operation, **CC112X** includes frequency synthesizer self-calibration circuitry. This calibration should be done regularly, and must be performed after turning on power and before using a new radio frequency.

CC112X has one manual calibration option (using the `SCAL` strobe), and three automatic calibration options that are controlled by the `SETTLING_CFG.FS_AUTOCAL` setting:

- Calibrate when going from IDLE to either RX or TX (or FSTXON)
- Calibrate when going from either RX or TX to IDLE automatically
- Calibrate every fourth time when going from either RX or TX to IDLE automatically

If the radio goes from TX or RX to IDLE by issuing an `SIDLE` strobe, calibration will not be performed.

Note: The calibration values are maintained in SLEEP mode, so the calibration is still valid after waking up from SLEEP mode unless supply voltage or temperature has changed significantly.

9.4 FS Out of Lock Detection

To check whether the PLL is out of lock, the user can enable the lock detector through the `FS_LOCK_EN` bit in the `FS_CFG` register. The lock indication can either be read through `FSCAL_CTRL.LOCK` or set available on GDO0 or GDO1 (`IOCFG0/1.GPIO0/1_CFG = PLL_LOCK` (35)) as an interrupt to the MCU. A negative transition on the lock indication indicates that the FS is out of lock.

10 System Considerations and Guidelines

10.1 Voltage Regulators

CC112X contains several on-chip linear voltage regulators that generate the supply voltages needed by the low-voltage modules. These voltage regulators are invisible to the user, and can be viewed as integral parts of the various modules. The user must however make sure that the absolute maximum ratings and required pin voltages are not exceeded.

By setting the CSn pin low, the voltage regulator to the digital core turns on and the crystal oscillator starts. The SO pin on the SPI interface must go low before the first positive edge of SCLK (setup time is given in Table 1).

If the chip is programmed to enter power-down mode (`SPWD` or `SWOR` strobe issued), the power will be turned off after CSn goes high. The power and crystal oscillator will be turned on again when CSn goes low.

The voltage regulator for the digital core requires one external decoupling capacitor.

The voltage regulator output should only be used for driving the ***CC112X***.

10.2 SRD Regulations

International regulations and national laws regulate the use of radio receivers and transmitters. Short Range Devices (SRDs) for license free operation below 1 GHz are usually operated in the 433 MHz, 868 MHz, or 915 MHz frequency bands. The ***CC112X*** is specifically designed for operation in these bands.

Please note that compliance with regulations is dependent on the complete system performance. It is the customer's responsibility to ensure that the system complies with regulations.

10.3 Frequency Hopping and Multi-Channel Systems

The 433 MHz, 868 MHz, or 915 MHz bands are shared by many systems both in industrial, office, and home environments. It is therefore recommended to use frequency hopping spread spectrum (FHSS) or a multi-channel protocol because frequency diversity makes the system more robust with respect to interference from other systems operating in the same frequency band. FHSS also combats multipath fading.

CC112X is highly suited for FHSS or multi-channel systems due to its agile frequency synthesizer and effective communication interface. Using the packet handling support and data buffering is also beneficial in such systems as these features will significantly offload the host controller.

Charge pump current, VCO current, and VCO capacitance array calibration data is required for each frequency when implementing frequency hopping for ***CC112X***. There are 2 ways of obtaining the calibration data from the chip:

1) Frequency hopping with calibration for each hop.

2) Fast frequency hopping without calibration for each hop can be done by performing the necessary calibrating at startup and saving the resulting calibration registers in MCU memory. Between each

frequency hop, the calibration process can then be replaced by writing the calibration values that corresponds to the next RF frequency.

The recommended settings change with frequency. This means that one should always use SmartRF Studio to get the correct settings for a specific frequency before doing a calibration, regardless of which calibration method is being used.

10.4 Wideband Modulation when not Using Spread Spectrum

Digital modulation systems under FCC part 15.247 include 2-FSK, GFSK, and 4-FSK modulation. A maximum peak output power of 1 W (+30 dBm) is allowed if the 6 dB bandwidth of the modulated signal exceeds 500 kHz. In addition, the peak power spectral density conducted to the antenna shall not be greater than +8 dBm in any 3 kHz band.

Operating at high data rates and frequency separation, the CC112X is suited for systems targeting compliance with digital modulation system as defined by FCC part 15.247. An external power amplifier is needed to increase the output above +14 dBm.

10.5 Continuous Transmissions

In data streaming applications, the CC112X opens up for continuous transmissions at up to 100 ksps effective data rate. As the modulation is done with a closed loop PLL, there is no limitation in the length of a transmission (open loop modulation used in some transceivers often prevents this kind of continuous data streaming and reduces the effective data rate).

10.6 Battery Operated Systems

In low power applications, the SLEEP state with the crystal oscillator core switched off should be used when the CC112X is not active. It is possible to leave the crystal oscillator core running in the SLEEP state if start-up time is critical. The eWOR functionality should be used in low power applications.

11 Soldering Information

The recommendations for lead-free reflow in IPC/JEDEC J-STD-020 should be followed.

12 Development Kit Ordering Information

Orderable Evaluation Module	Description
CC1120DK	Performance Line Development Kit
CC1120EMK-169	CC1120 Evaluation Module Kit 169 MHz
CC1120EMK-420-470	Evaluation Module Kit 420-470 MHz
CC1120EMK-868-915	CC1120 Evaluation Module Kit 868-915 MHz
CC1120EMK-955	CC1120 Evaluation Module 955 MHz
CC1121EMK-868-915	CC1121 Evaluation Module Kit 868-915 MHz

Table 28: Development Kit Ordering Information

13 References

[1] EN 300 220 V2.3.1: Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1000 MHz frequency range with power levels ranging up to 500 mW" (www.etsi.org)

14 General Information

14.1 Document History

Revision	Date	Description/Changes
SWRU295	30.06.2011	Advanced Information

Table 29: Document History

Products		**Applications**	
Audio	www.ti.com/audio	Communications and Telecom	www.ti.com/communications
Amplifiers	amplifier.ti.com	Computers and Peripherals	www.ti.com/computers
Data Converters	dataconverter.ti.com	Consumer Electronics	www.ti.com/consumer-apps
DLP® Products	www.dlp.com	Energy and Lighting	www.ti.com/energy
DSP	dsp.ti.com	Industrial	www.ti.com/industrial
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Security	www.ti.com/security
Logic	logic.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Power Mgmt	power.ti.com	Transportation and Automotive	www.ti.com/automotive
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com	Wireless	www.ti.com/wireless-apps
RF/IF and ZigBee® Solutions	www.ti.com/lprf		
		TI E2E Community Home Page	e2e.ti.com