Clock Drift Test with ANT Extended Messages

ABSTRACT

This document introduces a method to measure the clock drift of ANT devices with respect to a tested golden unit. This method uses the timestamp information that can be provided in ANT extended messages.

TABLE OF CONTENTS

LIST OF FIGURES

1 Introduction

System clock stability is one of the key requirements for reliable wireless communication. The type of crystal used greatly affects the system clock stability in embedded designs.

The ANT protocol was developed with a high tolerance to clock drift in order to withstand a maximum drift of 50 parts per million (ppm), while still maintaining a stable channel. However, poor quality crystals can cause significant clock drift that will affect ANT performance; particularly under severe circumstances. For example, extremely low or high temperatures can cause significant clock drifts leading to link degradation.

The purpose of this application note is to provide a reliable and simple method for using ANT extended messages to identify and measure crystal-caused clock drift in an ANT device.

2 Relevant Documents

It is strongly recommended that the following documents be reviewed prior to using this application note. To ensure you are using the current versions, check the ANT+ website at www.thisisant.com or contact your ANT+ representative.

- ANT Message Protocol and Usage
- Module or chip specific datasheets
- AN14 – Continuous Scanning Mode for Asynchronous Topologies

3 Background Information and Test Methodology

3.1 ANT Extended Message Timestamp Format

Certain ANT devices have the capability to pass additional information to the host, along with the received data messages. This message format, known as the 'Extended Message Format', is of two types: legacy and flagged. The flagged format allows passing different types of additional information indicated by a flag byte, including flag 0x20; timestamp information. This timestamp is generated by the slave when it receives a message from the master. It is formed using a two byte value that increments by one on every slave clock tick, and rolls over after every (2^{16}/clock frequency) seconds..

$$Clock\ frequency = f\ Hz \Leftrightarrow Number\ of\ clock\ ticks\ every\ second = f$$

$$Minimum\ time\ measurable = Time\ between\ successive\ ticks = \frac{1}{f}\ s$$

$$Rollover\ time = \ Maximum\ tick\ count\ recordable\ \times Time\ between\ successive\ ticks = \frac{2^{16}}{f} s$$

Note that both the slave and the master clocks influence the recorded timestamp values, as the master clock determines when each data message is sent and the slave clock generates the timestamp on receipt of the data message. The elapsed time between successive received messages can be compared with the configured message period to estimate the clock drift in the master-slave system.

If the ANT device used as the slave is known to be a golden unit, this method can be used to calculate the clock drift in the other device, which is the unit under test (UUT). The tester is responsible for obtaining (and determining an acceptable standard for) the golden unit. Refer to the ANT Message Protocol and Usage document for device capability information.

Figure 1 illustrates the extended message data format with the timestamp flag turned on. Note that the standard ANT data format consisting of Sync, Message Length (ML), Message ID (ID), Channel Number (C#), 8 byte data payload, and Checksum (CS) is maintained; however, the extended format additionally includes the flag byte (0x20) and extended message bytes containing the timestamp information. Refer to the ANT Message Protocol and Usage for further details.

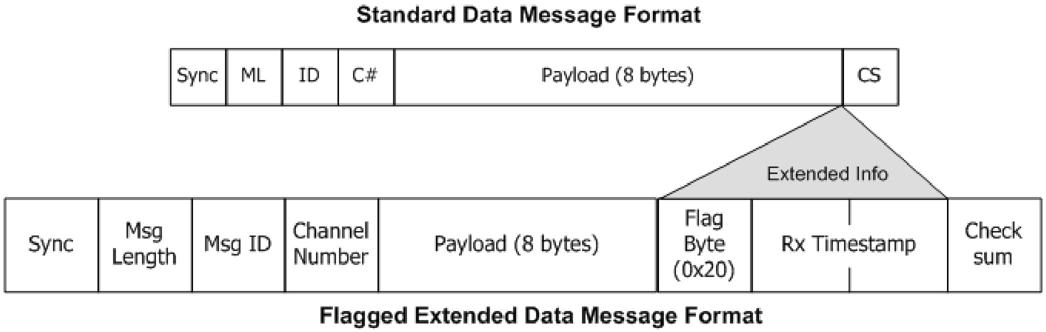

Figure 1. ANT Flagged Extended Message Format with Timestamp

3.2 Enabling ANT Extended Messages

The timestamp flagged ANT extended message format must be enabled on the slave device before the channel is opened. This is done using the Lib Config command (message ID 0x6E) with the flag 0x20. Please see the platform specific ANT library documentation for implementation details. An example of the implementation using the ANT .NET Windows Library Package is as follows:

```
// Example usage
ANTDevice.setLibConfig(ANT_ReferenceLibrary.LibConfigFlags.MESG_OUT_INC_TIME_STAMP_0x20);
```

As the response, the ANT MCU will send the response message (message ID 0x40) with the channel number, initiating message ID (0x6E), and response code RESPONSE_NO_ERROR (0x00) to confirm that the timestamp extended message has been enabled.

3.3 Detecting and Measuring Clock Drift

To measure the clock drift, the master should be configured as a master transmit only channel (channel type 0x50) in order to disable ANT's adaptive isochronous capability.

An ANT module capable of supporting flagged extended messages (e.g. AP2, C7, N5, ANTUSB-m, or ANTUSB2) is required. This device should be configured as a slave, with timestamp flagged extended messages enabled.

To obtain the clock drift from the message timestamps, the expected time taken to complete the test is calculated using the claimed message period and number of messages received, and is compared to that calculated using the received timestamps. As each message is received, the difference between successive timestamps is calculated and added to a zero initialized value. This is the measured elapsed time (T_m). Note that the timestamp rollover, that occurs nearly every two seconds for a 32.768 KHz clock, must be accounted for in this calculation. Simultaneously, the expected number of clock ticks for each message period is added to the (zero initialized) expected time (T_e). At the end of the test duration, the difference between these two values is calculated.

If the calculated difference is zero, there is no measured clock drift. If however, the difference is non-zero, clock drift exists in the system. This accumulated difference is caused due to the clock drift ($\pm\Delta$) that occurs in every message cycle, causing the actual message period to diverge from the expected one (MsgPrd). This is illustrated in the example shown in Figure 2.

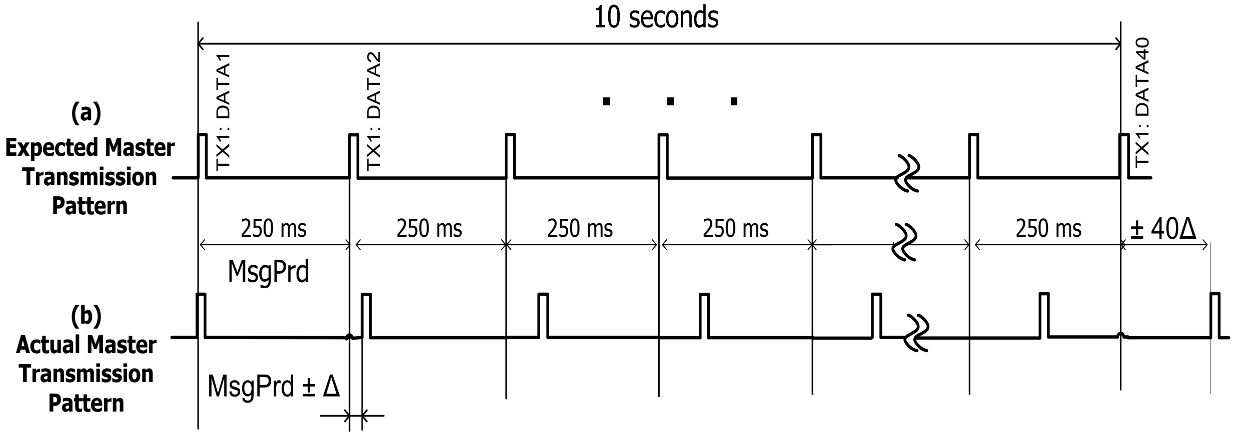

Figure 2. Test Duration Difference Caused by Clock Drift

3.4 Clock Drift Calculations

The following example illustrates the method for calculating the clock drift (in ppm) from the difference value measured in section 3.3.

3.4.1 Clock Drift Example Calculation

Assume a hypothetical situation in which the UUT transmits broadcast data for a duration of 10 seconds, and the channel period is set to 4Hz. Assuming a crystal frequency of 32.768 kHz, the expected number of ticks (T_e) in the 10 second interval is 327680:

$$Tick\ time\ period = \frac{1}{32768}\ seconds;$$

$$Total\ number\ of\ Ticks\ in\ the\ 10\ second\ interval = \frac{10\ seconds}{\frac{1}{32768} seconds} = 327680$$

For this example, assume that the measured elapsed time (T_m) calculated as discussed in section 3.3 is 327711 ticks (i.e. ≠327680), indicating that clock drift exists. The clock drift can be obtained from these values using the following equation:

$$Clock\ Drift\ (ppm) = \frac{T_m - T_e}{T_e} \times 10^6$$

Hence, in this case:

$$Clock\ Drift = \frac{327711 - 327680}{327680} \times 10^6 \approx 95\ ppm$$

A positive calculated result implies that the UUT's clock drifts towards left of the center frequency, i.e. the clock is slower than the claimed frequency. Similarly, a negative result shows that the UUT's clock drifts towards right of the center frequency, i.e., the clock is faster than the claimed frequency. Hence, in this example, the clock is slower than the claimed crystal frequency by 95 ppm. Since this value is greater than the maximum recommended clock drift value of ±50 ppm, the use of this device for ANT would not be recommended.

4 Example PC Application: Clock Drift Calculator

4.1 Getting Started

An example PC application (Clock Drift Calculator) is included with this application note. This application serves as the slave host application that calculates the clock drift using the method discussed in section 3. Microsoft Visual C# 2008 Express Edition (or newer) is required for compiling and using the application. Figure 3 shows the application UI when it is launched. The master should be configured in transmit only mode and controlled by its own host application (e.g. ANTware II).

Prior to beginning the clock drift test, the application needs to be configured through the user interface. To find and connect the slave ANT USB module to the application, press the 'Find USB' button at the top after choosing the desired USB port.

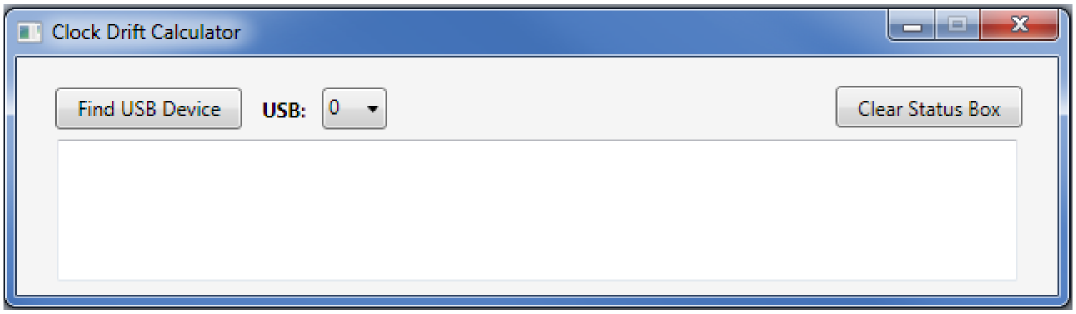

Figure 3. The Initial User Interface

If the ANT module is not connected to the appropriate USB port, or a module that does not possess the timestamp capability is found, or if the module is connected to another host application, a popup window will appear with an error message as shown in Figure 4.

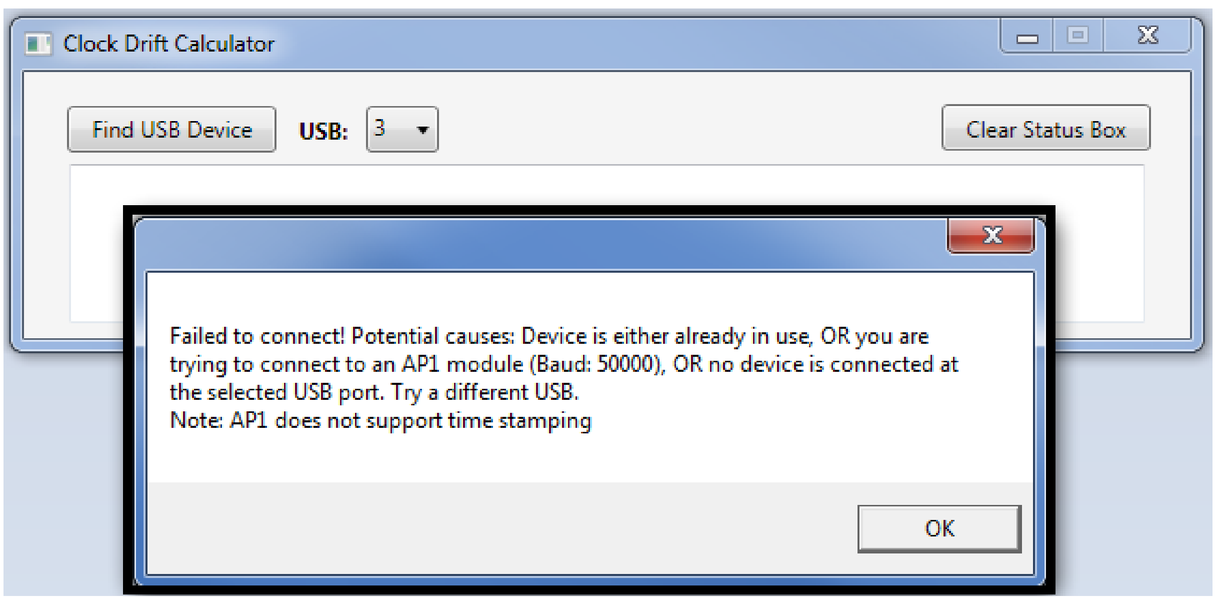

Figure 4. The USB Connection Error Message

4.2 Verify the Capabilities

If the ANT module is successfully connected to the application, the status box at the top of the window will display the device name and its capabilities (Figure 5). Verify the device name and capabilities to ensure that the correct ANT device is connected before continuing.

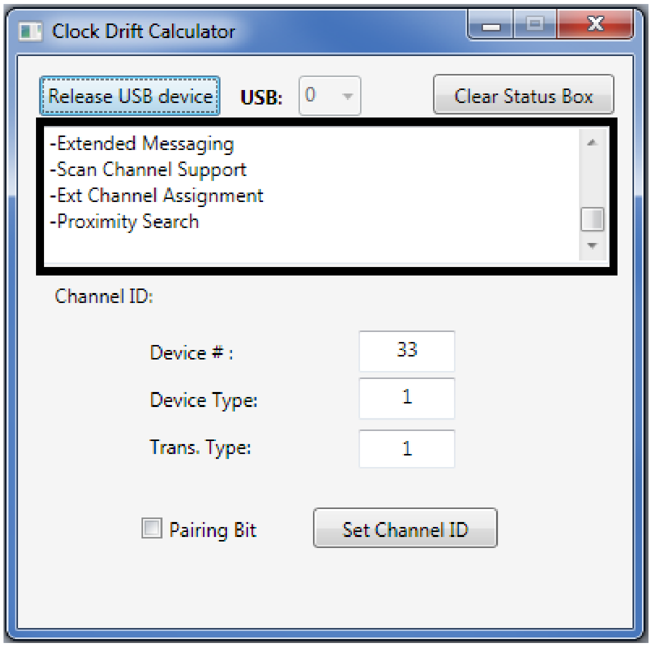

Figure 5. Correct USB Connection

4.3 Channel ID and Network Properties Setup

The 'Channel ID' pane (Figure 6) allows the user to configure the channel ID appropriately. It appears once the USB device is successfully recognized by the application. Ensure that the channel ID matches that chosen for the UUT, or that wildcard values are used. Click on 'Set Channel ID' to continue.

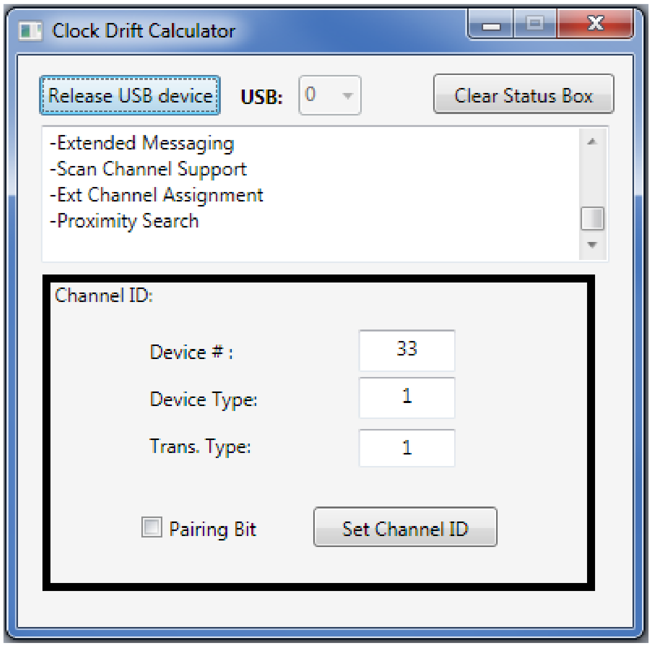

Figure 6. Channel ID Setup

The 'Network Properties' pane allows the configuration of the network key, radio frequency, and channel message period (Figure 7). These should be set to match the UUT. If the values are changed from the default ones, make sure that the corresponding 'Set' buttons are also pressed. Note that working at a higher channel period (in Hz) will result in a more frequent timestamp sampling rate, leading to statistically better results. Click on 'Back' to go back to the 'Channel ID' pane; click on 'Next' to continue.

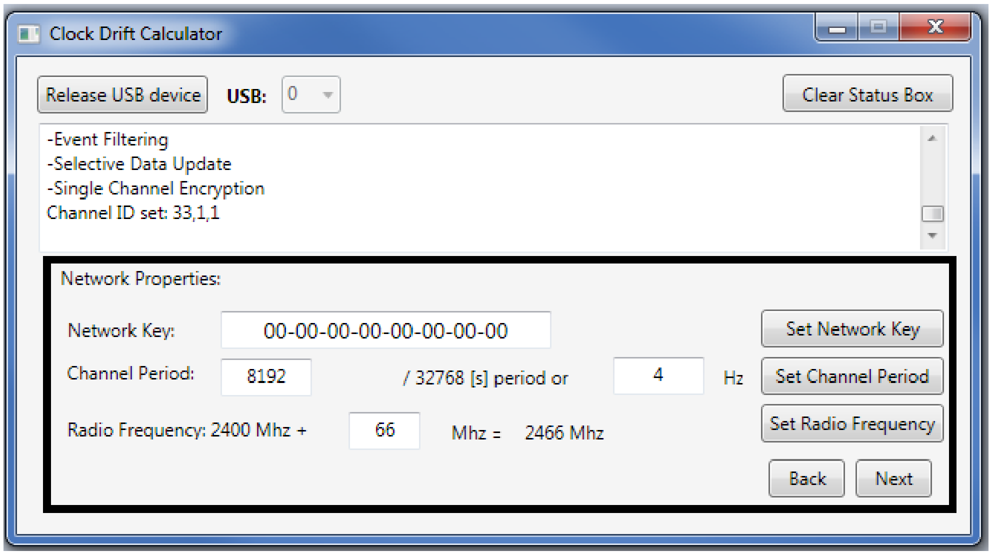

Figure 7. Network Properties Setup

4.4 Running the Test

The 'Test Properties and Results' pane (Figure 8) allows the user to configure the test duration and run the test. The 'Test Duration' textbox is used to specify the total test time in seconds. The random error will be minimized by increasing the test duration, because more timestamp values will be recorded. Generally, test duration of 10s should produce be sufficient.

To initiate the test, press the 'Begin Test' button. The instantaneous timestamp values and the calculated clock drift (in ppm) will be displayed. To go back and change any setup values, click on 'Back'.

The '0s Time Stamp' is the tick value obtained when the test was first commenced. The 'Number of Rollover Events' box tells the number of times the timestamp field rolled over. This, along with the '0s Time Stamp', can be used for manual verification of results. The 'RxFails' box on the user interface indicates the number of EVENT_RX_FAIL events that occurred during the test. The EVENT_RX_FAIL events are rejected from calculating the measured elapsed time, but the expected elapsed time value is increased as normal. See the source code for details.

Assuming that one of the devices is a golden unit, a calculated drift of less than ±50 ppm implies that the device can be used with the ANT protocol. If a golden unit is not used, the calculated drift should be taken as an indication of the total system drift only. For example, drifts of +100 ppm and -100 ppm on the two devices respectively would produce a net result of 0 ppm, even though neither of the devices is suitable for ANT. Repeated tests using different device pairings can be conducted to isolate the error attributable to each device.

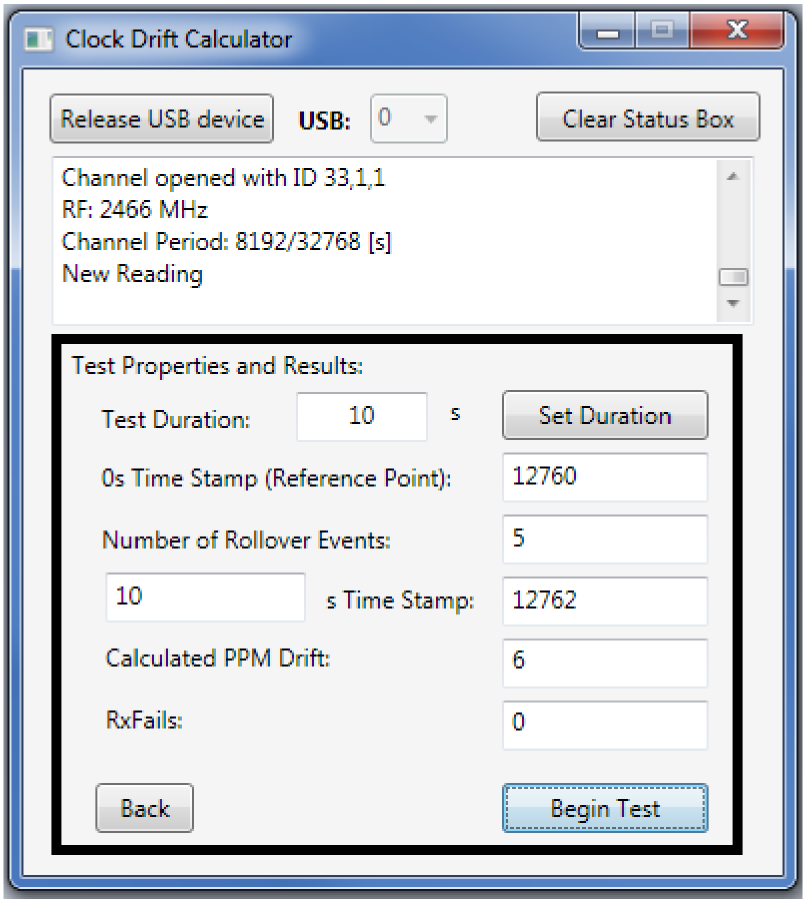

Figure 8. Running the Test

5 Closing Remarks

This application note describes a method to test for clock drift in an ANT master-slave device combination. A Clock Drift Calculator PC application, along with the accompanying source code, is also provided. If one of the two ANT devices used is known to be a golden unit, this method can be used to calculate the clock drift in a UUT.

For any further inquiries, please use the developer forum at www.thisisant.com.