Continuous Data Streaming Applications – Using Serial Synchronous Mode in CC110x and CC2500 Devices

By Suyash Jain

Keywords

- *CC1100*
- *CC1100E*
- *CC1101*
- *CC1150*
- *CC2500*
- *CC2550*
- *Serial Synchronous Mode*

1 Introduction

Numerous applications require transferring data continuously over a wireless channel, where the source of the data can be information from some sensor, digitized audio data, etc., that needs to be transmitted continuously. In these applications, the CC110x (CC1100, CC1100E and CC1101) and CC2500 wireless transceivers and CC1150 and CC2550 RF-transmitters can be configured in serial synchronous mode, as described in this document. In serial synchronous mode the data is transmitted or received serially over a two wire interface. This application note describes the register settings to configure the radio in serial synchronous mode and the considerations to minimize the packet error rate (PER) for continuous data streaming applications. The note concludes by providing example register settings for the configurations illustrated.

The software example library [10] gives an "ex_serial_link" example describing how to implement the serial synchronous mode in CC110x or CC2500 transceivers.

Table of Contents

2 Abbreviations

ASK	Amplitude Shift Keying
FEC	Forward Error Correction
GDO	General Digital Output
HW	Hardware
kBaud	kilo Baud
kbps	kilo bits per second
kHz	kilo Hertz
MCU	Micro Controller Unit
MHz	Mega Hertz
MSK	Minimum Shift Keying
PER	Packet Error Rate
RX	Receive
TI	Texas Instruments
TX	Transmit

3 Serial Synchronous Mode

Setting the register bit `PKTCTRL0.PKT_FORMAT=1` in the CC2500 or CC110x radios enables serial synchronous mode. In serial synchronous mode the data is transmitted or received serially over a two wire serial interface. The GDOx pins of the radio are programmed to transfer the clock (referred as clock pin in this document) and data (referred as data pin in this document) between the radio and the MCU interfaced to it. Note that in both cases, i.e., TX and RX, clock is provided by the radio. When TX is active, the data input to the radio is always on the GDO0 pin, and this pin is automatically configured as an input pin. For transmit only devices, CC1150 and CC2550, clock signal is always out of GDO1, which is configured by writing to `IOCFG1.GDO1_CFG` field with the value given in Table 1. Whereas for transceivers, data out (active during RX) and the clock pin can be any one of the GDOx pins, which are configured by writing to the `IOCFG0.GDO0_CFG`, `IOCFG1.GDO1_CFG`, and `IOCFG2.GDO2_CFG` fields with the values given in Table 1.

When using 2-FSK, GFSK, ASK/OOK, or MSK modulation with Manchester encoding disabled, the frequency of the clock on the clock pin is equal to the programmed bit-rate (`MDMCFG4.DRATE_E` and `MDMCFG3.DRATE_M` register fields) in the radio. For example, if the programmed data rate is 10 kbps, the frequency of the clock coming out of the clock pin will be 10 kHz.

Enabling Manchester encoding (available for 2-FSK, GFSK, ASK/OOK modulations) makes the transmit/receive data rate half the programmed data rate. The radio provides the clock at programmed frequency to buffer the data, but it buffers the data in steps of 8-bits, i.e., it buffers 8-bits by providing clock and then the clock pin remains low for next 8 bit period and so on as shown in Figure 1. (For the example in Figure 1 data 0xAA was transmitted repeatedly.)

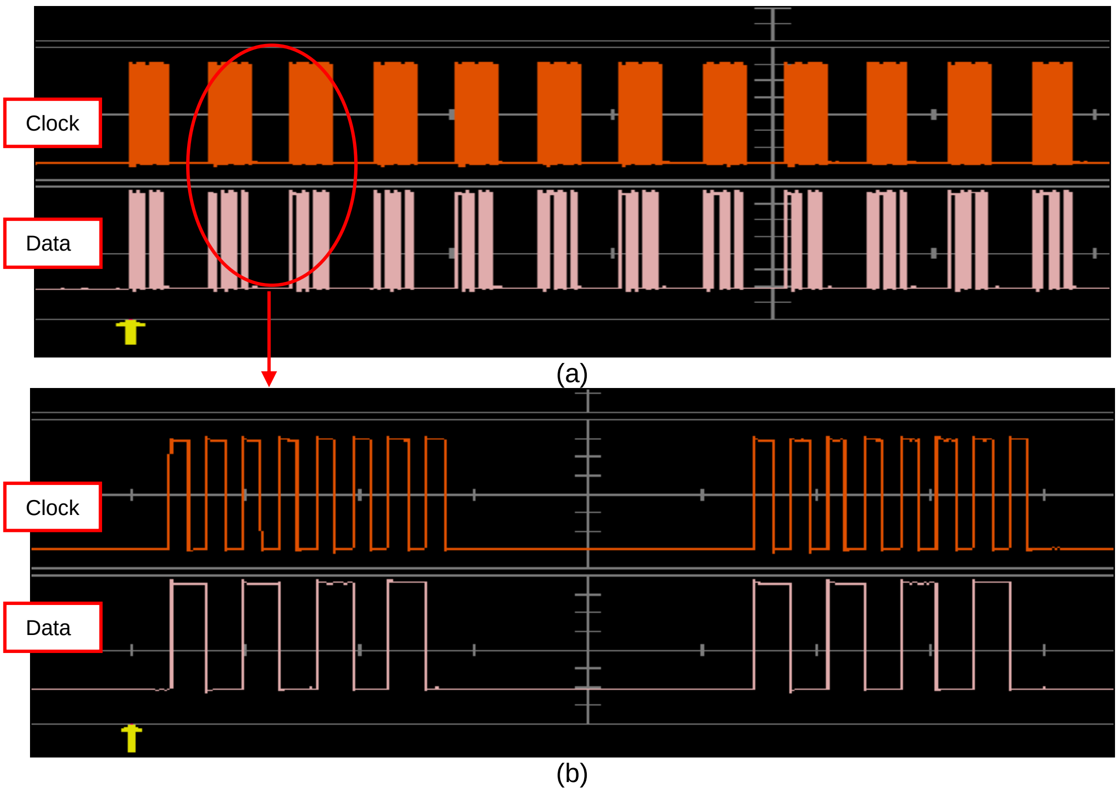

Figure 1: (a) A data packet was transmitted to show the behavior of the clock and data pin when using the Manchester encoding when using 2-FSK, GFSK, ASK/OOK. (b) Zoomed in view of (a)

If the 4-FSK modulation scheme (not available for CC2500) is used, the frequency of the clock from the clock pin is double the programmed data rate.

GDOx_CFG[5:0]	Configuration Values for Serial Synchronous Mode
11 (0x0B)	Serial Clock. Synchronous to the data in synchronous serial mode. In RX mode, data is set up on the falling edge by CC1101 when GDOx_INV=0. In TX mode, data is sampled by CC1101 on the rising edge of the serial clock when GDOx_INV=0.
12 (0x0C)	Serial Synchronous Data Output. Used for serial synchronous mode.

Table 1. Configuration `for IOCFGx.GDOx_CFG`.

NOTE: SmartRF™ Studio [8] should be used to obtain the recommended register settings for the radio. Additionally, for ASK modulation refer to DN022 [7] to determine the optimum register settings.

4 Radio Configurations for Continuous Streaming Applications

Serial synchronous mode provides easy implementation of continuous data streaming applications. Depending on whether the application needs to support a protocol not directly supported by the hardware of the radio, either of the two configurations described in Section 4.1 (Packet Handling Hardware Enabled) and Section 4.2 (Packet Handling Hardware Disabled) may be used.

4.1 Packet Handling Hardware Enabled

If packet handing is enabled when using serial synchronous mode, insertion/detection of the preamble and sync word is done by the radio. All other packet handling features of the radio are available (see [1], [2], [3], [4] ,[5] , and [6]) and FEC can be used, with the exception that the address filtering feature is unavailable when using serial synchronous mode. Also, the MCU interfaced to the radio will have to provide/will receive only the data (that is, the preamble and sync words are handled automatically by the radio hardware). Operation of the radio in transmit and receive mode is explained in Section 4.1.1 and Section 4.1.2.

4.1.1 TX Mode

With the radio in serial synchronous mode and the packet handling hardware enabled (`MDMCFG2.SYNC_MODE≠0/4)`, following the TX strobe, the radio transmits the programmed number of preamble and sync bytes and then the data payload. The transmission starts with the clock pin being low (Figure 3), then the radio buffers a few bytes of data to be transmitted where the radio provides burst of clock, after which the clock again goes to a zero logic level. The duration for which the clock remains low is equal to the time required to transmit preamble and sync. After transmitting the preamble and sync word, the radio provides the clock and continuously transmits the serial data presented by the MCU on the data pin until the programmed number of data bytes have been transmitted. The MCU interfaced to the radio must synchronize with this clock to provide the transmit data correctly. The radio reads the data pin either on the rising or falling edge of the clock depending on the configuration of the GDO0_INVx as defined in Table 1.

4.1.2 RX Mode

With the radio in serial synchronous mode and the packet handling hardware enabled (`MDMCFG2.SYNC_MODE≠0/4)`, following the RX strobe, the radio starts by searching for preamble and sync word and, until these bytes are detected, the clock pin remains at a low logic level. After detecting the preamble and sync word, the radio continuously provides the received data serially on the data pin synchronous to the clock until the programmed number of data bytes have been received. The MCU interfaced to the radio must synchronize with this clock in order to receive the data correctly.

4.1.3 Example on TX and RX Operation when Packet Handling Hardware is Enabled

To illustrate the operation in this mode, the message shown in Figure 2 was transmitted and received.

Preamble	Sync	Data (16 bytes of 0xF0)	Preamble	Sync	Data (16 bytes of 0xF0)	- - - - - - - -

Figure 2. Message Format (Packet Handling HW Enabled)

Figure 3 shows the voltage waveforms on the data and clock pin of the transmitter and receiver in this configuration (in this example `IOCFGx=0x0B` and `GD0x_INV=0` (see Table 1)). As seen in Figure 3, after the TX strobe, the radio transmits the preamble and sync word, and the clock pin remains low with a burst of clock pulses at the start to buffer a few bytes of data. After that, the radio provides the clock and provides the data from the data pin. After the transmission of the programmed number of bytes, the radio repeats the transmit sequence until taken out of TX mode. In receive mode the clock pin is low until the preamble and sync word are found, after which the radio provides the programmed number of data bytes serially on the data pin synchronous to the clock and then repeats this sequence until taken out of RX mode.

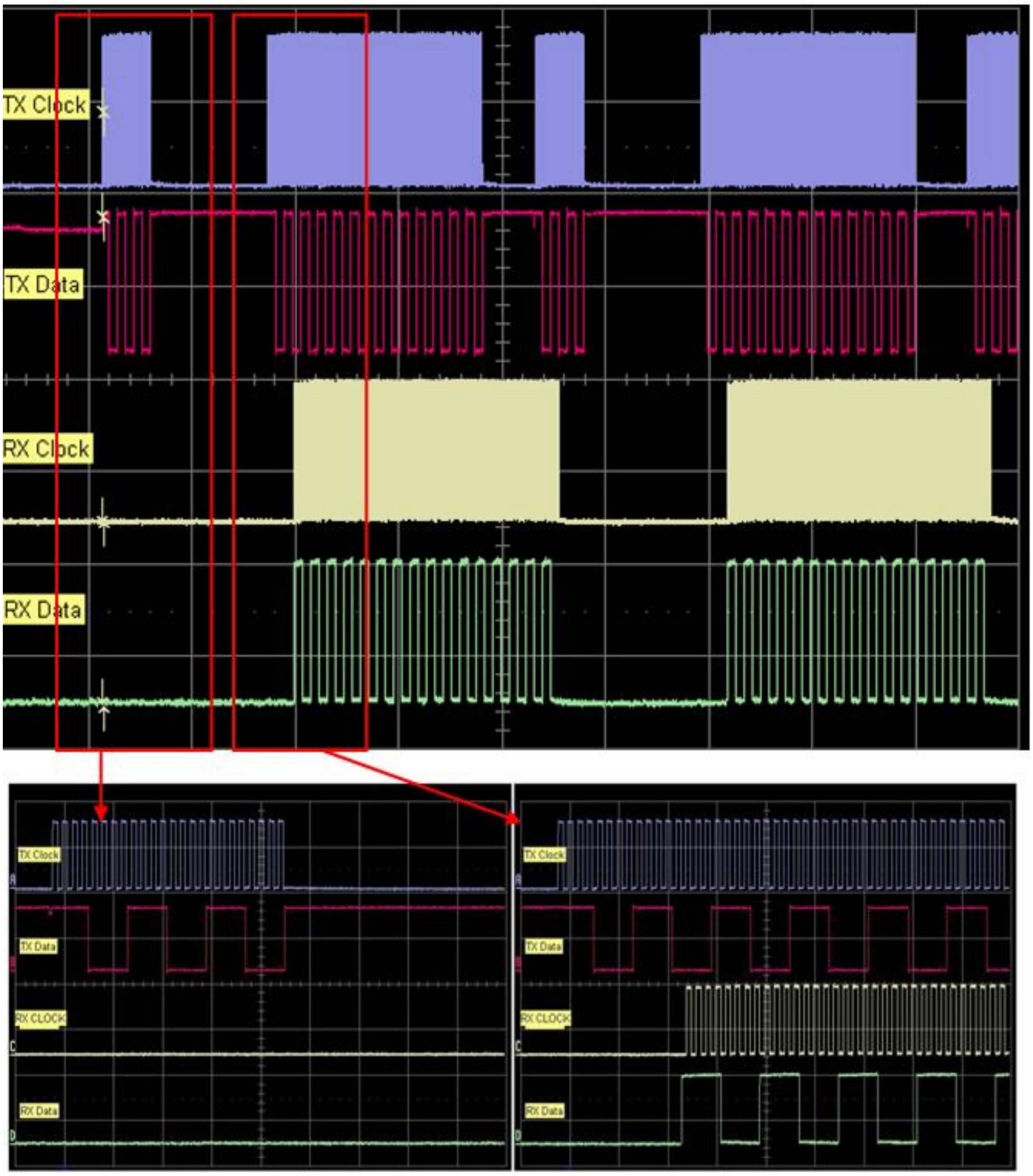

Figure 3. Voltage Transitions on the Clock and Data Pin for the Transmitter and Receiver configured in Serial Synchronous Mode with packet handling hardware enabled.

4.2 Packet Handling Hardware Disabled

If the packet handling hardware is disabled when using serial synchronous mode, all the packet handling features will have to be implemented in the firmware of the MCU interfaced to the radio and should be disabled in the radio register settings. The MCU should then handle the insertion and detection of preamble and sync bytes in the software. This mode allows implementation of protocols not supported by the hardware of CC110x and CC2500 radios.

If packet handing hardware is disabled (`MDMCFG2.SYNC_MODE=0/4`), then, following the TX or RX strobe, the radio will provide the clock on the clock pin and transmit or receive the synchronous serial data on the data pin depending on the configuration of `IOCFGx` pin (see Table 1). To illustrate the operation in this mode message shown in Figure 4 was transmitted and received

0xF0	0xF0	0xF0	---

Figure 4. Message Format (Packet Handling HW Disabled)

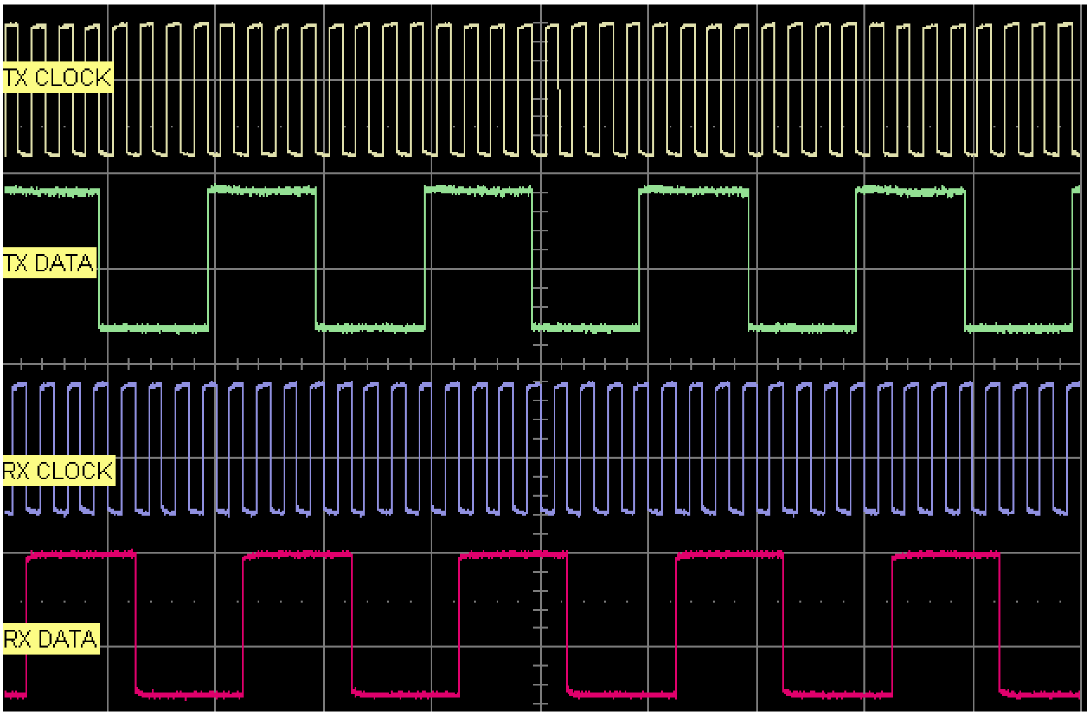

Figure 5. Voltage Transitions on the Clock and Data Pin for the Transmitter and Receiver Configured in Serial Synchronous Mode with Packet Handling Disabled.

As seen in Figure , following the TX or RX strobe, the radio provides the clock on the clock pin and, depending upon the mode selected, captures or provides the data on the data pin.

5 Considerations

This section lists important considerations while implementing continuous streaming applications using serial synchronous mode in TI's CC110x and CC2500 devices.

1. Insert Preamble/Sync data structure periodically within the transmitted data

   For continuous transmission applications where the receiver and transmitter will be in RX and TX mode for extended durations, inserting the preamble/sync word in the transmitted data periodically will allow the receiver's loops to transition from acquisition to tracking mode if the receiver is turned off and then restarted (for example, for periodic calibration).

   NOTE: If the packet handling hardware is disabled and the standard preamble/sync word structure is not used, it is recommended to configure the register `BSCFG.BS_LIMIT = 3` to reduce the PER floor.

2. Avoid long stretches of zeros or ones in the transmitted data

   Use of data whitening when packet handling is enabled or keeping the transmitted data run length limited when packet handling is disabled helps optimize receiver operation. Long stretches of repeated bits should be avoided, as this makes it difficult for the receiver to synchronize with the clock on an incoming data stream and may increase the PER. Data can be run length limited either by implementing data encoding techniques such as 4B/5B, 8B/10B, etc., or through using some form of a data randomization technique such as data whitening, etc., in the firmware of the MCU interfaced to the radio.

3. Perform periodic frequency calibration

   Crystal inaccuracies due to temperature, aging, and loading effects can change the LO frequency on either the transmitter or receiver and cause the IF frequency to change between the transmitter and receiver. This can increase the PER floor if the automatic frequency compensation (AFC) algorithm is not able to track the incoming frequency on the receiver. It is therefore important to consider the limit on the span of frequency tracked by frequency offset compensation algorithm as described in the DN015 [9]. After a packet has been received the accumulated value of the frequency offset estimate status register (`FREQEST.FREQOFF_EST`) must be read and added to the `FSCTRL0.FREQOFF` register, as explained in design note DN015, for the next time radio enters the RX mode to reduce the PER. Additionally, as recommended above; periodically inserting preamble and sync word in the transmitted data allows the receiver, when put in RX mode after this calibration, to successfully transition from acquisition to tracking mode.

6 Conclusion

Serial synchronous mode in TI's CC110x and CC2500 wireless transceivers allows for simple implementation of continuous data streaming operation. Different configurations of the radio in transmit and receive modes and important considerations while using serial synchronous mode have been described. Register settings for the radio with and without packet handling enabled for continuous streaming operation using serial synchronous mode for CC110x and CC2500 devices are given in Appendix A.

7 Appendix A

This section provides examples of the register settings for the radio configurations presented for continuous streaming applications using serial synchronous mode in CC110x and CC2500 transceivers. Register settings (CC110x) to configure the radio in serial synchronous mode with packet handling disabled are given in Table 2. After that, the changes in the register settings required to configure the radio for packet handling enabled configurations are provided (see Table 3). For the given example register settings, the radio is configured with following parameters:

- Serial synchronous mode
- MSK modulation
- Data rate of 500 kBaud
- Frequency of operation 915 MHz
- GDO0 and GDO2 pins are programmed for Data and Clock respectively

Depending on the strobe (i.e. either TX or RX), the radio configured with these settings can either be a transmitter or receiver. Additionally, to change the modulation type, data rate, number of preamble and sync bytes (when using packet handling hardware), etc., it is recommended to use the SmartRF™ Studio [8] to obtain optimum register settings.

Register	Value	Description
IOCFG2	0x0B	GDO2 Output Pin Configuration
IOCFG1	0x2E	GDO1 Output Pin configuration
IOCFG0	0x0C	GDO0 Output Pin configuration
FIFOTHR	0x07	RXFIFO and TXFIFO thresholds
SYNC1	0xD3	Sync word, High byte.
SYNC0	0x91	Sync word, Low byte
PKTLEN	0xFF	Packet length
PKTCTRL1	0x00	Packet automation control
PKTCTRL0	0x12	Packet automation control
ADDR	0x00	Device address
CHANNR	0x00	Channel number
FSCTRL1	0x0E	Freq. synthesizer control
FSCTRL0	0x00	Freq. synthesizer control
FREQ2	0x23	Freq. control word, high byte
FREQ1	0x31	Freq. control word, mid byte
FREQ0	0x3B	Freq. control word, low byte
MDMCFG4	0x0E	Modem configuration
MDMCFG3	0x3B	Modem configuration
MDMCFG2	0x70	Modem configuration
MDMCFG1	0x42	Modem configuration
MDMCFG0	0xF8	Modem configuration
DEVIATN	0x00	Modem deviation setting
MCSM2	0x07	Main radio control state Machine configuration
MCSM1	0x30	Main Radio Control State Machine configuration
MCSM0	0x18	Main Radio Control State Machine configuration
FOCCFG	0x1D	Freq Offset Compensation configuration
BSCFG	0x1F	Bit Synchronization configuration
AGCCTRL2	0xC7	AGC control
AGCCTRL1	0x00	AGC control
AGCCTRL0	0xB0	AGC control
WOREVT1	0x87	Wake On Radio Control
WOREVT0	0x6B	Wake On Radio Control
WORCTRL	0xF8	Wake On Radio Control
FREND1	0xB6	Front end RX configuration.
FREND0	0x10	Front end TX configuration.
FSCAL3	0xEA	Frequency Synthesizer Calibration control

FSCAL2	0x2A	Frequency Synthesizer calibration
FSCAL1	0x00	Frequency Synthesizer calibration
FSCAL0	0x1F	Frequency Synthesizer calibration
RCCTRL1	0x41	RC Oscillator Configuration
RCCTRL0	0x00	RC Oscillator Configuration
FSTEST	0x59	Frequency synthesizer calibration
PTEST	0x7F	Production Test
AGCTEST	0x3F	AGC Test
TEST2	0x88	Various Test settings
TEST1	0x31	Various Test settings
TEST0	0x09	Various Test settings

Table 2. Register Settings to Configure Radio in Serial Synchronous Mode with Packet Handling Hardware Disabled

If packet handling hardware should be enabled, four of the registers listed in Table 2 should be changed as shown in Table 3.

Register	Value	Description
PKTLEN	0x08	ex: Packet length = 8 bytes
PKTCTRL1	0x00	Packet Automatic Control
PKTCTRL0	0x10	Packet Automatic Control
MDMCFG2	0x77	RXFIFO and TXFIFO thresholds

Table 3. Register Settings to Configure Radio in Serial Synchronous Mode with Packet Handling Hardware Enabled

Note: Software example library at [10] gives example “ex_serial_link” describing how to implement the serial synchronous mode for CC110x or CC2500 transceivers.

8 References

[1] CC1100 Low-Cost Low-Power Sub- 1 GHz RF-Transceiver, Data Sheet (SWRS038)

[2] CC1101 Low-Cost Low-Power Sub-1GHz RF-Transceiver, Data Sheet (SWRS061)

[3] CC1100E Low-Cost Low-Power Sub-1GHz RF-Transceiver, Data Sheet (SWRS082)

[4] CC1150 Low-Cost Low-Power Sub-1GHz RF Transmitter, Data sheet (SWRS037)

[5] CC2500 Low-Cost Low-Power 2.4GHz RF-Transceiver, Data sheet (SWRS040)

[6] CC2550 Low-Cost Low-Power 2.4GHz RF-Transmitter, Data sheet (SWRS039)

[7] DN022 CC11xx OOK/ASK Register Settings (SWRS215)

[8] SmartRF® Studio

[9] DN015 Permanent frequency offset compensation (SWRA159)

[10] CC110x and CC2500 Example Libraries (SWRC021)

9 General Information

9.1 Document History

Revision	Date	Description/Changes
SWRA359	2011.01.05	Initial release.
SWRA259A	2011.07.18	Rev. 1