# Burst Transfers


## ABSTRACT

The ANT Burst transfer mode offers a fast and efficient method for transferring bulk data through the ANT wireless link. Rather than using an increased message rate, the ANT Burst transfer mode can achieve higher data throughput while maintaining a simple serial protocol and interface. ANT Burst transfers use a rapid series of acknowledged messages for transferring data with automatic retries and a 3-bit embedded sequence to ensure successful transmission and data integrity. The serial interface protocol is similar to that during normal operation; however, flow control signals rather than event messages are used to trigger the host for more data. The effects of serial interface speed on data throughput are discussed and various design considerations, techniques and ANT features to optimize data throughput are also described.




P +1 403.932.4620 F +1 403.932.6521

228 River Avenue, Cochrane, Alberta, Canada T4C 2C1 thisisant.com

# TABLE OF CONTENTS



## LIST OF FIGURES

# 1 Introduction

Many wireless applications have a need for bulk data transfer. Although such transfers are often occasional, they typically require faster communication rates than the standard mode of operation. ANT offers a burst transfer mode which is ideal for efficiently transferring bulk data at significantly faster rates. Additional benefits include automatic retries of lost message packets and a simple application interface for serial data transfer.

The following application note provides a detailed explanation of the ANT burst mode.

# 2 Relevant Documents

It is strongly recommended that the following documents be reviewed prior to using this application note.

- ANT Message Protocol and Usage
- Interfacing with ANT General Purpose Chipsets and Modules
- ANT chip/module datasheets

# 3 ANT Burst Mode

There are two ways to increase data throughput on an ANT device: increasing the message rate, or using burst mode.

ANT Burst mode optimizes data transfer over the serial interface by using hardware signaling, instead of event messages, for pipelining data to/from ANT. The result is more condensed serial activity which allows for faster data throughput. Bursting can be thought of as extending the allotted channel timeslot and inserting additional transmit pulses at a burst rate of 300 Hz. This results in a 20kbps maximum data throughput when in a favorable RF environment, and when an appropriate serial interface is used (see section 5).

Figure 1.a shows a master device operating in its typical mode at a channel period of $T_{ch}$, which is optimized for low power and typical data throughput. Occasionally the master device needs to send larger amounts of data which it can do by sending the data in burst mode (b), or by increasing the message rate (c). When data is sent in burst mode, as can be seen in Figure 1.b, it is sent every $T_{burst}$ (i.e. 1/burst_rate). Once the burst transfer is complete, the master once again transmits at Tch, maintaining the lower power communication state. Increasing the message rate, as shown in Figure 1.c, will result in higher data throughput, however the lower power transmission state is lost, and maximum data throughput is still significantly less than that achieved through bursting.

Additional to the increased throughput advantage, ANT Burst mode uses acknowledged messages with automatic retries to ensure successful data transfer while maintaining a simple interface to the application MCU.

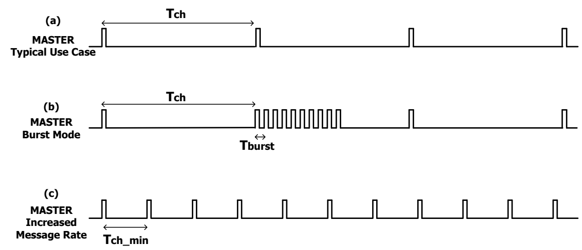


**Figure 1. ANT Burst mode vs Increased Message Rate**

As explained in the ANT Message Protocol and Usage document, a 3-bit sequence number is also embedded in each data packet to further increase data integrity and application control.

# 4 Data Throughput

RF link quality and serial interface configuration are the two prominent factors that affect the data throughput in ANT Burst mode.

### 4.1 RF Link Quality

ANT Burst mode automatically retries lost packets up to a pre-defined maximum number of times, and data throughput can be significantly affected by the quality of the RF link. In other words, the more packets ANT has to retry, the lower the data throughput.

### 4.2 Serial Interface Setup

The rate at which the application MCU can provide new data to ANT directly affects the throughput of ANT Burst mode. The maximum data throughput is 20kbps. This maximum throughput can be achieved using either bit or byte synchronous modes or asynchronous mode at 50000 baud or higher (57600). A slower baud rate will result in throughputs less than 20kbps.

ANT Burst mode is designed to operate at baud rates of 19200 and higher. Efficiency is less than adequate at lower serial interface speeds.

# 5 Serial Interface Protocol

As mentioned above, ANT Burst mode uses flow control signaling to efficiently pipeline data messages from the application MCU to ANT. For simplicity, the serial interface signaling during bursting is very similar to the serial protocol during normal operation. The application MCU need only be aware of the absence of EVENT_TX messages which are usually used to send the next data message to ANT.

### 5.1 Asynchronous Mode

In asynchronous mode, flow control of data from host to ANT during a burst transfer is performed by the RTS signal. ANT asserts (logic 1) this signal to halt message transfers from the application MCU. Similarly, ANT de-asserts (logic 0) RTS to allow the host to send a data message.

The timing diagram below illustrates the ANT burst mode signaling using an asynchronous serial interface; showing transfer of an entire message packet including sync byte (0xA4), message length (ML), message ID (ID), 8-byte data message (D0:D7) and checksum (CS).

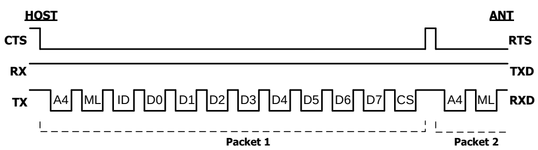


**Figure 2. ANT Burst mode signaling using asynchronous serial mode**

To achieve the fastest data throughput, the application MCU must respond to the RTS transitions as rapidly as possible. Figure 2, above, shows only the serial activity for a burst transfer that is going from host MCU to ANT. For a burst receive (i.e. ANT to host), the data will appear on the RX line. Note, there is no flow control for data in this direction; therefore the application MCU must be able to receive data at any time.

### 5.2 Synchronous mode

The SEN signal is used for flow control when using a synchronous (bit or byte) serial interface. This signal is asserted (logic 0) to enable the transfer of a data message and de-asserted (logic 1) to halt any transfers. From the host MCU's perspective, the operation of the serial interface is the same for burst mode as it is for normal operation. The difference is that SEN will be signaling at the burst rate (Tburst) rather than the message rate (Tch).

Figure 2 shows the timing diagram of a burst transfer from application MCU to ANT using the byte synchronous serial mode. Again, the figure shows the transfer of an entire message packet from sync byte (0xA5 for serial host-> ANT) to checksum (CS). If bit synchronous mode were used instead, SRDY would be pulsed for each bit, rather than each byte.

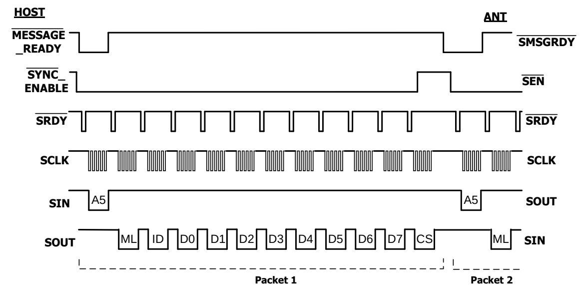


**Figure 3. ANT Burst mode signaling using byte synchronous serial mode**

The application MCU must ensure a fast response to the flow control to achieve the fastest data throughput.

### 5.3 Baud Rate Considerations

As mentioned in the prior sections, the application MCU's ability to provide data quickly to ANT can effect the data throughput while bursting. In other words, ANT can only send data over the air as quickly as the data is provided by the host. Not only does the serial interface's speed of data transfer affect data throughput, it also has an effect on the burst mode's immunity to failure and the receiving device's power consumption.

In burst mode, ANT will only transmit new data on the next designated burst (not channel) timeslot. If no new data has been presented by the host, ANT will not retransmit the old data. The receiver, however, will be active during the next burst timeslot anticipating the next burst packet. As no packet was transmitted, the receiver will register a failed packet. This effectively "wastes" one of the allocated retries, hence affecting failure immunity, as well as wasting power due to the receiver's radio being active.

Transmit queues, as explained in a later section, can be used to counter the serial port latency and help reduce the effects of slower serial interfaces in some situations.

## 6 Techniques for Burst Control

When requesting a burst data transfer, the host can send packets to "prime" ANT's buffers prior to the next channel timeslot. Take, for example, an ANT node typically sending broadcast messages at channel period Tch. As shown in Figure 4, the application can use EVENT_TX to initiate the burst transfer and start sending burst data packets. Once ANT's buffers are full, the flow control line will go to logic 1. This will indicate to the host that no further data can be sent to ANT.

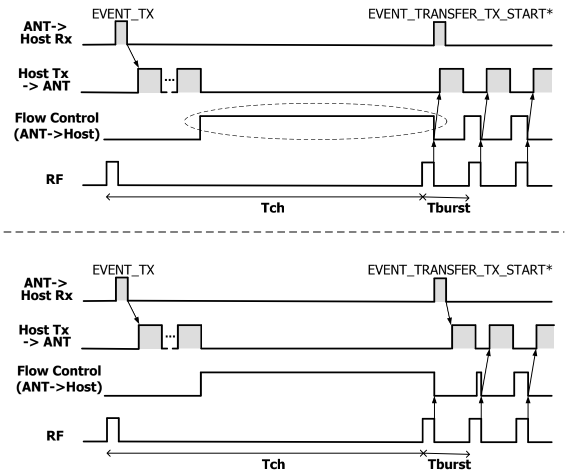


**Figure 4. Polling Flow Control (above) vs using event message (below)**

The application MCU can then monitor the flow control signal to determine when the next packet can be sent. Once "primed", ANT will hold the flow control signal at logic 1 until bursting commences and the first packet is sent on the next channel timeslot. Once this packet is sent over the air, ANT is able to buffer more data and will indicate this to the host by setting the flow control signal to logic 0. The time between priming ANT and the bursting beginning (circled) is relatively significant.

An alternative to monitoring the flow control signal would be to use the EVENT_TRANSFER_TX_START[*] message. This event message allows the host to prime ANT's buffers and then continue with other processing. Upon receiving the EVENT_TRANSFER_TX_START[*] message, the application MCU can resume sending burst data to ANT. For those devices that do not have this message, monitoring flow control line is the only method available.

Unless a transmit queue is present (described below), ANT can be primed with two burst packets.

# 7 Transmit Queues

On some devices (refer to datasheets for capabilities), ANT has in-built transmit queues which provide the ability to buffer up to 9 data packets (or messages) per channel.

[*] EVENT_TRANSFER_TX_START message is only available on some ANT devices. Refer to datasheets for capabilities.

Up to 8 data packets can be sent without "locking up" the serial interface. In other words, the flow control signal returns to logic 0 and further messages (data or command) can be sent from host to ANT. The timing diagram above the dashed line in Figure 5, shows the transfer of 8 data messages (M0:M7) from host to ANT prior to bursting on the next channel timeslot. Note that after the $8^{th}$ message (M7), the flow control line has returned to logic 0. This allows the host MCU to continue to communicate with ANT in the remaining time prior to the start of the burst.

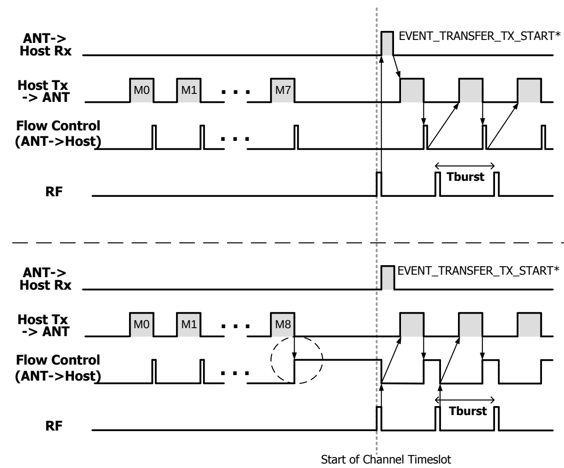


**Figure 5. Queuing and effects on flow control: 8 data packets (above) vs 9 data packets (below)**

The timing diagram below the dashed line in Figure 5 shows 9 data packets (M0:M8) sent to ANT. Flow control remains logic 1 after the $9^{th}$ message (circled), preventing any further communication with ANT until the burst commences and flow control returns to logic 0.

As described in section 6, the host application can either monitor the flow control signal or use the EVENT_TRANSFER_TX_START[*] message to indicate when the burst has commenced and ANT is able to receive more data messages.

Transmit queues alleviate the effects of slower serial ports, providing a buffer against the latency in transferring data from host to ANT. They are ideal for slow serial interface applications requiring small burst transfers.

# 8 Event Messages

As explained in the ANT Message Protocol and Usage document, ANT generates event messages and sends them to the host either in response to a message or as generated by an RF event. The event messages relevant to bursting are detailed below:

### 8.1 EVENT_TRANSFER_TX_COMPLETED

This message will be generated on the node that initiated the burst transfer, indicating that the entire burst message was successfully transmitted.

### 8.2 EVENT_TRANSFER_TX_FAILED

This message will be generated on the node that initiated the burst transfer, indicating that the burst transfer was not successfully transmitted. This message does not indicate at which point in the transfer the message failed. It is recommended to resend the entire burst transfer rather than just sending the remaining packets. Despite the fact that the previously successful packets will be re-transmitted, this is far simpler than trying to calculate at which point the transfer failed. The protocol for this would be implemented at the application level and would need to be agreed upon by both nodes in the established communication channel. Determining which data packet in the burst failed is difficult as it is dependent not just on which message the host was preparing to send to ANT, but on how many other messages were already buffered on ANT.

### 8.3 EVENT_TRANSFER_RX_FAILED

This message will be generated on the node receiving the burst, indicating that the receive transfer has failed. This occurs when a burst transfer message was incorrectly received after the maximum number of retries. For example, a 20 packet burst transfer was initiated and 15 packets successfully received. The 16th packet failed to transmit and the receiving node's ANT will inform the host of the failure with this message. Depending on the application, the host can use this to determine if the next burst is new data, a repeated attempt of the same data, or even if the burst is just sending the remaining data. The protocol for this would be implemented at the application level and would need to be agreed upon by both nodes in the established communication channel. It is recommended to resend the entire burst transfer rather than just sending the remaining packets. Despite the fact that the previously successful packets will be re-transmitted, this is far simpler than trying to calculate at which point the transfer failed.

### 8.4 EVENT_TRANSFER_TX_START*

This message will be generated on the node that initiated the burst transfer, indicating that the burst transfer has commenced. Effectively, this message is sent on the next channel period, when the first burst message is sent over the RF channel.

*Not available on all ANT devices. Refer to datasheets.

### 8.5 TRANSFER_IN_PROGRESS

This message will be generated on the node that initiated the burst transfer. This occurs when the host requests a burst and there is already a burst (or acknowledged) message pending or in progress. This will occur whether the active/pending burst (or acknowledged) message is on the same or different channel.

### 8.6 TRANSFER_SEQUENCE_NUMBER_ERROR

This message will be generated on the node that initiated the burst transfer. This occurs when the host sends a packet with a sequence number that is not in the order expected.

### 8.7 TRANSFER_IN_ERROR

This message will be generated on the node that initiated the burst transfer. It is returned when a burst message passes the sequence number check, but has the wrong channel number. This indicates to the host that the transfer is in error and should be halted as soon as possible.

# 9 Closing Remarks

This application is aimed at providing a detailed description of the ANT burst mode along with its features, advantages and proper usage.