

# Power States

## ABSTRACT

The ANT wireless protocol achieves extremely low power consumption by carefully managing its on-chip resources. The base current of the ANT MCU will depend on the ANT solution chosen, the type of serial communication implemented, and the presence or absence of an external 32kHz clock. To fully achieve minimal power usage in an application, it is important to understand how ANT manages its power and what the application processor's role should be to achieve this result.

## COPYRIGHT INFORMATION

This application note is the property of Dynastream Innovations Inc. and is intended for limited circulation only. Any reproduction or distribution without written consent from Dynastream Innovations Inc. is strictly prohibited.

© 2016 Dynastream Innovations Inc. All rights reserved.

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION .....</b>	<b>3</b>
<b>2</b>	<b>RELEVANT DOCUMENTS .....</b>	<b>3</b>
<b>3</b>	<b>POWER STATES IN ASYNCHRONOUS SERIAL MODE .....</b>	<b>3</b>
3.1	AT3 AND AP1/AP2 WITH 32KHz EXTERNAL CLOCK SOURCE. ....	3
3.1.1	ACTIVE STATE.....	4
3.1.2	IDLE STATE.....	4
3.1.3	SLEEP STATE.....	4
3.1.4	SUSPEND STATE.....	4
3.1.5	DEEP SLEEP STATE.....	5
3.2	AP1 AND AP2 POWER STATES WITHOUT EXTERNAL 32 KHz CLOCK SOURCE.....	5
3.3	RESETTING IN ASYNCH MODE .....	5
<b>4</b>	<b>POWER STATES IN SYNCH SERIAL MODE .....</b>	<b>5</b>
<b>5</b>	<b>CLOSING REMARKS.....</b>	<b>6</b>

## LIST OF FIGURES

Figure 1. Possible Power States and Transitions in Asynchronous Serial Mode .....	3
Figure 2. AP2 DEEP SLEEP .....	5
Figure 3. Resetting using the <i>SUSPEND</i> signal .....	5
Figure 4. Possible Power States and Transitions in Synchronous Serial Mode.....	6

## LIST OF TABLES

Table 1. State Transitions in Asynch Mode.....	3
Table 2. ANT Power States in Asynchronous Mode. ....	4
Table 3. AP1 and AP2 Power States w/o 32kHz Ext. Clock .....	5
Table 4. ANT Power States in Synchronous Mode.....	6

## 1 Introduction

The ANT wireless protocol has been optimized to use minimal resources to achieve ultra-low power, wireless communication. It facilitates outfitting small, coin cell operated devices with the ability to communicate wirelessly, often allowing for years of operation without having to replace the battery. The power consumption of ANT depends on several factors including ANT hardware chosen, data type of messaging used, message rate, and serial interface. An estimated average current consumption for a given implementation can be easily calculated by using current tables and formulas available in the particular ANT chipset's or module's datasheet. Alternatively, the online power calculator can be used to get the same estimate. To achieve the theoretical current values given by the data sheet or calculator, it is important to understand ANT's power states and how they are affected by the external HW configuration and serial interface. This application note discusses the power states of the AP1, AP2 and AT3 chipsets focusing on the effects of external 32 kHz clock sources and the serial interface.

## 2 Relevant Documents

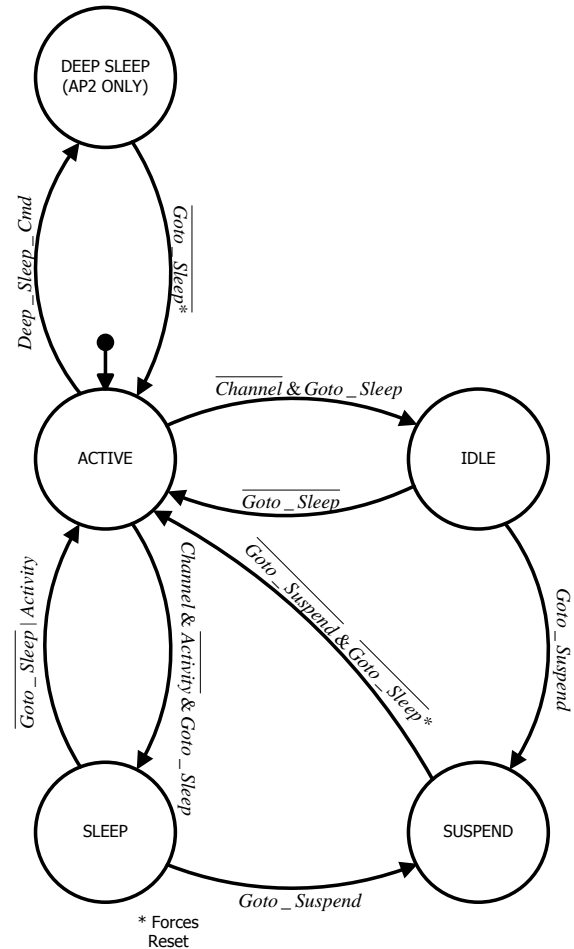
It is highly recommended that the following documents be read and understood prior to using this application note.

- Interfacing with ANT General Purpose Chipsets and Modules
- ANT Message Protocol and Usage
- ANT Chip/Module Datasheet

## 3 Power States in Asynchronous Serial Mode

When running in asynchronous serial mode, the power states of ANT are controlled by the host MCU using the SLEEP and SUSPEND lines. The states and state transitions for the AP1, AP2 and AT3 are mostly identical, differing only in power consumption and timing between the different solutions and states available. The AP2 introduces a new state known as DEEP SLEEP, which is controlled by a serial command.

For the AP1 and AP2 devices, the presence of an external 32 kHz clock or crystal source strongly influences power consumption and possible states. Figure 1 depicts *all* power states and transitions available to ANT in asynchronous serial mode.



**Figure 1. Possible Power States and Transitions in Asynchronous Serial Mode**

Table 1 summarizes the state transitions. These are also elaborated upon in the following sections.

**Table 1. State Transitions in Asynch Mode**

Transition	Description
Goto_Sleep	SLEEP signal asserted (active high)
Goto_Suspend	SUSPEND signal asserted (active low)
Activity	RF activity or events that require the CPU to become active.
Channel	Indicates at least 1 ANT channel open
Deep_Sleep_Cmd	Deep sleep command (0xC5 serial message)

The power states are described in detail in the following sections.

### 3.1 AT3 and AP1/AP2 with 32kHz external clock source.

When interfacing to ANT using an asynchronous serial connection, optimal power states may be

achieved on the AP1/AP2 if an external 32 kHz clock is provided. The table below illustrates the power states available to ANT depending on HW solution and assuming an external 32 kHz clock source on the AP1 and AP2.

**Table 2. ANT Power States in Asynchronous Mode.**

Power State	AP1*	AT3	AP2*
Active	~3mA	~2.5mA	~3mA
Suspend	70uA	1.1uA	2uA
Sleep	30uA	2.6uA	3uA
Idle	2uA	1.1uA	2uA
Deep	n/a	n/a	0.5uA

ANT will not traverse through these states automatically. It is up to the host MCU to control the power states by using the SLEEP and SUSPEND signals appropriately.

### 3.1.1 ACTIVE STATE

On start-up or after a reset, ANT will be in the ACTIVE state. This state represents the active current mode of the CPU, necessary to process any serial messages. All clocks are active while ANT is in this state. ANT must be in this state in order to receive messages from the host MCU and to send messages to the host. If no ANT channels are open, the host may put ANT into the IDLE state by asserting the SLEEP signal. This is indicated in Figure 1 by the (*Channel* & *Goto\_Sleep*) transition from the ACTIVE state.

### 3.1.2 IDLE STATE

The IDLE state generally represents the lowest power state of a device, except for the AP2 which has a DEEP SLEEP state available. The CPU clock and all peripheral clocks are not active in this state. There can be no open ANT channels while in the IDLE state. The IDLE state can be used when ANT needs to be in standby mode: for example, when the device does not need to communicate for a prolonged period of time. Entering the IDLE state does NOT erase any configurations from ANT. If channels have been configured, they do not need to be re-configured before being opened again (after exiting the IDLE state). The host MCU may not send any messages to ANT while it is in the IDLE state. As no ANT channels can be open, the host MCU should not expect to receive any messages from ANT. The host may reset ANT by asserting *SUSPEND* while it is in the IDLE state. If the host wishes to send serial messages to ANT, the SLEEP signal must first be de-asserted, this is indicated by the

(*Goto\_Sleep*) transition in Figure 1. This action will put ANT back into the ACTIVE state.

### 3.1.3 SLEEP STATE

If any ANT channels are open, then asserting SLEEP while in ACTIVE mode will put ANT into the SLEEP state. This is indicated by the (*Channel* & *Activity* & *Goto\_Sleep*) transition in Figure 1. While in this state, the timers necessary to run ANT are active. The peripheral clock needed to service the serial port is disabled. If ANT needs to process any RF activity while it is in SLEEP state, it will transition to the ACTIVE state and return to SLEEP provided that the SLEEP signal is still asserted. This is indicated in Figure 1 by the inclusion of the *Activity* condition in transitions between ACTIVE and SLEEP. Hence, the power consumption of ANT will fluctuate while the host has the SLEEP signal asserted, depending on how much RF activity it needs to process. Any RF data messages or channel events will also be passed to the host MCU. If the host is in the SLEEP state and all channels become closed (for example if a slave channel times out), ANT will revert to the IDLE state (provided SLEEP is still asserted). ANT may also enter the SUSPEND state from the SLEEP state by asserting *SUSPEND*. If the host wishes to send any serial messages to ANT, the SLEEP signal must first be de-asserted.

For optimal power consumption, the SLEEP signal should be asserted by the host any time the host does not need to be sending serial messages to ANT. To send a serial message to ANT, de-assert the SLEEP signal and ensure the RTS signal is not asserted. Depending on the channel state, ANT will either be in SLEEP or IDLE state while the SLEEP signal is asserted. The host must always be ready to receive messages from ANT unless it is in the IDLE state.

### 3.1.4 SUSPEND STATE

The SUSPEND state is required to allow ANT to quickly enter a known and stable low-power mode. It is not necessarily the lowest possible power state and it forces all ANT channels to immediately close. This is useful for applications such as USB sticks that require suspend functionality. The transition to the SUSPEND state is also an easy way to reset ANT, as a reset will occur each time ANT is taken out of SUSPEND. To exit, the *SUSPEND* signal must be de-asserted prior to the de-assertion of the SLEEP signal. This is indicated in Figure 1 with the (*Goto\_Suspend* & *Goto\_Sleep*) transition. The SUSPEND state should only be utilized if its unique functionality is required in an application.

For general operation, if it is desired to enter a low power mode for standby purposes, the host MCU should first close all ANT channels and then assert the SLEEP signal. This will ensure that ANT is in its lowest power, or IDLE, state.

### 3.1.5 DEEP SLEEP STATE

The AP2 introduces a new low power state, known as DEEP SLEEP. This state may be entered into by sending the SLEEP serial message (0xC5) followed by the assertion of the SLEEP signal. The SLEEP signal must be asserted within 1.2ms. If SLEEP is not asserted within this time window, ANT will reset and move to IDLE mode. DEEP SLEEP causes a hard reset of the ANT device. Any previous channel configurations will be erased. The timing restrictions around the use of the SLEEP command are illustrated in Figure 2.

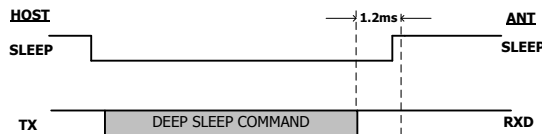


Figure 2. AP2 DEEP SLEEP

The DEEP SLEEP command is not available in the AP1 or AT3.

### 3.2 AP1 and AP2 Power States without external 32 kHz clock source.

Without an external clock, the AP1 and AP2 will not be able to fully realize the low power potential described in the previous sections. This is because the 32 kHz clock will need to be synthesized from the internal 16 MHz clock, effectively reducing the number of low power states available.

The AP1 can still enter all 4 states, ACTIVE, SLEEP, IDLE and SUSPEND. However, the current consumption in the SUSPEND, SLEEP and IDLE state will be the same (70uA). Transitions into and out of these states, and behavior while in these states does not change.

The AP2 will still be able to enter all of its low power states, including the ultra low DEEP SLEEP state. The only difference is that the power consumption while in the SLEEP state will be 100uA. The power consumption of the other power states does not change. Transitions into and out of these states, and behavior while in these states does not change.

A summary of AP1 and AP2 power states is described in table 3.

Table 3. AP1 and AP2 Power States w/o 32kHz Ext. Clock

Power State	AP1*	AP2*
Active	~3mA	~3mA
Suspend	70uA	2uA
Sleep	70uA	100uA
Idle	70uA	2uA
Deep	n/a	0.5uA

### 3.3 Resetting in Asynch Mode

As described in Section 3, the SUSPEND STATE may only be entered from the IDLE or SLEEP states. This requires that the SLEEP signal be asserted before the SUSPEND signal. If the SUSPEND signal is asserted directly from the ACTIVE state, ANT will reset and return to the ACTIVE state immediately. This is a useful feature that may be utilized for resetting ANT if the asynchronous interface is used. Figure 3 illustrates this behavior.

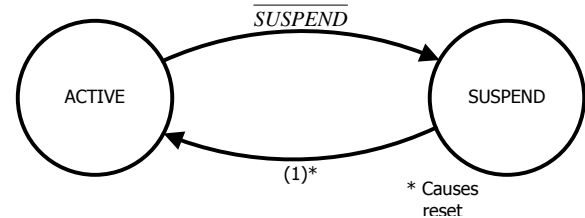


Figure 3. Resetting using the  $\overline{SUSPEND}$  signal

## 4 Power States in Synch Serial Mode

When running in synchronous mode, power state transitions are generally automatic. However, it is useful to understand the ANT power states such that observed behavior can be understood and verified. As synchronous mode is not generally used for USB devices, the SUSPEND state is not supported. As with asynchronous mode, the presence of an external 32 kHz clock makes a significant difference in the power consumption of the AP1 and AP2. Figure 4 depicts the states and transitions available in synchronous serial mode. Please refer to Table 2 and Table 3 for current consumption numbers, as these do not change for the given states in either synchronous or asynchronous mode.

On start-up or after reset, ANT will be in the ACTIVE state. If no channels are open and if  $\overline{SMSGRDY}$  is NOT asserted, ANT will

automatically transition to the IDLE state. To send a message to ANT, the host needs to first assert *SMSGRDY*, thus putting ANT into the ACTIVE state.

Once an ANT channel has been opened, ANT can only operate in one of two states: ACTIVE and SLEEP. ANT will transition to the SLEEP state any time it is waiting for an SRDY pulse from the host MCU.

As with asynchronous mode, the AP2 utilizes a DEEP SLEEP state which may be entered by sending the DEEP SLEEP command. Unlike asynchronous mode, there are no additional timing constraints to enter this mode if the synchronous message protocol is implemented properly.

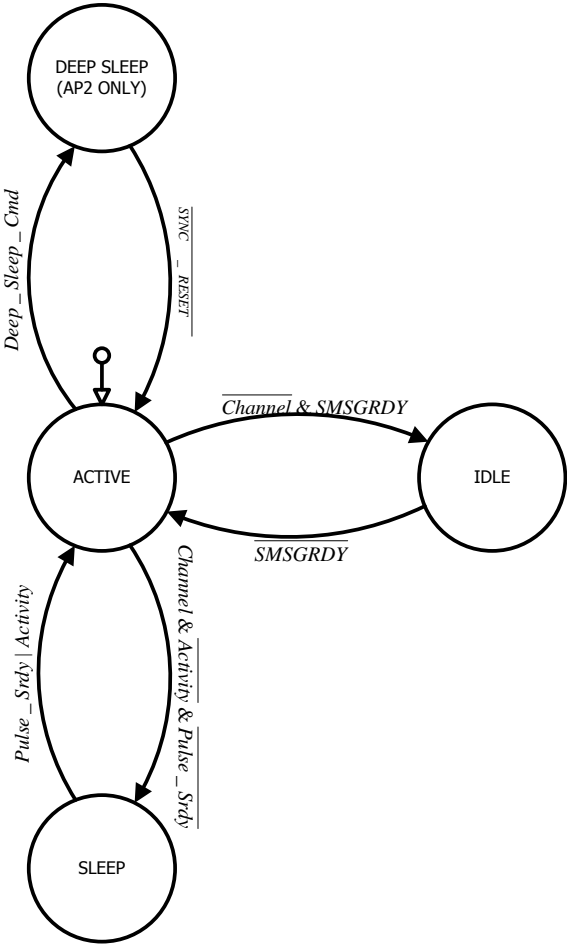


Figure 4. Possible Power States and Transitions in Synchronous Serial Mode

The transitions in Figure 4 are detailed in Table 4.

Table 4. ANT Power States in Synchronous Mode

Transition	Description
<i>Pulse_Srdy</i>	Sync pulse from host to ANT (per bit or per byte depending on interface used)
<i>SMSGRDY</i>	Message ready signal from host to ANT
<i>Activity</i>	RF activity or events that require the CPU to become active.
<i>Channel</i>	Indicates at least 1 ANT channel open
<i>SYNC_RESET</i>	Synchronous Reset
<i>Deep_Sleep_Cmd</i>	Deep Sleep command (serial message 0xC5)

5 Closing Remarks

This application note is an accompaniment to the “Interfacing with ANT General Purpose Chipsets and Modules” document. It discusses the power states available to ANT while operating in synchronous and asynchronous serial communication modes. An understanding of these concepts is critical to realizing an estimated or targeted low power solution utilizing the ANT wireless protocol.