1- and 8-Channel ANT™ RF Network Processors

Check for Samples: CC2570, CC2571

FEATURES

- **RF Section**
  - **Single-Chip 2.4-GHz Radio, Including Embedded ANT Protocol**
  - **Excellent Link Budget, Enabling Long Range Without External Front-Ends**
  - **Excellent Output Power (4 dBm)**
  - **Suitable for Systems Targeting Compliance With Worldwide Radio Frequency Regulations: ETSI EN 300 328 and EN 300 440 Class 2 (Europe), FCC CFR47 Part 15 (US), and ARIB STD-T66 (Japan)**
  - **Accurate Full-Range RSSI Function, Especially Suited for Accurate Proximity Pairing**
- **Layout**
  - **Few External Components**
  - **Reference Designs Available**
  - **40-Pin, 6-mm × 6-mm QFN Package**
- **Low Power**
  - **Powered Down With Low-Power Timer: 1 µA**
  - **Powered Down Without Timer: 0.5 µA**
  - **Wide Supply Voltage Range (2 V–3.6 V)**
- **ANT Protocol Support**
  - **Fully Compatible With the ANT and ANT+™ Protocols and Existing ANT Devices**
  - **Built-In ANT-FS™ Support**
  - **Easy Connection to Host MCU Through Asynchronous or Synchronous Serial Interface**
  - **CC2570 Supports One ANT Channel**
  - **CC2571 Supports Eight ANT Channels**
  - **Support for Both Public, Private and Managed ANT Networks**
  - **Support for High-Resolution Proximity Pairing**

APPLICATIONS

- **Sports and Fitness Equipment**
- **Health and Medical Equipment**
- **Consumer Health Devices**
- **Consumer Electronics**

DESCRIPTION

The CC2570 and CC2571 are ANT RF network processors that implement the easy-to-use, power-efficient ANT protocol. The CC2570 supports one ANT channel, whereas the CC2571 supports eight ANT channels. The CC2570/71 can be connected to a host MCU (such as an MSP430) through a UART or SPI serial interface and accessed through a set of API calls. The majority of the ANT protocol is built into the CC2570/71, including the ANT-FS file system functionality; only the application and profile layers must reside on the host MCU, thus keeping host MCU memory requirements to a minimum.

The ANT protocol has been designed to be very power-efficient, yet is flexible enough to support various network topologies (point-to-point, star, connected star, 1-to-N, and N-to-1) and data transfer modes (broadcast, acknowledged, burst data transfer). Each logical ANT channel can be independently configured for one-way or two-way operation, depending on requirements of the application.

This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

ABSOLUTE MAXIMUM RATINGS(1)

over operating free-air temperature range (unless otherwise noted)

		VALUE	UNIT
Supply voltage	All supply pins must have the same voltage	–0.3 to 3.9	V
Voltage on any digital pin		–0.3 to VDD + 0.3, ≤ 3.9	V
Input RF level		10	dBm
Storage temperature range		–40 to 125	°C
ESD	All pads, according to human-body model, JEDEC STD 22, method A114	2	kV
	According to charged-device model, JEDEC STD 22, method C101	500	V

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS

over operating free-air temperature range (unless otherwise noted)

		MIN	NOM	MAX	UNIT
T_A	Operating ambient temperature range	–40		85	°C
	Operating supply voltage	2		3.6	V

ELECTRICAL CHARACTERISTICS

Measured on CC2570/71 reference design with T_A = 25°C and VDD = 3 V, unless otherwise noted.

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
I	Current consumption	RX current consumption		23.7		mA
		TX current consumption, –6-dBm output power		25.9		mA
		TX current consumption, 0-dBm output power		28.8		mA
		TX current consumption, 4-dBm output power		34.3		mA
		Power-down current, 32-kHz oscillator active		1		µA
		Power-down current, 32-kHz oscillator disabled		0.5		µA

GENERAL CHARACTERISTICS

Measured on CC2570/71 reference design with T_A = 25°C and VDD = 3 V, unless otherwise noted.

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
RADIO PART					
RF frequency range	Programmable in 1-MHz steps	2400		2495	MHz
Data rate and modulation format		1 Mbps, GFSK, 160-kHz deviation			

RF RECEIVE SECTION

1 Mbps, GFSK, 160-kHz deviation. Measured on Texas Instruments CC2570/71 reference design with T_A = 25°C, VDD = 3 V, and f_C = 2440 MHz, unless otherwise noted.

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Receiver sensitivity	0.1% BER		–86		dBm
Saturation			10		dBm
Co-channel rejection			–9		dB
Adjacent-channel rejection	±2 MHz		23		dB
Alternate-channel rejection	±4 MHz		39		dB
Frequency error tolerance(1)	Including both initial tolerance and drift	–150		150	kHz
Symbol rate error tolerance(2)		–50		50	ppm
Spurious emission. Only largest spurious emission stated within each band.	Conducted measurement with a 50-Ω single-ended load. Complies with EN 300 328, EN 300 440 class 2, FCC CFR47, Part 15 and ARIB STD-T-66.		–70		dBm

(1) Difference between center frequency of the received RF signal and local oscillator frequency
(2) Difference between incoming symbol rate and the internally generated symbol rate

RF TRANSMIT SECTION

Measured on CC2570/71 reference design with T_A = 25°C, VDD = 3 V, and f_C = 2440 MHz, unless otherwise noted.

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Output power, maximum setting	Delivered to a single-ended 50-Ω load through a balun using maximum recommended output power setting.		4		dBm
Output power, minimum setting	Delivered to a single-ended 50-Ω load through a balun using minimum recommended output power setting.		–21		dBm
Programmable output power range	Delivered to a single-ended 50-Ω load through a balun. Conducted measurement with a 50-Ω single-ended load.		25		dB
Spurious emissions, conducted	Complies with EN 300 328, EN 300 440 class 2, FCC CFR47, Part 15 and ARIB STD-T-66. (1)		–45		dBm

(1) Designs with antenna connectors that require conducted ETSI compliance at 64 MHz should insert an LC resonator in front of the antenna connector. Use a 1.6-nH inductor in parallel with a 1.8-pF capacitor. Connect both from the signal trace to a good RF ground.

ANT OVERVIEW

ANT is a proven ultralow-power wireless protocol operating in the license-free 2.4-GHz ISM band. It has been specifically designed to target battery operated devices that require years of battery life (often on a single coin cell) without sacrificing the robust features expected of a mature wireless protocol. These features include sophisticated co-existence mechanisms, practical topologies that go beyond simple peer-to-peer or star, easy proximity-based pairing methods, and the seamless transfer of bulk data from one coin-cell-operated device to another. Combined, these features enable products that are easy to use and quickly adopted by the end user. ANT is also the fundamental building block of ANT+, which further enhances the user experience by allowing manufacturers to create interoperable wireless devices based on ANT+ device profiles.

ANT provides for an easy development and integration experience. By fully integrating the lower four layers of the OSI stack onto a single chip, designers can focus on their applications and not the wireless protocol. This enables quick development cycles, fast time to market, and lower BOM costs.

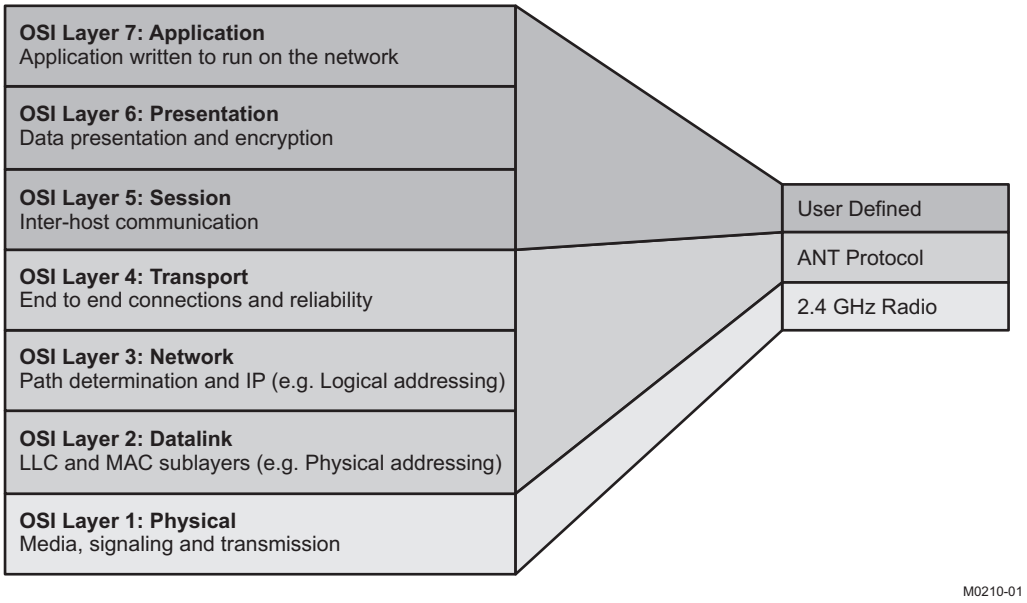


Figure 1. ANT OSI Layers

The following sections offer a broad overview of the many features of ANT. For a detailed description of ANT and ANT+, please visit www.thisisant.com or www.ti.com/ant.

ANT NODES

An ANT node is any device that is capable of ANT wireless communications. It typically coconsists of an ANT chip, such as the TI CC2571, and an application host MCU, such as an MSP-430. The interface between the ANT chip and the host MCU is serial UART or SPI, depending on the requirements of the application. There is also a bit-synchronous serial protocol available, which may be fully implemented in software for very resource-constrained systems. The overhead required to interface to ANT is minimal, typically requiring less than 1K of flash space to implement a simple API. A block diagram of an ANT node is depicted in Figure 2.

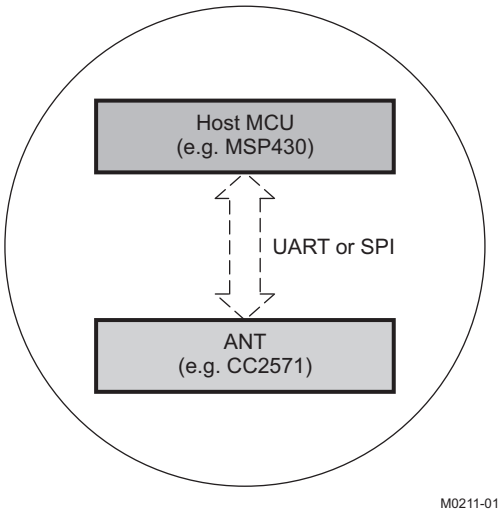


Figure 2. ANT Node

ANT nodes may host multiple ANT channel endpoints, up to eight for the CC2571. Examples of ANT nodes include wireless watches, heart-rate straps, smart phones, glucose meters, blood-pressure monitors, and other sports and medical sensors.

ANT CHANNELS

ANT channels are the fundamental building blocks of ANT networks. An ANT channel is a logical connection between two ANT nodes. ANT channels are fully independent, meaning that multiple channels can form a network without the need for a central coordinator or a network master, with each channel managing itself relative to other channels in the RF space. A single ANT node can support multiple channel endpoints, each completely independent of the other, allowing master and slave combinations on a single device. ANT channels are also *ad hoc*, allowing connections to be created and destroyed on an as-needed basis.

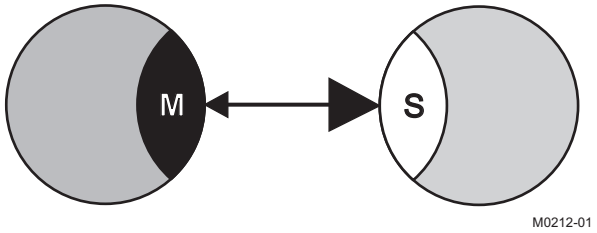


Figure 3. ANT Channel

An ANT channel is formed between a master and a slave endpoint (Figure 3). The master endpoint is the communication initiator and controls the channel. The master is often referred to as the *primary transmitter*, as it is always transmitting a data packet at a specific channel period to keep the channel synchronized. The channel period can vary from 0.5 Hz to 200 Hz and can be changed dynamically by the application host MCU. Each data packet carries exactly 8 bytes of application data payload. The slave endpoint is the communication acceptor, or *primary receiver*. It searches for master transmissions and then synchronizes to the master at the channel period of the master, or a multiple thereof. The slave only transmits data to the master if it is instructed to do so by the application MCU. Figure 4 illustrates the behavior of a bidirectional synchronized ANT channel.

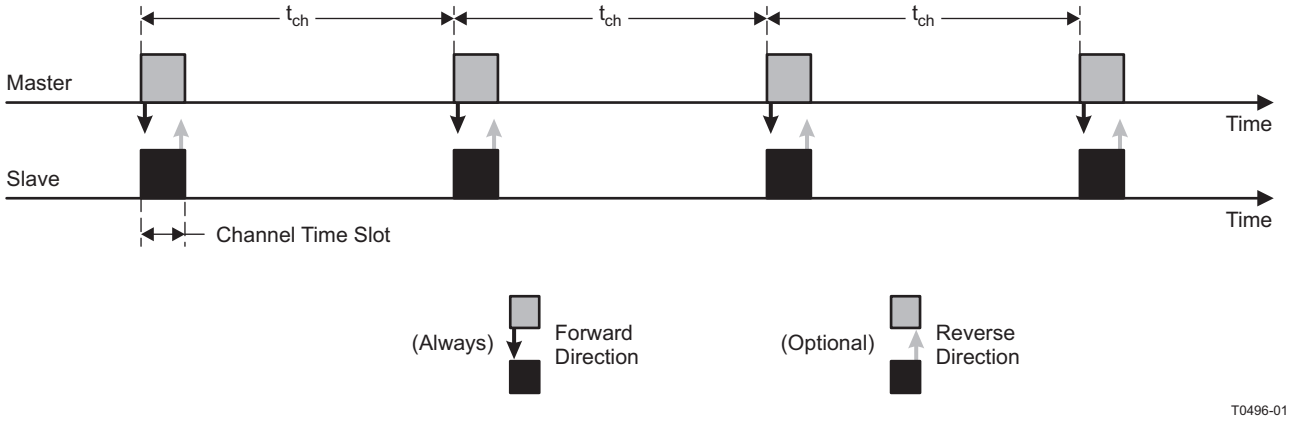


Figure 4. ANT Channel Timing

The search algorithm used by the slave to find the master has been specifically designed to optimize battery life. The slave duty-cycles its radio at a specific rate to trade off acquisition latency for power consumption, allowing for prolonged searches that can still operate on a coin cell battery.

Each packet transmitted by ANT is characterized by the following parameters:

- 8-byte data payload
- Frequency (2400 MHz to 2485 MHz)
- 2-byte network key
- 4-byte channel ID

Each of these parameters can be changed by the application MCU between message periods.

ANT Channel Types

There are different types of ANT channels, including bidirectional channels, unidirectional channels, shared channels, and scanning channels. The type of channel chosen for a particular application is dependent on the application requirements for use case, power consumption, topology, and battery life expectancy.

Bidirectional channels support two-way communication between the master and the slave. The master transmits a message on every channel period, whereas the slave only transmits a message to the master if required to do so by the application. For every message that the slave receives from the master, the slave can transmit a message back. Hence, the effective data through put is the same in both directions. Bidirectional channels are also necessary to take advantage of ANT's unique channel co-existence mechanisms.

Unidirectional, or transmit-only, channels allow for communication from the master to the slave only. These types of channels support the broadcast message type only and do not offer the co-existence mechanisms built into bidirectional channels. Although unidirectional channels provide the lowest power solution, they are generally not recommended for systems that require co-existence of several ANT channels in a common RF space.

Shared channels are a special type of bidirectional channel that allow a single master endpoint to address up to 64K slaves. Each slave synchronizes to the channel period of the master. The master can address a different slave on each channel timeslot by specifying the address of the slave in the first 2 bytes of the data payload. The master may also send a broadcast message to all of the slaves simultaneously by specifying an address of 0. Shared channels are an excellent solution for networks that require many of nodes and are sensitive to power requirements but not to latency requirements.

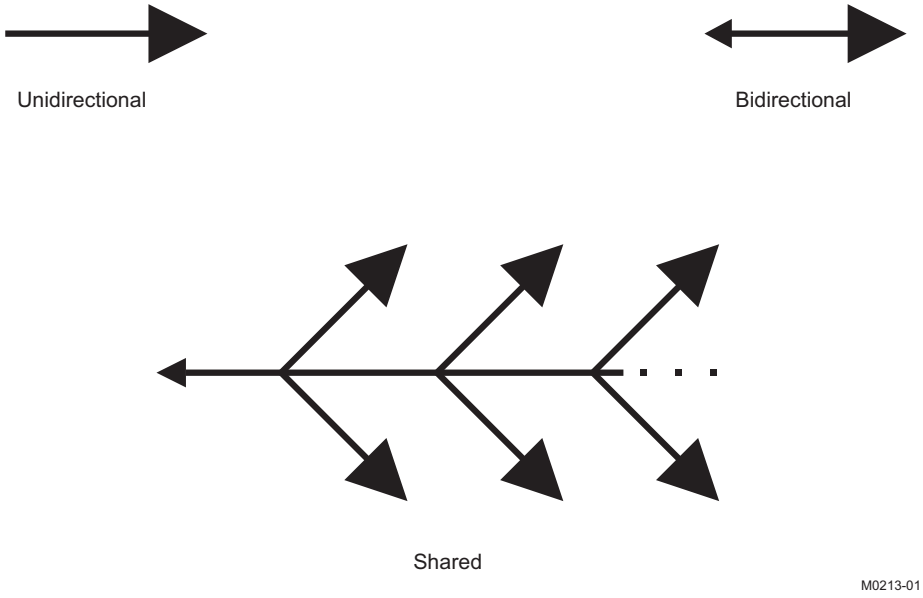


Figure 5. Synchronous ANT Channel Types

Unlike the channel types listed previously, scanning channels are different in that they are asynchronous. That is, the slave does not synchronize to a master channel. Instead, any message received from a master is passed onto the application MCU, along with the channel ID of the device received from. This allows a slave to receive from multiple masters without actually forming a connection to any particular master. Scanning channels come in two flavors – continuous-scanning and background-scanning. The difference between the two amounts to a tradeoff between power consumption and latency. A continuous-scanning channel keeps its receiver on all the time. This means that any transmissions are received immediately, eliminating latency at the cost of power consumption to keep the radio on all the time. Continuous-scanning channels are an ideal solution for systems that must be very low power on one side of the link but not the other, for example, a remote control. A background scanning channel does not keep its receiver on all of the time. Instead, it continually searches for transmissions, without ever actually synchronizing to a channel. The power consumption is drastically reduced at the expense of data latency. Background scanning channels are ideal for battery-operated systems where the hub must be able to communicate with multiple nodes with reasonable latency, for example, a proximity awareness application.

Message Types

Although the data payload is fixed at 8 bytes per packet, ANT channels do support different types of messages including broadcast messages, acknowledged messages and burst messages. The choice of message type to use depends on the needs of a particular application.

Broadcast messages are messages that do not expect a response. This means that the sender of a broadcast message (whether it be a slave or master) has no way of knowing if the message it sent was successfully received by its intended recipient. Broadcast messages offer the lowest power consumption of the three different message types. As these types of messages do not ellicit a response from their target, these messages are ideally suited for applications where one-to-many type architectures are required. For example, a sport sensor may broadcast data to a watch and smartphone simultaneously. Broadcast type messages are also useful for sensor data that changes slowly relative to the channel period, and it is more important to have the latest data rather than every single packet of data. For example, a temperature sensor is an ideal candidate for a broadcast data type. Broadcast messages are the default message type used by a master channel and are the only data type supported by transmit-only channels.

Acknowledged messages are messages that expect and ellicit a response from the receiving device. The response to an acknowledged message is automatically handled by the ANT protocol (transparent to the application MCU). On sending an acknowledged message type, the application MCU will be notified whether or not the message successfully reached its target. Any retries must be handled by the application MCU, as ANT does not re-transmit any messages that were not acknowledged. Acknowledged type messages are best suited for control applications where the transmitter must know whether or not a message got through. For example, a remote control or actuator is an ideal application for an acknowledged message type.

Burst messages are designed to allow the transfer of bulk data as fast as possible without compromising the ability of a device to run using a coin-cell battery. A burst always starts on a channel-period timeslot and send packets as fast as possible, potentially extending the channel period. The maximum data throughput for a burst is 20 kbps. Each burst packet is re-tried by ANT up to five times if necessary. The success or failure of a burst operation is communicated to the application MCU. The burst message type is most appropriate for sending large amounts of episodic data. For example, a watch may upload data to a computer after a workout using burst-type messages.

Pairing

In order to ensure that a slave is talking to the correct master, the slave must know the channel ID of the master. If the channel ID is not known, the slave must attain the channel ID through a process called pairing. Pairing is an issue that affects all wireless technologies. There are several methods available for pairing, each with its merits and drawbacks. Ultimately, the best pairing solution is one that is most seamless to the user. ANT has many pairing methods built in, ranging from simple to sophisticated, and allowing the designer to customize the pairing experience for a particular application.

The simplest pairing method is for the slave to wildcard the channel ID (by explicitly setting it to 0). When the channel ID is wildcarded, the slave connects to the first master it finds (provided it matches non-wildcarded portions of the ID and the frequency and network number). This method is easy and relatively seamless to the user. The user generally only must perform a UI operation on the slave device and then wait to connect to the master. However, if the user is in a crowded environment, where several devices to pair with are available, this method breaks down. For example, if a user is trying to pair to a heart-rate strap at a gym, the user may easily pair to a device that is not the user's own. Hence, a wildcard pairing method is generally suitable if pairing is expected to happen only occasionally and in isolation.

To reduce the chances of pairing to an incorrect device in a crowded environment, ANT allows the user to put the ANT channel into a pairing mode. This is done by setting a particular bit of the channel ID of the master device and forcing the slave to only pair to devices that have this bit set. This method reduces the chances of pairing to the wrong device; however, it forces the user to perform a UI operation on the slave device and on the master device. This may not be practical both from a user point of view and from a production point of view, as adding a button or switch may not be feasible for a particular device.

Another, more-powerful, method of pairing is to use the relative proximity of a slave device to a master device to determine if pairing should occur. This is known as *proximity pairing*. ANT has a very simple interface that allows the user to specify a proximity threshold, effectively blocking other devices that are outside of this threshold. Once pairing has occurred, the effective transmission range is returned to normal, allowing the device to function normally.

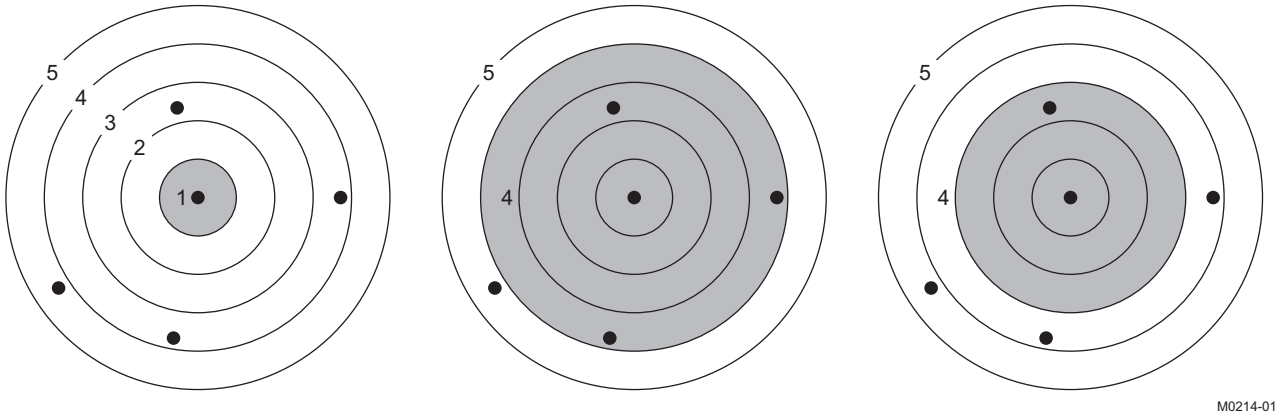


Figure 6. Varying Proximity Thresholds

There are 10 proximity bins available to the application. Each bin represents a relative distance from the device. The actual distance varies from implementation to implementation and is affected by many factors, including housing, component tolerances, and RF environment. In Figure 6 (third on the left diagram), if the user put the slave device (center) into proximity pairing mode with a bin of 3, the user would only be able to pair to one device. All other devices would be outside of the pairing range.

Proximity pairing is a very powerful feature of ANT and is applicable in almost all applications, from simple sensor pairings that occur occasionally to complex pairing environments such as gyms where pairing may be done continuously with pieces of equipment in a crowded environment. It is seamless to the user and requires no special hardware or UI to implement.

ANT NETWORKS

With up to eight independent ANT channels available on a single device, it is possible to connect ANT nodes to many different types of network topologies. Sophisticated network topologies are just collections of ANT channels and simple star networks. Figure 7 illustrates some of the possible network arrangements possible with ANT.

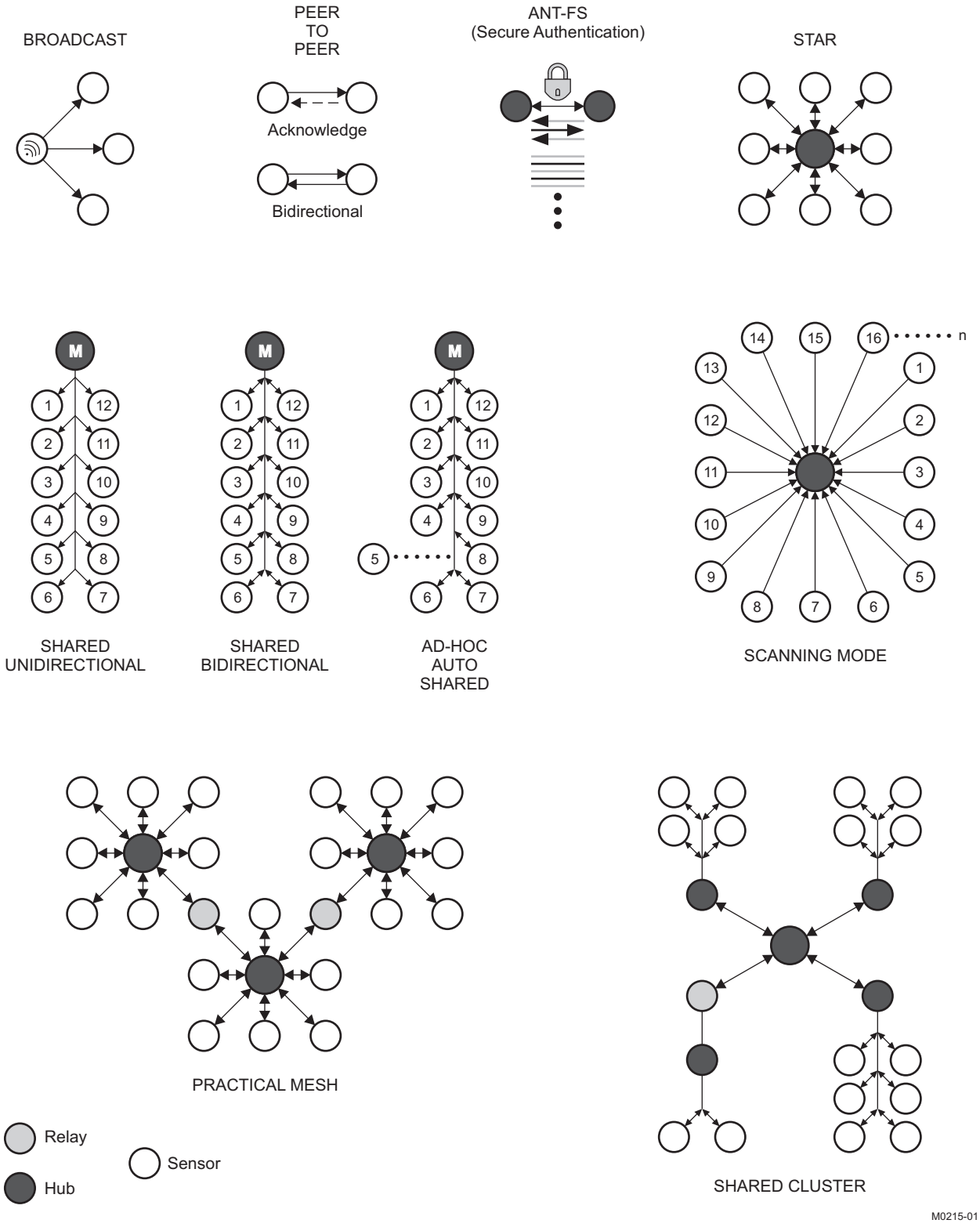


Figure 7. ANT Network Topologies

ANT-FS

A powerful feature of ANT is the ability to do efficient, automated, and secure downloads of bulk data from one device to another. This is enabled using an extension to the ANT protocol called ANT-FS (File Share). While the ANT-FS protocol can be implemented at the application level, the TI CC257X family has been designed with ANT-FS integrated on-chip, allowing designers access to this powerful utility with minimal development effort, enabling quick time to market.

The ability to perform file transfers enables all sorts of interesting use cases. In the personal monitoring environment, data from sensors may be stored locally on a battery-operated hub and then uploaded to a PC, cellphone, or other collection device for display, further processing, or transferring of data across a network.

The ANT-FS protocol defines communication between two devices, a client and a host. The host is typically the higher-power device and may also be a gateway or a hub device. It is implemented as an ANT slave channel, and its job is to download and upload files from and to client devices. The client is typically the lower-power device and is the mobile storage device that interfaces to sensors. It is implemented as an ANT master channel. The distinction between the host and client may not always be easy. For example, an ANT-FS session may occur between two identical peer devices. The important point to remember is that both the client and the host may be implemented with minimal resources that can be powered by a coin-cell battery.

ANT-FS defines three layers of communication: the link, authentication, and transport layers. Both the client and the host traverse through these layers before any data exchange can occur. The purpose of these layers is to provide a seamless, yet secure interface for transferring files between devices.

In the link layer, the client device advertises its presence by sending out a periodic beacon. The beacon contain information pertaining to the client's state and capabilities, as well as manufacturer and device type. This information is used by the host device in the link layer to determine whether or not it should connect to a particular device. Once a host detects a client device in the link layer that it wishes to communicate with, it sends a link command to that device.

On receiving a link command from the host, the client moves to the authentication state. In the authentication state, the client authenticate the host in order to establish a secure link. The authentication method is flexible and may depend on a particular application. Three methods are currently defined by the ANT-FS specification: pass-through, passkey, and pairing.

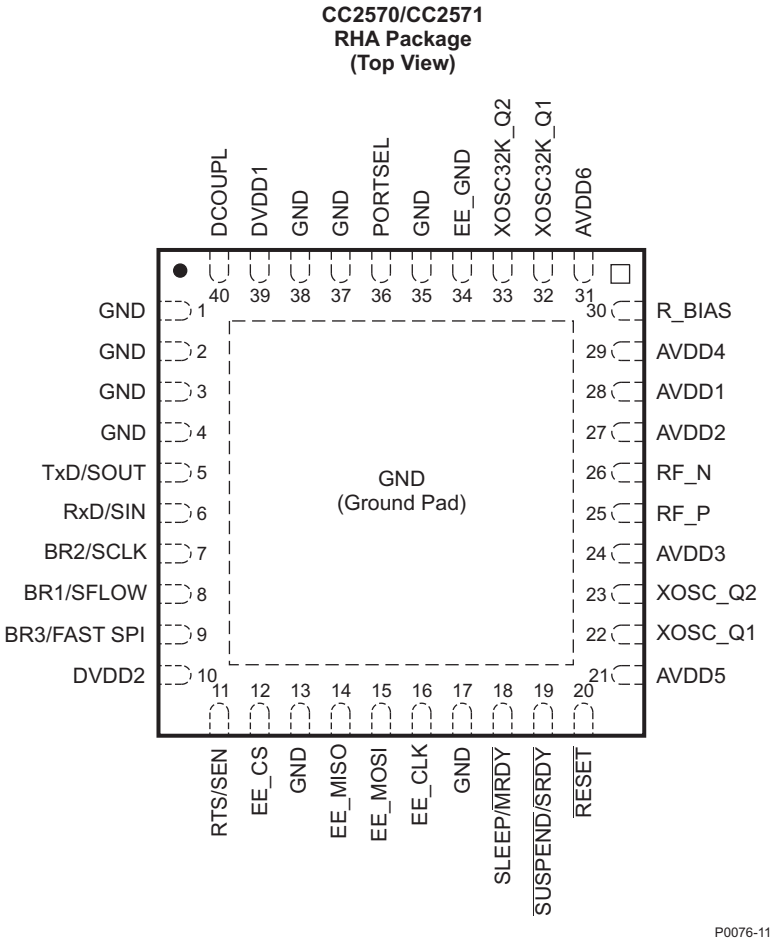
Pass-through authentication is really no authentication at all. The host simply sends a request to pass through authentication. If the client supports this method, it accepts this request; otherwise, it is rejected. Passkey authentication requires that the host send a unique passkey to the client. If the passkey matches that of the client, authentication passes; otherwise, it is rejected. This method requires that the passkey be attained by the host at some earlier point. The pairing method of authentication requires that the host send a command to the client to request pairing. The client either passes this request to the user of the device (requiring a UI) or uses some other method to determine if the pairing should be accepted (for example, proximity of the host device). If pairing is accepted, the client may then send a passkey to the client to be used for future authentication attempts.

Once authentication is accepted by the client, the client and the host move to the transport layer. In the transport layer, the host can request a directory of files (analogous to a DOS-style directory). It can then request to download, upload, and erase files stored on the client.

The ability to do secure file downloads is very powerful. While other technologies can support this use case, only ANT is able to provide a system that is low-power enough to support the seamless use cases provided by ANT-FS. In addition to performing generic file transfer, ANT-FS is also the base technology used by file-based device profiles defined within the ANT+ ecosystem.

DEVICE INFORMATION

PIN ASSIGNMENT



NOTE: The exposed ground pad must be connected to a solid ground plane, as this is the ground connection for the device.

PIN FUNCTIONS

PIN		DIRECTION	DESCRIPTION
NAME	NO.		
AVDD1	28		
AVDD2	27		
AVDD3	24		
AVDD4	29		
AVDD5	21		
AVDD6	31		
BR1/SFLOW	8	IN/IN	Serial communication pin. Synchronous: SFLOW (bit- or byte-mode selection – see Table 1) Asynchronous: Baud-rate configuration pin (see Table 2)
BR2/SCLK	7	IN/OUT	Serial communication pin Synchronous: SCLK (SPI master clock out) Asynchronous: Baud-rate configuration pin (see Table 2)
BR3/FAST SPI	9	IN	Serial baud rate selection Synchronous: FAST SPI (selects fast SPI clock) Asynchronous: BR3 (baud-rate configuration pin for asynchronous serial communication (see Table 2).

PIN FUNCTIONS (continued)

PIN		DIRECTION	DESCRIPTION
NAME	NO.		
DCOUPL	40		
DVDD1	39		
DVDD2	10		
EE_CLK	16	OUT	EEPROM SPI master clock
EE_CS	12	OUT	EEPROM SPI chip select
EE_GND	34	OUT	EEPROM ground
EE_MISO	14	IN	EEPROM SPI master in
EE_MOSI	15	OUT	EEPROM SPI master out
GND	1, 2, 3, 4, 13, 17, 35, 37, 38	N/A	Ground
PORTSEL	36	IN/IN	PORTSEL: Configuration pin to select synchronous or asynchronous serial communication
R_BIAS	30		
$\overline{\text{RESET}}$	20	I	Reset signal
RF_N	26		
RF_P	25		
RTS/SEN	11	OUT/OUT	Digital output for serial communication Synchronous: SEN (Serial-enable flow control) Asynchronous: RTS (UART flow control)
RxD/SIN	6	OUT/OUT	Digital input for serial communication Synchronous: SIN (SPI master in) Asynchronous: RxD (UART receive)
SLEEP/$\overline{\text{MRDY}}$	18	IN/IN	Serial communication pin Synchronous: $\overline{\text{MRDY}}$ (message-ready flow control signal) Asynchronous: SLEEP (sleep assert)
$\overline{\text{SUSPEND}}$/$\overline{\text{SRDY}}$	19	IN/IN	Serial communication pin Synchronous: $\overline{\text{SRDY}}$ (bit- or byte-ready flow control) Asynchronous: $\overline{\text{SUSPEND}}$ (suspend signal)
TxD/SOUT	5	OUT/OUT	Digital output for serial communication Synchronous: SOUT (SPI master out) Asynchronous: TxD (UART transmit)
XOSC32K_Q1	32		
XOSC32K_Q2	33		
XOSC_Q1	22		
XOSC_Q2	23		

SERIAL INTERFACE

The CC257x supports various serial interfaces to accommodate almost any application requirements. These include asynchronous, byte synchronous, and bit synchronous (bit bashing). The serial interface is selected by using external configuration pins, as described in Table 1. Because these pins are sampled by the CC257x on startup to determine serial interface, it is important that these pins retain their state while powered up. Changing the state of these pins after start-up may result in undefined behavior.

Table 1. Serial Interface Selection

Serial Mode	PORTSEL (Pin 36)	SFLOW (Pin 8)
Asynchronous	0	X
Byte synchronous	1	0
Bit synchronous	1	1

The asynchronous serial interface may be used with a UART peripheral on an external MCU. The baud rates are selectable via a hardware pin configuration.

Byte-synchronous serial may be used with an SPI peripheral (three-wire), though flow control must be implemented in software using GPIOs. The bit-synchronous serial interface is designed to be implemented in software on the external MCU, allowing the interface to be implemented using GPIOs only.

ASYNCHRONOUS SERIAL INTERFACE

The asynchronous serial interface is selected by setting the PORTSEL pin to VSS (GND). The asynchronous serial interface is functionally equivalent to a standard UART with 8 data bits, no parity, and 1 stop bit. Unlike a standard UART, the asynchronous serial interface supports unidirectional flow control only. Using the asynchronous serial interface also requires that the host MCU tightly control the SLEEP state of the CC257x by using the SLEEP and SUSPEND control lines. The following block diagram shows the interconnections required for the asynchronous serial interface.

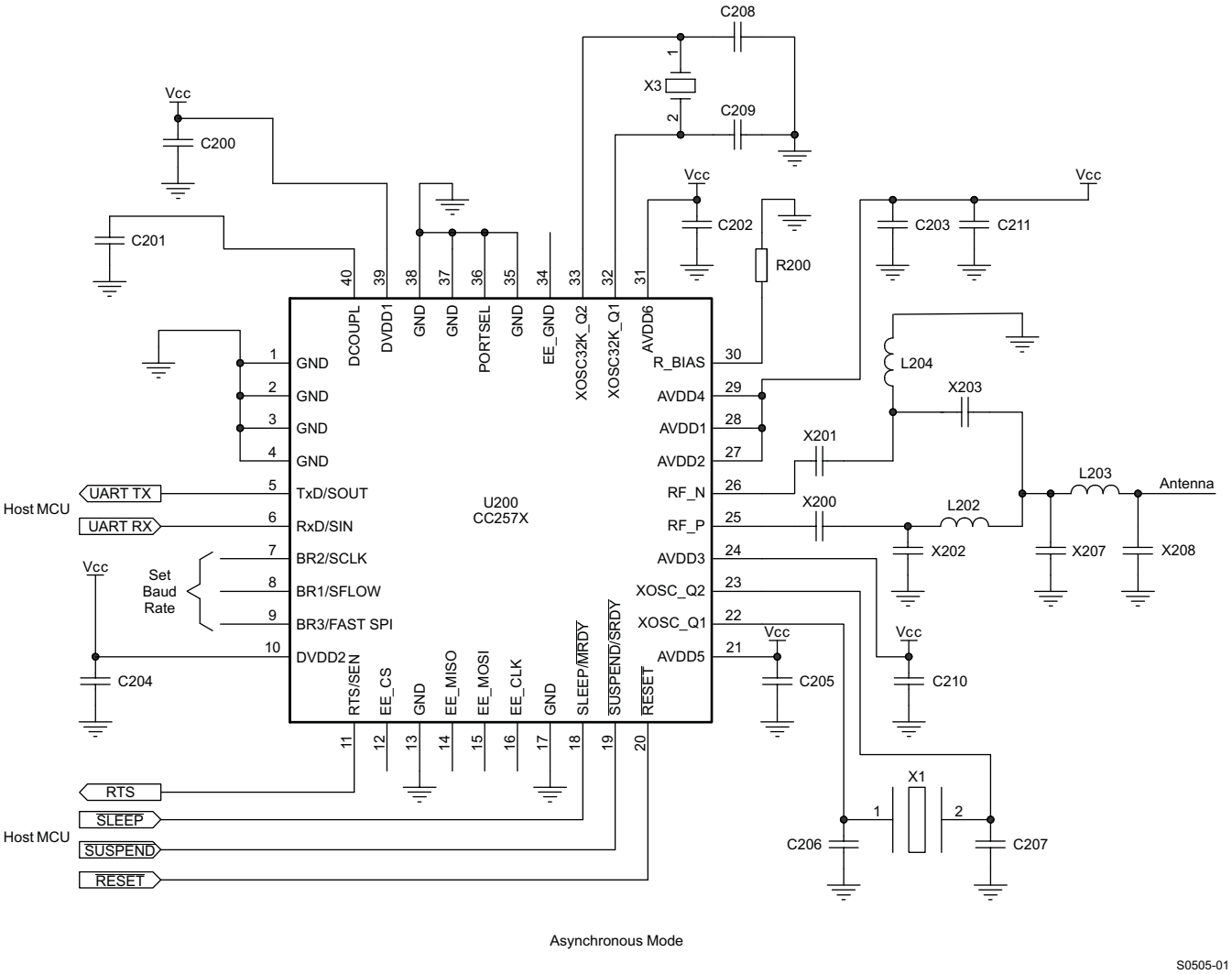


Figure 8. Asynchronous Serial Hardware Setup

The baud rate is selected using a hardware pin configuration using pins BR1, BR2, and BR3. Table 2 lists the available baud rates and the corresponding pin configurations.

Table 2. Baud Rate Selection

Baud Rate	BR1	BR2	BR3
57,600	1	1	1
50,000	1	1	0
38,400	1	0	0
19,200	0	1	0
9,600	1	0	1
4,800	0	0	0

The asynchronous serial interface supports unidirectional hardware flow control. As indicated in Figure 8, the RTS line from the CC257x should be connected to the CTS line of the host MCU. The host MCU must halt the sending of any messages while the RTS signal is asserted. Note that the RTS signal is briefly asserted following every message received by ANT on the serial port. The length of this pulse is approximately 50 µs. Any bytes sent while RTS is high are ignored by ANT. For this reason, it is recommended that two padding bytes or a reasonable delay be inserted between serial messages. This is especially important when sending burst serial messages to ANT.

USE OF SLEEP AND SUSPEND

To ensure lowest power operation, the host MCU must control the power state of the CC257x explicitly using the SLEEP and $\overline{\text{SUSPEND}}$ signal lines. Review the ANT document *Interfacing with ANT General Purpose Chipsets and Modules* and the Application note *Power States* for complete details on how these signals should be used.

The SLEEP signal should be used to put the CC257x into its lowest state when serial communication is not required. This is accomplished by asserting the SLEEP signal between any serial messages being sent from the host MCU to ANT. This is illustrated by the timing diagram in Figure 9.

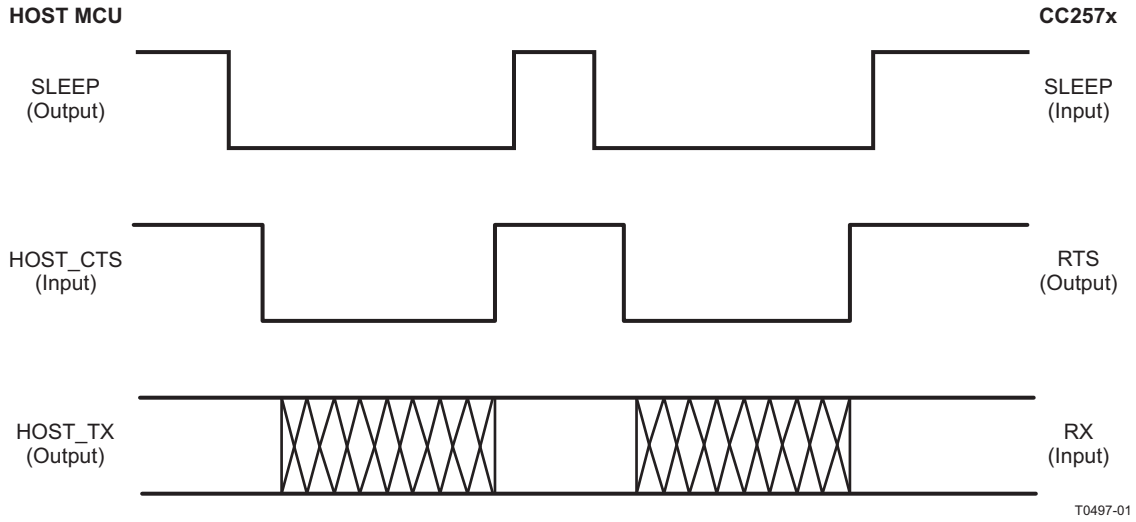


Figure 9. Use of Sleep

Regardless of the state of the SLEEP signal, ANT sends any messages to the host MCU immediately.

The $\overline{\text{SUSPEND}}$ signal should only be used in USB applications where the USB specification requires that the MCU be put into a known low power state very quickly. Asserting $\overline{\text{SUSPEND}}$ terminates all RF activity, regardless of the state of ANT. Entering and exiting the suspend state requires the use of the SLEEP signal as depicted in Figure 10.

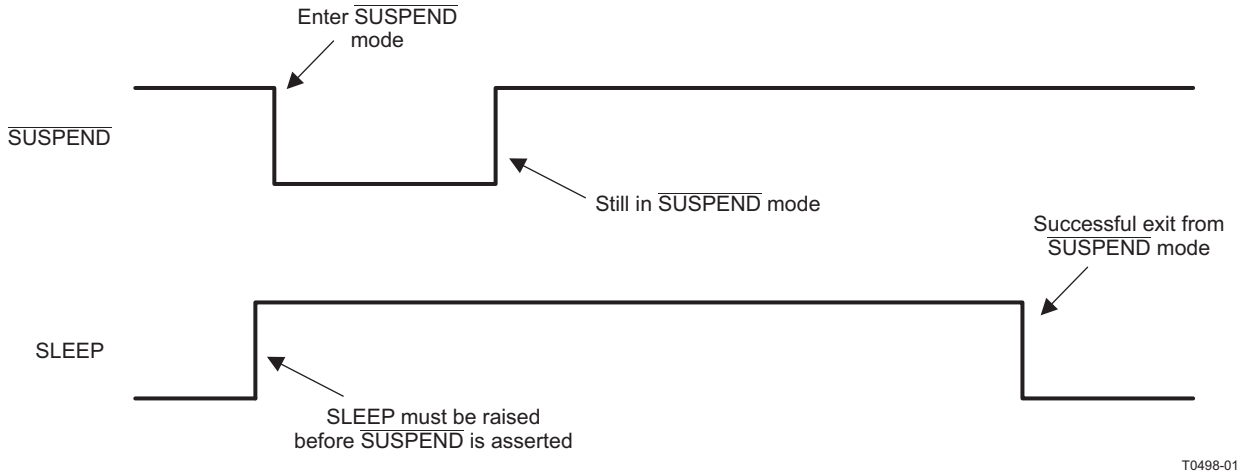


Figure 10. Coming Out of Suspend

Note that after exiting from the suspend state, ANT is in the power-up reset state. Any channel configurations must be re-sent.

SYNCHRONOUS SERIAL INTERFACE

The synchronous serial interface may be configured as bit- or byte-synchronous, as defined in Table 1. To ensure synchronization with the host MCU, the host should reset the CC257x by using the RESET pin. The CC257x is ready to communicate within 100 ms after the reset is applied. If it is desired to ensure software compatibility with other ANT chips that do not support the RESET pin, a synchronous reset may be issued instead. A synchronous reset is depicted in Figure 11.

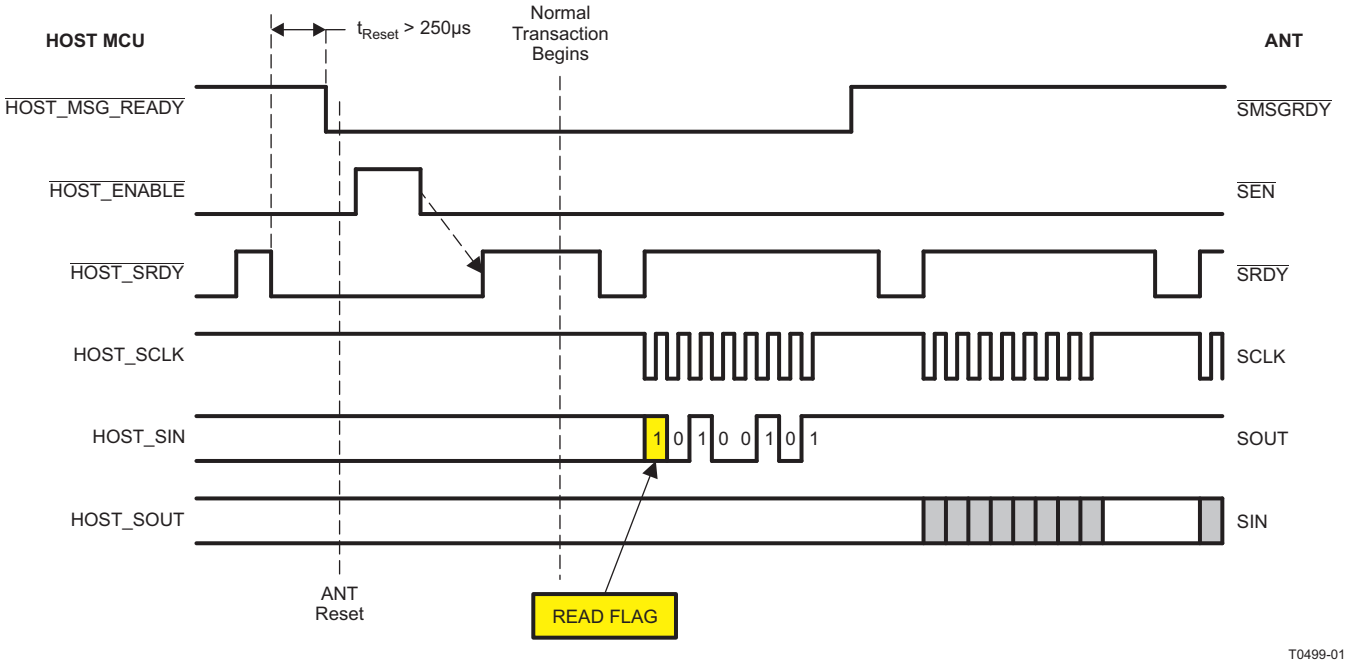
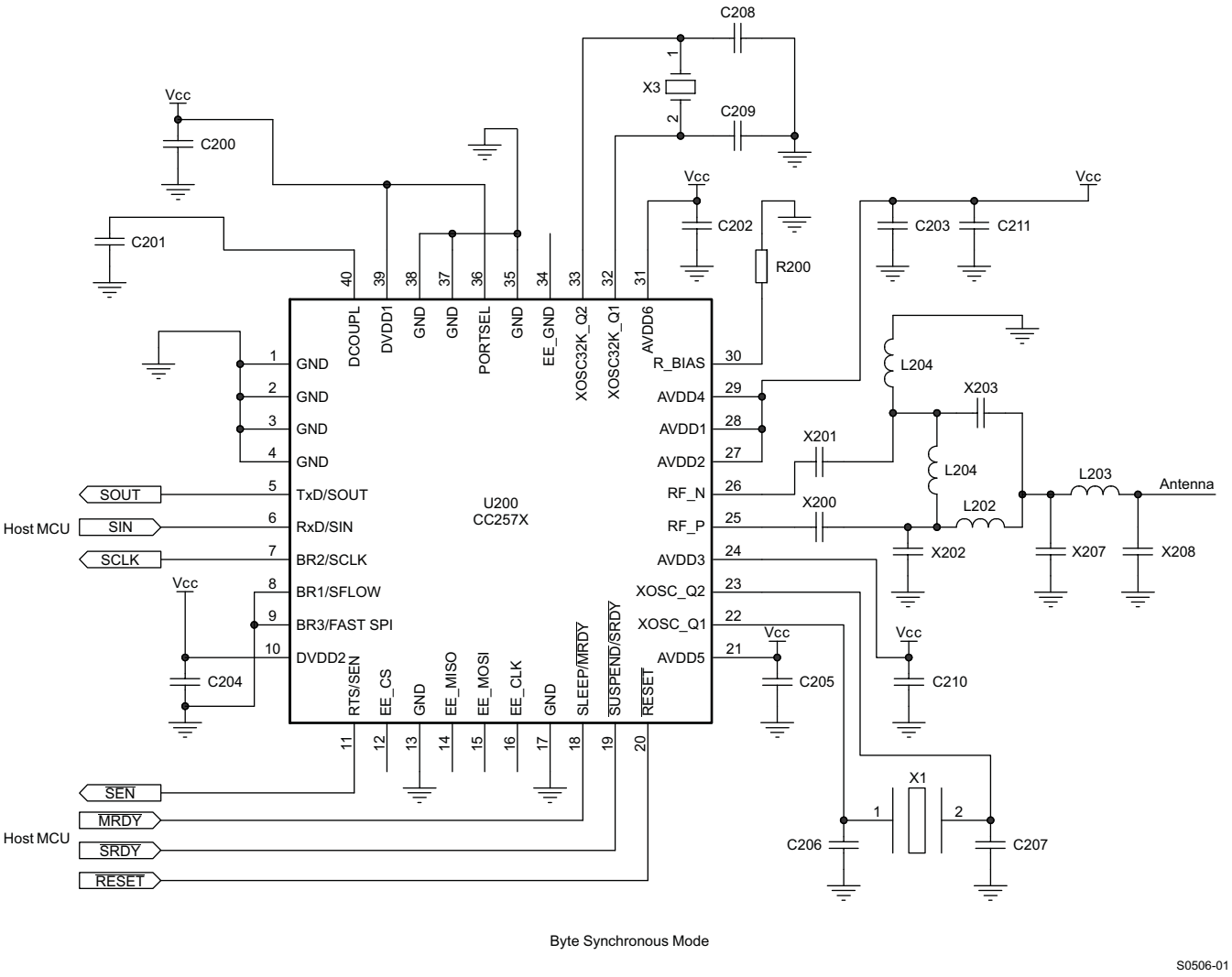


Figure 11. Synchronous Reset Timing

The synchronous reset may also be used in systems where it is not feasible to route a pin exclusively for reset, thereby reducing the number of pins required to interface to ANT.

BYTE-SYNCHRONOUS SERIAL INTERFACE

The byte-synchronous serial interface may be used in conjunction with an SPI (three-wire) peripheral on a host MCU. Additional control lines are used to control the power states and flow of the data. The CC257x always acts as the SPI master in the interface. To select the byte-synchronous serial interface, the PORTSEL pin must be set to VCC and the SFLOW pin must be set to GND. This is illustrated in Figure 12.



Byte Synchronous Mode

S0506-01

Figure 12. Byte-Synchronous Serial Hardware Setup

The speed of the SPI clock can be set to either 500 kHz or 4 MHz by setting FAST SPI (pin 9) as specified in Table 3.

Table 3. Byte-Synchronous Clock Speed Selection

FAST SPI (Pin 9)	SPI CLK Speed
HIGH	4 MHz
LOW	500 kHz

The byte-synchronous protocol provides flow control at the byte level. If the host MCU has a message to send to the CC257x, it first asserts the SMSGRDY signal to indicate a readiness to communicate. The CC257x responds by asserting the SEN signal (if it is not already asserted). The host must then pulse the SRDY signal before each byte is written or read. Note that the first byte is always sent to the host MCU to indicate if ANT is ready to receive the message (0xA5), or if ANT has a message that it must send (0xA4). In the latter case, the host should first read out the message from ANT before attempting to send its own message. All bits are sent LSB-first. The timing for a host-to-ANT transaction is illustrated in Figure 13.

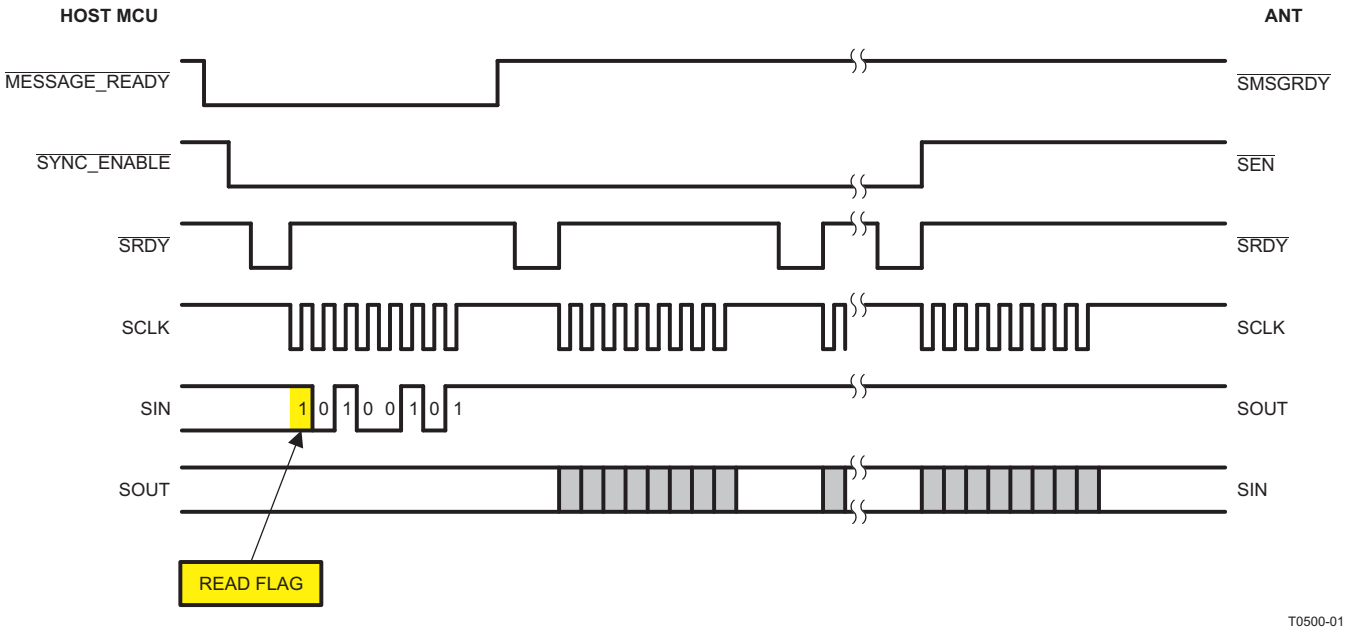


Figure 13. Synchronous ANT-to-Host Transaction

Whenever ANT has a message to send to the host, it asserts SEN to indicate a readiness to communicate. For this reason, it is recommended that SEN be connected to an interrupt-capable pin on the host processor, so the host can be woken up from a sleep state anytime ANT requires communication. An ANT-to-host serial transaction is very similar to a host-to-ANT transaction. A timing diagram is shown in Figure 14. Once SEN is detected to be asserted, the host pulses SRDY. The first byte received is 0xA4, indicating that ANT has a message pending. Subsequent bytes are clocked out following each assertion of the SRDY signal.

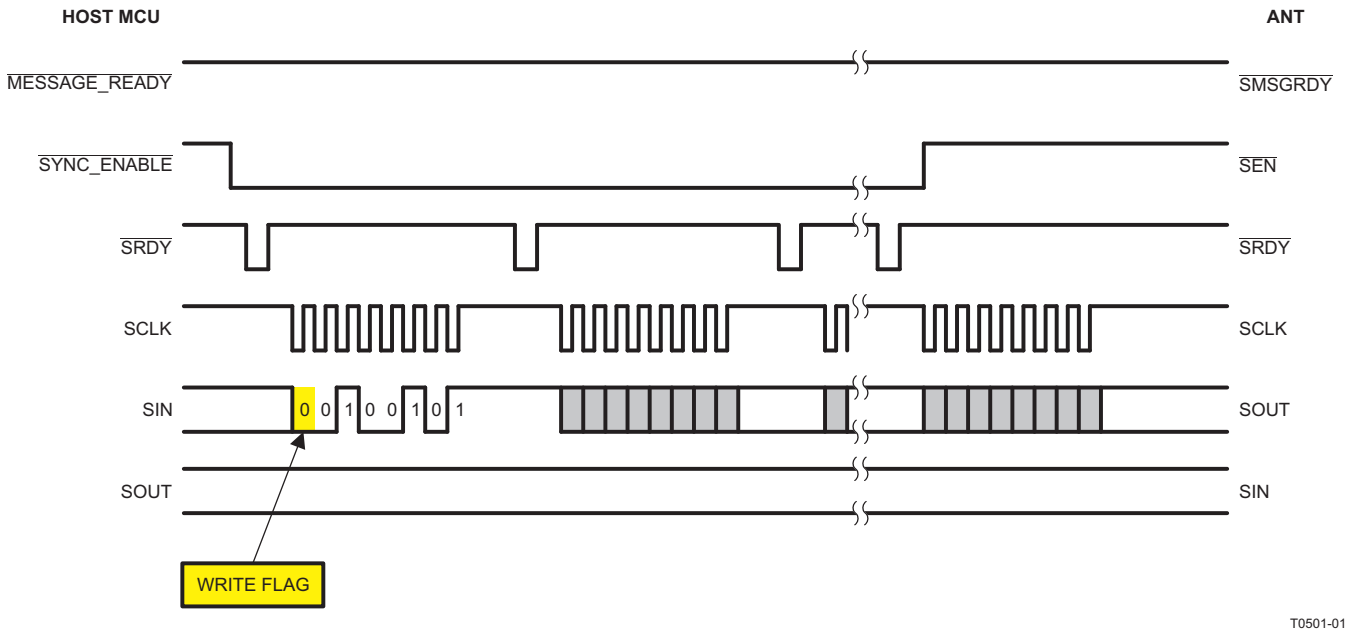


Figure 14. Synchronous Host-to-ANT Transaction

BIT-SYNCHRONOUS SERIAL INTERFACE

The bit synchronous serial protocol is designed to enable a host MCU to communicate with ANT using GPIO's only. The protocol on the host side can be implemented entirely in software, allowing the user to select a very inexpensive MCU, or to dedicate peripherals to other devices in the system. The difference between the bit synchronous and the byte synchronous protocol is that the bit synchronous protocol controls the flow of serial information on a 'per bit' level as opposed to 'per byte'. This means that the SRDY signal will need to be pulsed for each bit that is to be transported. For this reason the bit synchronous serial protocol will generally be higher power and lower data rate than the byte synchronous protocol. All other timing and characteristics are the same. To enable bit synchronous serial communications both the PORTSEL and SFLOW pin should be tied to VCC. See Figure 15 for hardware setup.

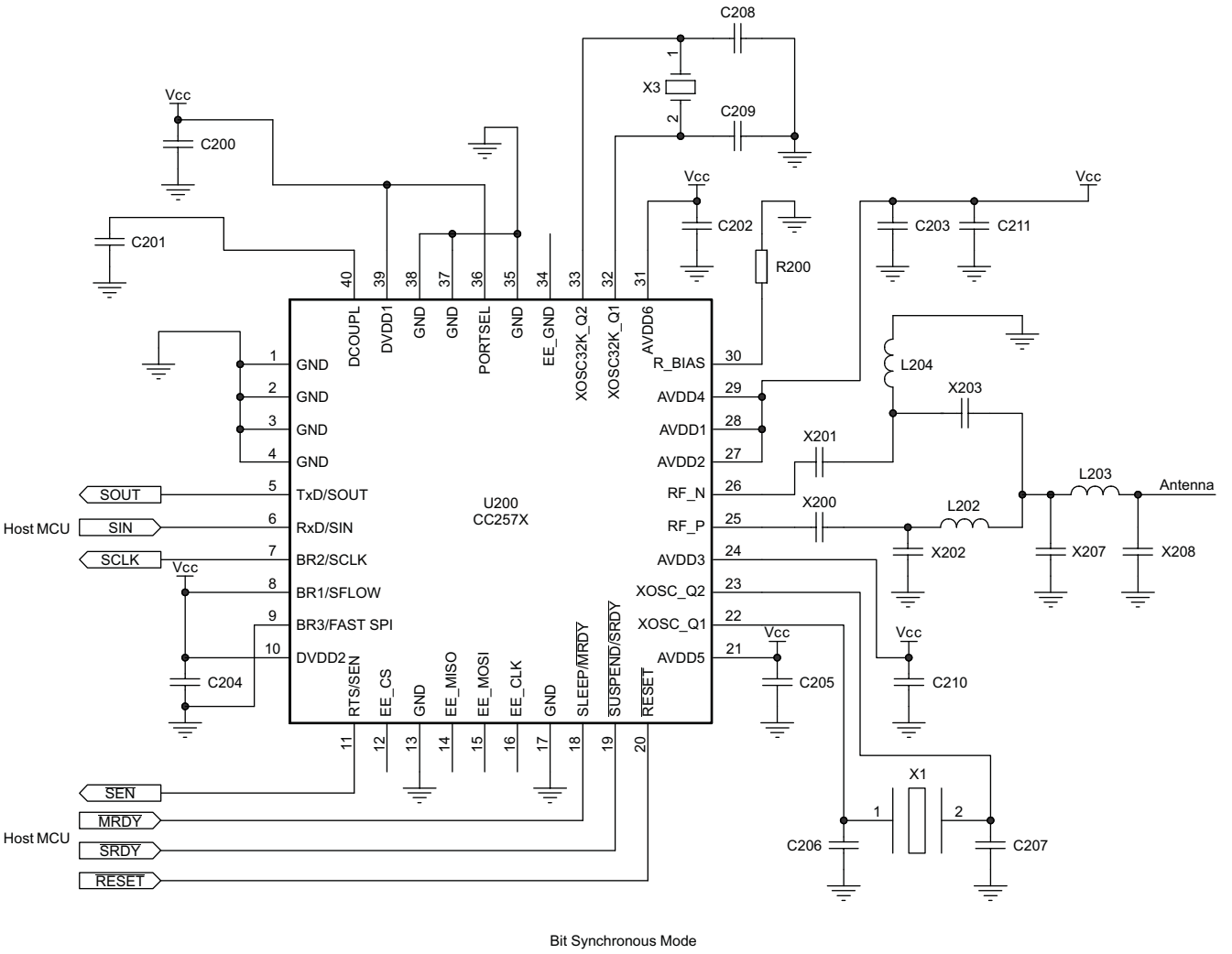


Figure 15. Bit-Synchronous Serial Hardware Setup

For a host-to-ANT transaction, the host must have the data ready on the SIN pin prior to the falling edge of the clock prior to the SRDY pulse. This is illustrated in the figure below.

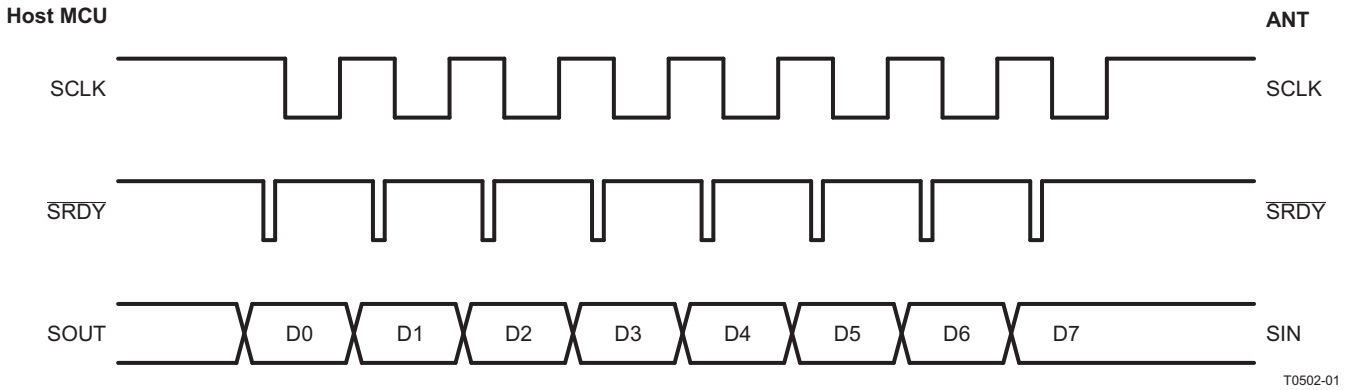


Figure 16. Bit-Synchronous Flow Control

For an ANT-to-host transaction, the host must read the SOUT pin following the rising edge of the clock, which is between the rising edge and the subsequent SRDY pulse. This is illustrated in Figure 17.

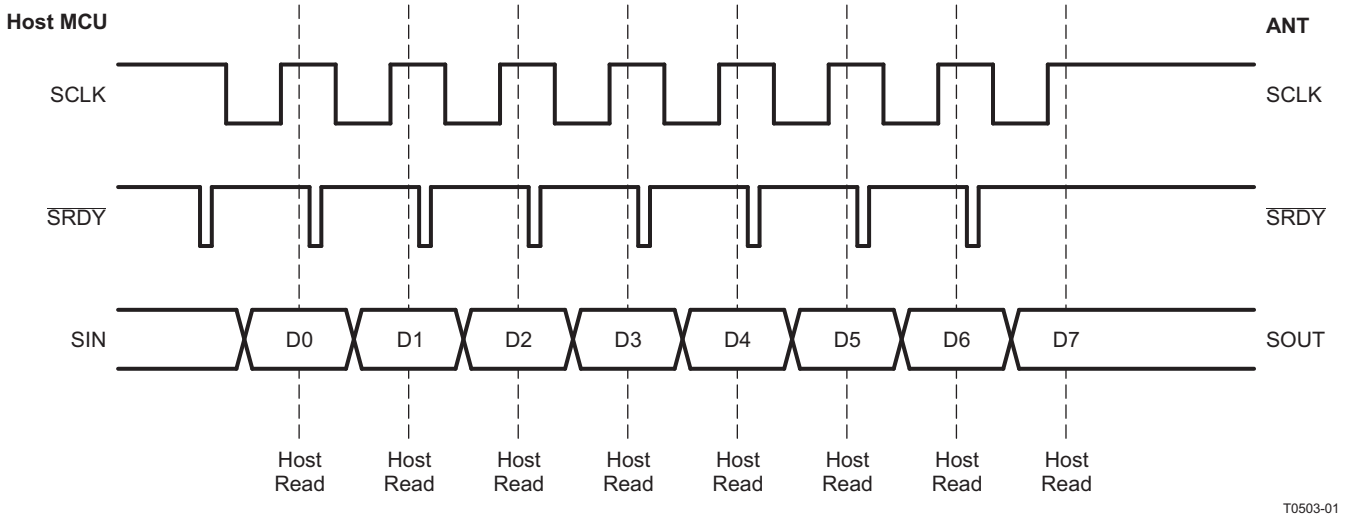


Figure 17. Bit Synchronous Read/Write Edge

ANT/HOST INTERFACE PROTOCOL

The host/ANT protocol consists of a simple command/response protocol. With the exception of the reset and request commands, each message sent to ANT is followed by a response message. Data messages and channel events are always sent to the host by ANT (there is no need or mechanism to request these messages explicitly). The general package structure of the serial host/ANT interface is described in Figure 18 and Table 4.

Figure 18. ANT Serial Message Structure

Table 4. ANT Serial Message Format

Byte	Name	Length	Description
0	SYNC	1	Sync byte = 0xA4
1	LENGTH	1	Number of data bytes in message (n)
2	ID	1	Message identifier
3 to n + 2	DATA1–DATAn	n	Message data bytes
n + 3	CHECKSUM	1	XOR of all previous bytes.

NOTE

Important: It should be noted that for the asynchronous serial mode, the SYNC byte is always sent as part of the messages and is always 0xA4. For the synchronous serial protocol, the SYNC byte is always sent by ANT and is 0xA4 for ANT-to-host transactions and 0xA5 for host-to ANT-transactions.

For complete details on the ANT/host interface, see the ANT document *ANT Message Protocol and Usage*. Table 5 is a list of the command set applicable to the CC257x. The Requested column indicates that the message must be requested by the host. The Event column indicates that the message is sent by ANT to the host without any request from the host—for example, after a channel event. The Reply column indicates that the command elicits a response from ANT.

Table 5. ANT Serial Messages

Class	Message(1)	ID	Description	Reply	Requested	Event
Config messages	UNASSIGN_CHANNEL_ID	0x41	Unassign a channel.	Yes	No	No
	ASSIGN_CHANNEL_ID	0x42	Assign a channel.	Yes	No	No
	CHANNEL_ID_ID	0x51	Set channel ID.	Yes	No	No
	CHANNEL_MESG_PERIOD_ID	0x43	Set channel period.	Yes	No	No
	CHANNEL_SEARCH_TIMEOUT_ID	0x44	Set the channel-search time-out period.	Yes	No	No
	CHANNEL_RADIO_FREQ_ID	0x45	Set the channel RF frequency.	Yes	No	No
	NETWORK_KEY_ID	0x46	Set the network key.	Yes	No	No
	RADIO_TX_POWER_ID	0x47	Set the transmit power (all channels)	Yes	No	No
	ID_LIST_ADD_ID	0x59	Add channel ID to search list.	Yes	No	No
	ID_LIST_CONFIG_ID	0x5A	Configure search list.	Yes	No	No
	CHANNEL_RADIO_TX_POWER_ID	0x60	Set the channel transmit power.	Yes	No	No
	SET_LP_SEARCH_TIMEOUT_ID	0x60	Set the low-priority search time-out.	Yes	No	No
	RX_EXT_MESGS_ENABLE_ID	0x66	Enable extended messages.	Yes	No	No
	AUTO_FREQ_CONFIG_ID	0x70	Configure frequency agility feature.	Yes	No	No
	PROX_SEARCH_CONFIG_ID	0x71	Set proximity search.	Yes	No	No
Notification	STARTUP_MESG_ID	0x6F	Startup message (following reset).	No	No	Yes
Control	SYSTEM_RESET_ID	0x4A	Reset system.	No	No	No
	OPEN_CHANNEL_ID	0x4B	Open a channel.	Yes	No	No
	CLOSE_CHANNEL_ID	0x4C	Close a channel.	Yes	No	No
	OPEN_RX_SCAN_ID	0x5B	Open channel in scan mode.	Yes	No	No
	REQUEST_ID	0x4D	Request a message from ANT.	–	No	No
Data messages	BROADCAST_DATA_ID	0x4E	Broadcast message (ANT→host and host→ANT).	No	No	Yes
	ACKNOWLEDGED_DATA_ID	0x4F	Acknowledged message (ANT→host and host→ANT).	No	No	Yes
	BURST_DATA_ID	0x50	Burst message (ANT→host and host→ANT).	No	No	Yes
Event	RESPONSE_EVENT_ID	0x40	Channel event message.	–	No	Yes

(1) Precede with MESG_

Table 5. ANT Serial Messages (continued)

Class	Message[1]	ID	Description	Reply	Requested	Event
Response messages	CHANNEL_STATUS_ID	0x52	Reports the channel status.	–	Yes	No
	CHANNEL_ID_ID	0x51	Reports the channel ID.	–	Yes	No
	VERSION_ID	0x3E	Reports the ANT version.	–	Yes	No
	CAPABILITIES_ID	0x54	Reports the capabilities.	–	Yes	No
Test mode	RADIO_CW_INIT_ID	0x53	Initialize CW (continuous-wave) mode.	Yes	No	No
	RADIO_CW_MODE_ID	0x48	Start CW mode.	Yes	No	No

ANT/HOST INTERFACE PROTOCOL

The CC257x has ANT-FS client functionality embedded directly on the chip. This includes the over-the-air ANT-FS protocol extension and the required file system utilities to manage stored data in non-volatile storage. This allows the user to connect an EEPROM directly to the CC257x. Raw files can be sent and managed by the CC257x and downloaded over ANT-FS as desired. The diagram below depicts how to connect an external EEPROM to the CC257x. The connection is SPI. Note that only EEPROM (not flash) is supported.

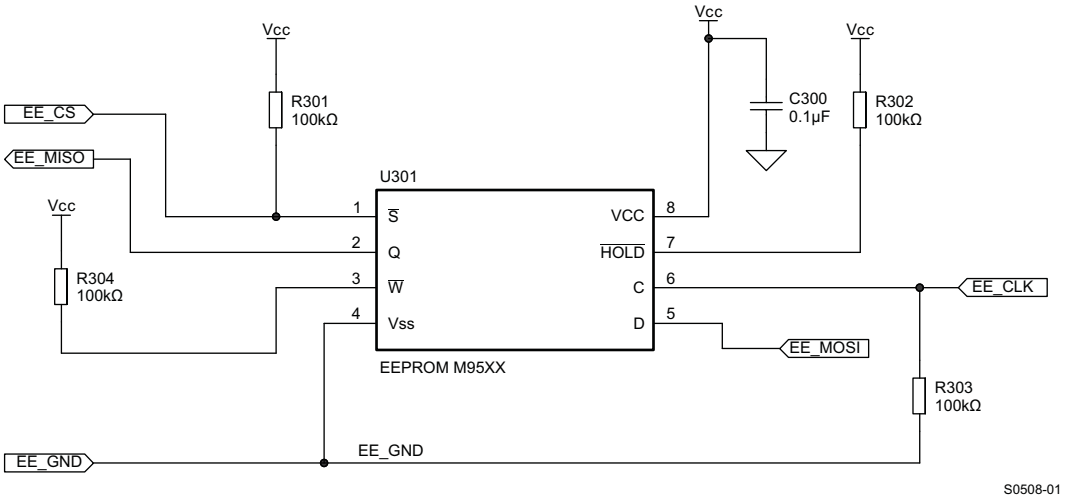


Figure 19. EEPROM Hardware Setup

If an EEPROM is NOT used, all EEPROM signals should NC (non-connect). The file system and ANT-FS protocol are managed by using an extended version of the ANT/host serial protocol. For complete details on how to use this protocol, see the ANT document *FS_ANTFS Serial Message*.

The extended host/ANT interface uses a 2-byte message ID to identify FS and ANT-FS specific messages. The packet structure is detailed in Figure 20 and Table 6.

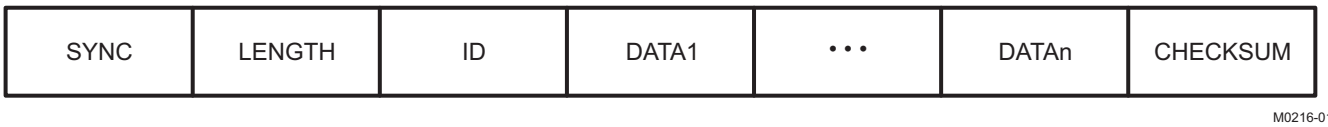


Figure 20. ANT Extended Serial Message Structure

Table 6. ANT Extended Serial Message Format

Byte	Name	Length	Description
0	SYNC	1	Sync byte = 0xA4
1	LENGTH	1	Number of data bytes in message (n)
2	ID	2	Message identifier
3 to n + 2	DATA1–DATAn	n	Message data bytes
n + 3	CHECKSUM	1	XOR of all previous bytes.

All extended message IDs begin with the top 3 MSBs set to 1 (0xE0). Note that responses and events for ANT-FS also use the extended message ID format. A summary of FS and ANT-FS specific commands are listed in the table below.

Table 7. FS/ANT-FS Specific Serial Messages

Class	Message[1]	ID	Description	Reply	Requested	Event
MEMDev commands	EEPROM_INIT[2]	0xE220	Initialize the EEPROM memory device.	Yes	No	No
FS commands	INIT_MEMORY	0xE200	Initializes the FS on the memory device. Saved memory preserved.			
	FORMAT_MEMORY	0xE201	Format memory.	Yes	No	No
	DIRECTORY_SAVE	0xE207	Saves all open files to memory device.	Yes	No	No
	DIRECTORY_BUILD	0xE209	Rebuild FS directory and condense directory size.	Yes	No	No
	FILE_DELETE	0xE20C	Delete existing open file.	Yes	No	No
	FILE_CLOSE	0xE20D	Close open flag or file.	Yes	No	No
	FILE_SET_SPECIFIC_FLAGS	0xE212	Update application-defined flags on file.	Yes	No	No
	REQUEST_ID[3]	0xE100	Extended request message	–	No	No
FS requests	GET_USED_SPACE	0xE202	Returns the number of used bytes in FS in sector-sized increments.	–	Yes	No
	GET_FREE_SPACE	0xE203	Return the number of free bytes in sector-sized increments	–	Yes	No
	FIND_FILE_INDEX	0xE204	Returns the file index of first file in directory that matches specified identifier.	–	Yes	No
	DIRECTORY_READ_ABSOLUTE	0xE205	Read from absolute offset into directory as if it were an ANT-FS directory.	–	Yes	No
	DIRECTORY_READ_ENTRY	0xE206	Returns ANT-FS directory entry for the file matching the specified file index.	–	Yes	No
	DIRECTORY_GET_SIZE	0xE208	Return size in bytes of ANT-FS directory.	–	Yes	No
	FILE_CREATE	0xE20A	Allocates a free sector and saves directory entry of new file.	–	Yes	No
	FILE_OPEN	0xE20B	Open an existing file.	–	Yes	No
	FILE_READ_ABSOLUTE	0xE20E	Read from absolute offset into a file.	–	Yes	No
	FILE_READ_RELATIVE	0xE20F	Read from current read pointer in a file.	–	Yes	No
	FILE_WRITE_ABSOLUTE	0xE210	Write to absolute offset into a file.	–	Yes	No
	FILE_WRITE_RELATIVE	0xE211	Write to current write pointer in a file.	–	Yes	No
	FILE_GET_SIZE	0xE213	Return size of open file in bytes.	–	Yes	No
	FILE_GET_SPECIFIC_FILE_FLAGS	0xE214	Returns specific flags of open file.	–	Yes	No

(1) Unless otherwise specified, precede with MESG_FS_
(2) Precede with MESG_MEMDEV_
(3) Precede with MESG_EXT_

Table 7. FS/ANT-FS Specific Serial Messages (continued)

Class	Message[1]	ID	Description	Reply	Requested	Event
FS crypto commands	CRYPTO_ADD_USER_KEY_ INDEX	0xE245	Adds specified user key to be stored in internal memory.	Yes	No	No
	CRYPTO_SET_USER_KEY_ INDEX	0xE246	Specify stored user key to be used by FS encryption/decryption process.	Yes	No	No
	CRYPTO_SET_USER_KEY_VAL	0xE247	Specify non-user key to be used by FS encryption/decryption process	Yes	No	No
ANT-FS commands	ANTFS_OPEN	0xE231	Opens ANT-FS beacon.	Yes	No	No
	ANTFS_CLOSE	0xE232	Closes ANT-FS beacon.	Yes	No	No
	ANTFS_CONFIG_BEACON	0xE233	Configure ANT-FS beacon.	Yes	No	No
	ANTFS_SET_AUTH_STRING	0xE234	Set authentication string.	Yes	No	No
	ANTFS_SET_BEACON_STATE	0xE235	Set beacon state.	Yes	No	No
	ANTFS_PAIR_RESPONSE	0xE236	Respond to pairing request.	Yes	No	No
	ANTFS_SET_LINK_FREQ	0xE237	Set Link state RF frequency.	Yes	No	No
	ANTFS_SET_BEACON_ TIMEOUT	0xE238	Set beacon timeout.	Yes	No	No
	ANTFS_SET_PAIRING_ TIMEOUT	0xE239	Set pairing timeout.	Yes	No	No
	ANTFS_REMOTE_FILE_ CREATE_EN	0xE23A	Enable creation of remote file.	Yes	No	No
	SYSTEM_TIME	0xE23D	Set system time.	Yes	No	No
ANT-FS requests	ANTFS_GET_CMD_PIPE	0xE23B	Get commands from command pipe.	–	Yes	No
	ANTFS_SET_CMD_PIPE	0xE23C	Write command to command pipe.	–	Yes	No
	SYSTEM_TIME	0xE23D	Request system time.	–	Yes	No
ANT-FS events	RESPONSE_ID	0xE000	Extended event message with payload. See table xx for list of ANT-FS events.	–	No	Yes

ANT-FS generates events to mark important occurrences during the ANT-FS transaction. These events are delivered to the host using the 0xE000 message. It is not necessary to request this message; rather, events are generated and sent by ANT to the host as required. The table below summarizes the ANT-FS events.

Table 8. ANT-FS Event Messages

Event[1]	Code	Description
PAIR_REQUEST	0x01	A pairing request has been received from the ANT-FS Host.
DOWNLOAD_START	0x02	The download of a file was started by an ANT-FS Host.
UPLOAD_START	0x03	The upload of a file by an ANT-FS Host has started.
DOWNLOAD_COMPLETE	0x04	The download of a file by an ANT-FS Host has completed.
UPLOAD_COMPLETE	0x05	The upload of a file from an ANT-FS Host has completed.
ERASE_COMPLETE	0x06	The erase of a file has been completed.
LINK_STATE	0x07	The ANT-FS client has entered the LINK state.
AUTH_STATE	0x08	The ANT-FS client has entered the AUTH state.
TRANSPORT_STATE	0x09	The ANT-FS client has entered the TRANSPORT state.
CMD_RECEIVED	0x0A	A command was received on the command pipe.
CMD_PROCESSED	0x0B	A command was received and processed on the command pipe.

(1) Precede with MESG_FS_ANTFS_EVENT_

The general procedure for configuring an ANT-FS channel is as follows:

1. Set network as usual.
2. Assign a master channel and set the channel ID for each desired beacon.
3. Send beacon configuration message (0xE233).
4. Set authentication strings if desired (0xE234). The default is no friendly name and no passkey.
5. Set beacon time-out (0xE238) and pairing timeout (0xE239), if desired. The default settings are 10s and 240s, respectively.
6. Set a link beacon frequency message (0xE237) for each beacon channel. A beacon channel can be disabled by setting the beacon frequency to 0xFF.
7. Start the session by sending the open ANT-FS message (0xE231).
8. Once an ANT-FS host connects, an AUTH_EVENT on the corresponding channel occurs. At this point, all other beacons are broadcasting a beacon indicating the BUSY state.
9. If a pairing request occurs (event 0x01), it can be accepted or rejected by sending a pairing request response (0xE236).
10. Once authentication passes, ANT-FS is in the TRANSPORT state. Events are generated as upload and download requests are made by the host. The host MCU is also informed if command-pipe commands have been received and whether or not they have been processed by the CC257x automatically.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
CC2570RHAR	ACTIVE	VQFN	RHA	40	2500	Green (RoHS & no Sb/Br)	NIPDAU	Level-3-260C-168 HR	-40 to 85	CC2570	Samples
CC2570RHAT	ACTIVE	VQFN	RHA	40	250	Green (RoHS & no Sb/Br)	NIPDAU	Level-3-260C-168 HR	-40 to 85	CC2570	Samples
CC2571RHAR	ACTIVE	VQFN	RHA	40	2500	Green (RoHS & no Sb/Br)	NIPDAU	Level-3-260C-168 HR	-40 to 85	CC2571	Samples
CC2571RHAT	ACTIVE	VQFN	RHA	40	250	Green (RoHS & no Sb/Br)	NIPDAU	Level-3-260C-168 HR	-40 to 85	CC2571	Samples

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".
RoHS Exempt: TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.
Green: TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

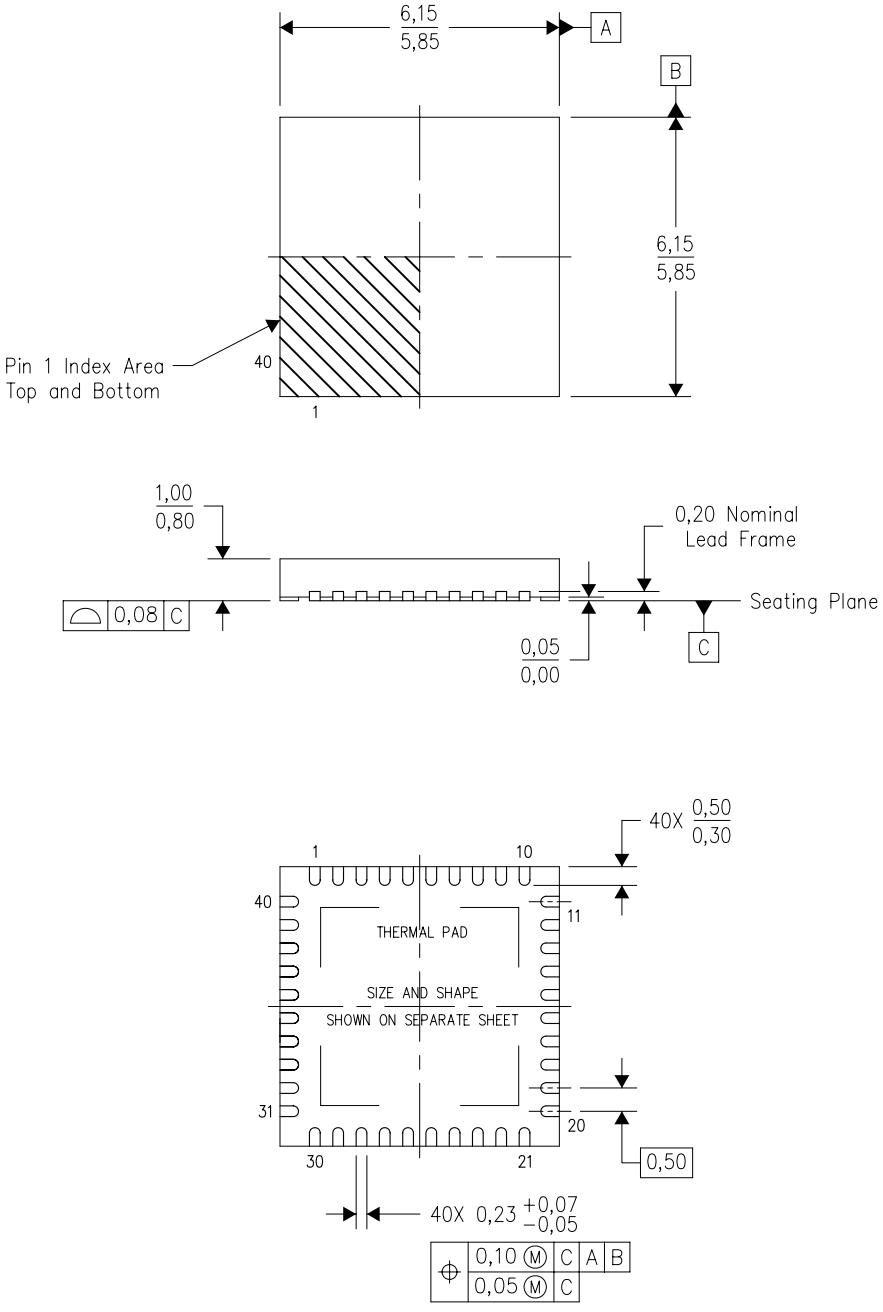
(6) Lead/Ball Finish - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead/Ball Finish values may wrap to two lines if the finish value exceeds the maximum column width.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and

continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals.
TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

RHA (S–PVQFN–N40) PLASTIC QUAD FLATPACK NO–LEAD



Bottom View

4204276/E 06/11

NOTES:
A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M–1994.
B. This drawing is subject to change without notice.
C. QFN (Quad Flatpack No–Lead) Package configuration.
D. The package thermal pad must be soldered to the board for thermal and mechanical performance.
E. See the additional figure in the Product Data Sheet for details regarding the exposed thermal pad features and dimensions.
F. Package complies to JEDEC MO–220 variation VJJD–2.

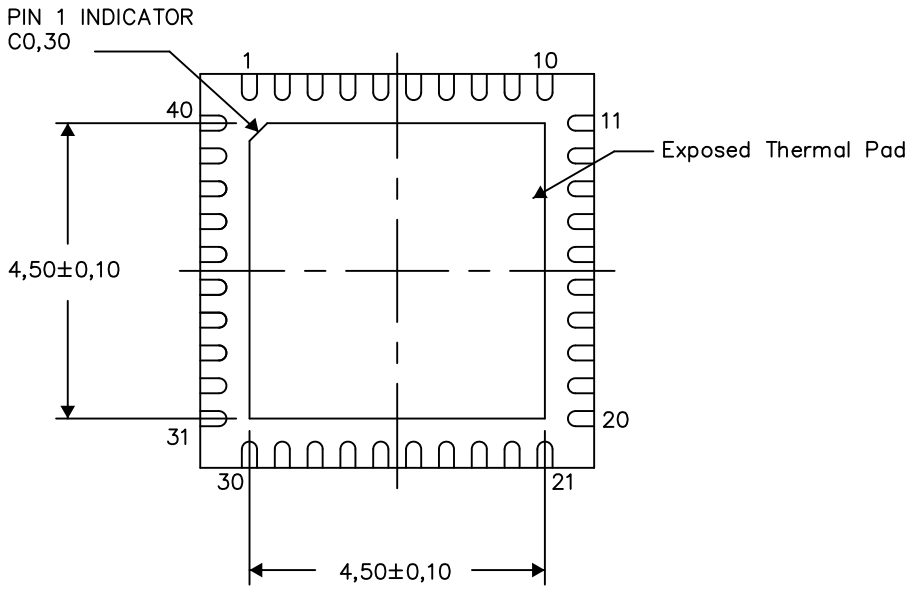
RHA (S–PVQFN–N40) PLASTIC QUAD FLATPACK NO–LEAD

THERMAL INFORMATION

This package incorporates an exposed thermal pad that is designed to be attached directly to an external heatsink. The thermal pad must be soldered directly to the printed circuit board (PCB). After soldering, the PCB can be used as a heatsink. In addition, through the use of thermal vias, the thermal pad can be attached directly to the appropriate copper plane shown in the electrical schematic for the device, or alternatively, can be attached to a special heatsink structure designed into the PCB. This design optimizes the heat transfer from the integrated circuit (IC).

For information on the Quad Flatpack No–Lead (QFN) package and its advantages, refer to Application Report, QFN/SON PCB Attachment, Texas Instruments Literature No. SLUA271. This document is available at www.ti.com.

The exposed thermal pad dimensions for this package are shown in the following illustration.



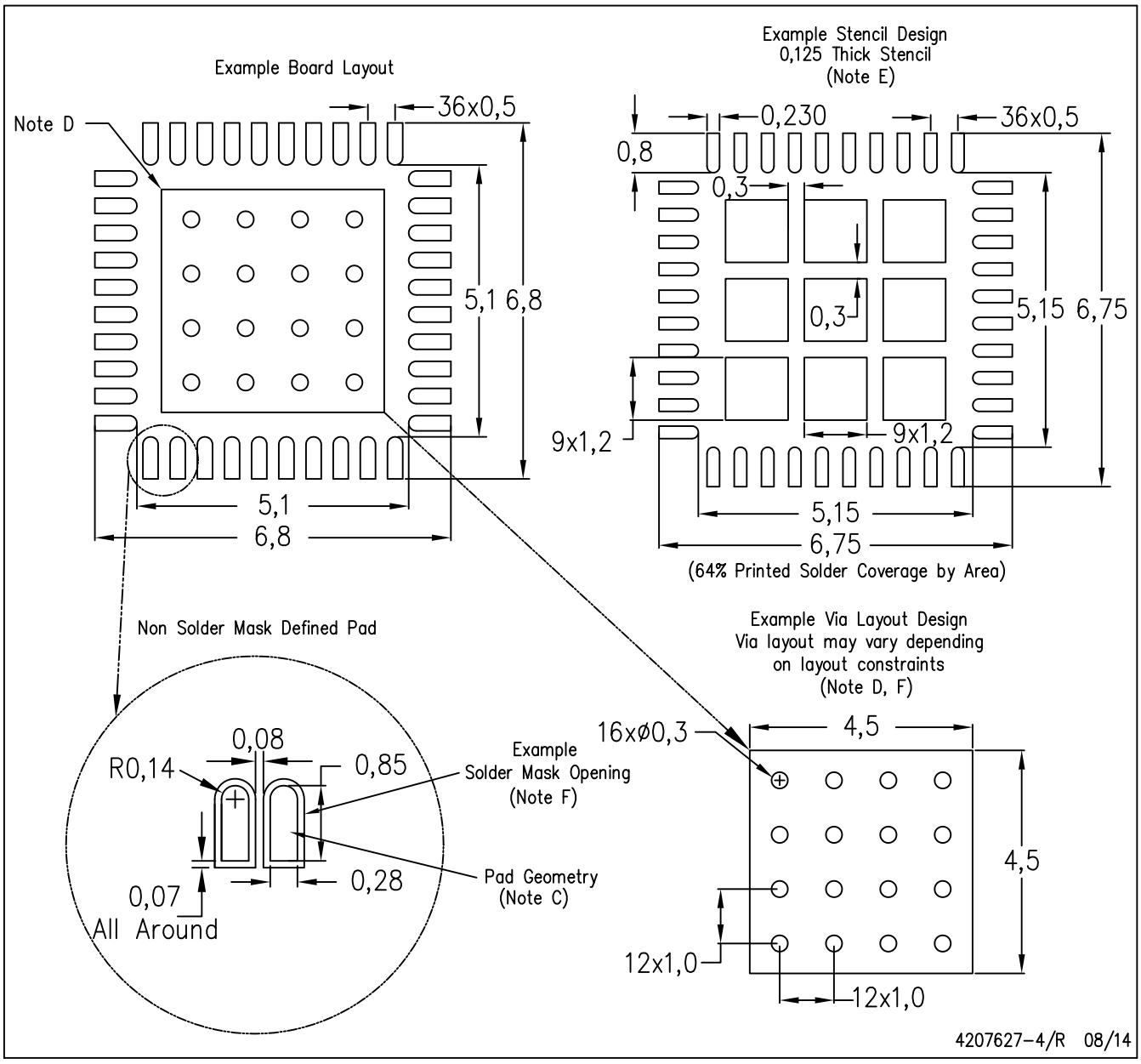
Bottom View

Exposed Thermal Pad Dimensions

4206355–4/X 08/14

NOTES: A. All linear dimensions are in millimeters

RHA (S–PVQFN–N40) PLASTIC QUAD FLATPACK NO–LEAD



NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Publication IPC–7351 is recommended for alternate designs.
D. This package is designed to be soldered to a thermal pad on the board. Refer to Application Note, Quad Flat–Pack Packages, Texas Instruments Literature No. SLUA271, and also the Product Data Sheets for specific thermal information, via requirements, and recommended board layout. These documents are available at www.ti.com <http://www.ti.com>.
E. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Refer to IPC 7525 for stencil design considerations.
F. Customers should contact their board fabrication site for recommended solder mask tolerances and via tenting recommendations for vias placed in the thermal pad.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265