ANT Frequency Agility

ABSTRACT

ANT frequency agility improves co existence between ANT devices and stationary interference sources (such as Wi-Fi). ANT frequency agility is built into some ANT devices (refer to datasheets for capabilities), and can be implemented at the application level for other ANT devices. For devices that have built-in frequency agility, specific instructions are described, as well as the key design considerations for implementing such a scheme at the application level. Algorithms for implementing both the master and slave side controllers are described, and source code for a PC-based controller application is provided.

TABLE OF CONTENTS

LIST OF FIGURES

1 Introduction

As wireless devices using the 2.4GHz radio band increase in popularity, the likelihood of encountering situations where ANT devices will be operating on the same frequency as other devices will also increase. In most cases, ANT communications will be largely unaffected due to the radio's extremely short "active" time required to transmit ANT data packets. However, in cases where other devices can monopolize certain radio frequencies, there will be a higher likelihood of interference. Enabling ANT frequency agility improves coexistence of ANT devices with these types of systems.

Similar to frequency hopping schemes, ANT frequency agility allows a channel to change its operating frequency; however, unlike hopping, operating frequencies are only changed when significant degradation is observed in the ANT channel's performance.

ANT frequency agility is included internally in some ANT devices (refer to data sheets for capabilities), or can be implemented at the application level.

Since IEEE 802.11 is typically the most common source of interference in the 2.4GHz band, the discussions in this document focus mainly on the avoidance of 802.11 transmissions.

2 Relevant Documents

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

- ANT Message Protocol and Usage
- ANT Development Kit User Manual

3 Bill of Materials

The following is a list of the hardware tools required for testing the example given in this application note:

- 1 x ANT Development Kit

4 Frequency Agility Design

In order to design an efficient frequency agility system, the following is required:

- A simple method of tracking the link performance,
- A method of synchronizing frequency transitions,
- A method of channel discovery,
- A set of RF channel frequencies that optimize the system.

4.1 Channel Discovery

One problem with switching frequencies is the increased time it takes to initially discover a device. The discovery or acquisition time increases with each additional frequency that the slave is required to search on prior to discovering the master. ANT frequency agility uses three different frequencies distributed over the usable 2.4 GHz ISM band. Due to the high transmit power and bandwidth of IEEE 802.11 devices it was found that further sub-dividing the ISM band with more frequencies did not result in better performance. Limiting the frequency agility to three configurable frequencies helps to reduce the slave's average acquisition time.

4.1.1 Slave

When a slave device configured in frequency agility mode opens a channel, it will perform a standard search on one of the three defined frequencies; however, on exceeding the specified search timeout, instead of closing the channel, the slave will switch to the next specified RF channel frequency and continue searching. Again, on exceeding the search timeout, the slave will change to the third specified channel frequency and continue searching. The slave will search twice on each of the three possible RF channel frequencies before closing its channel

4.1.2 Master

On opening a master channel configured in frequency agility mode, the master will transmit acknowledged messages on one of the three designated frequencies. If no slave responds within a set duration, the master will change to the second RF channel frequency. Again, if no slave responds, the master will change to the third RF channel frequency. The time it takes for a master device to rotate through the three RF channel frequencies is calculated from the search timeout as shown below:

Master Discovery Transmission Time = (Search timeout)/3

The master continues cycling through each of the three configured RF channel frequencies indefinitely, or until the application closes the channel.

4.1.3 Search Timeouts

The master and slave devices shall have identical search timeout values. This ensures the master and slave devices can find each other in the case of significant RF degradation.

4.2 Synchronizing Frequency Transitions

If a master and slave have established communication and the link performance is determined to be poor, ideally, the transition to a new frequency should occur simultaneously. Section 4.3 describes how the frequency transitions are based on link performance. As both sides are monitoring the link, this helps ensure that both sides of the channel shall transition at roughly the same time, but would not require that they do so at exactly the same time.

Both master and slave shall transition to the next frequency listed in a fixed table known by both sides (as described in Section 4.1). This method of synchronizing the transitions is simpler and more reliable than using methods which require additional messaging between the master and slave, especially when in poor RF conditions.

4.3 Tracking Link Performance

Tracking link performance is critical as the quality of a link serves as an indicator to switch frequencies. Master and slave devices monitor link performance slightly differently, as described in the following sections

4.3.1 Slave

The slave uses a simple method to detect a poor link. The ANT protocol specifies that after a certain number of consecutive missed messages, the slave shall drop into search. For a slave device configured in frequency agility mode, this search shall occur on the next RF channel frequency as specified in the frequency agility configuration list

The number of consecutive failures (C) before a slave drops into search is dependent on the channel period, as show below:

If T < 32768 (i.e. message rate is slower than 1.1 Hz)

$$C = 4$$

Else

$$C = \left\lfloor \frac{65\,535}{T} \right\rfloor + 1$$

Where T is the channel period, in 1/32768 seconds. Note that the above equation takes the floor of the division.

Some ANT devices support the EVENT_RX_FAIL_GO_TO_SEARCH message. For these devices, the number of EVENT_RX_FAIL messages received at the application level shall be one less than the number calculated for 'C' in the above formula. For example, a slave device at 4 Hz (T = 8192) will have a C value of 8. However, the number of EVENT_RX_FAIL messages received by the application will be 7 if the device supports the EVENT_RX_FAIL_GO_TO_SEARCH message.

4.3.2 Master

The master tracks link performance using acknowledged messages. This allows the master to determine if a slave is successfully receiving messages. The master can then monitor the link performance keeping a count of consecutive failed messages (i.e. EVENT_TRANSFER_TX_FAILED). This is accomplished using a counter that is incremented on each consecutive failed message. Anytime a message has been successfully received (i.e. EVENT_TRANSFER_TX_COMPLETED), the counter is reset to 0. Once the counter equals a specific number of missed messages, (calculated based on the channel period and search timeout), the link is judged to be poor and the device shall switch to a different RF channel frequency.

The number of consecutive failed messages (C) before a master switches to a different RF channel frequency is calculated as shown below:

$$C = \frac{S}{3} \times \frac{32\,768}{T}$$

Where T is the channel period, in 1/32768 seconds, and S is the search timeout, in seconds.

Again, a successful message resets the counter to zero, and the process starts again.

In general, the number of failed messages detected by the master will be higher than the number of missed messages seen by the slave. This is due to the master requiring two RF transmissions(data message from master to slave, and acknowledgement from slave to master) in order to receive a successful message (i.e. EVENT_TRANSFER_TX_COMPLETED).

4.4 Choosing RF Frequencies

The primary purpose of ANT frequency agility is to avoid interference from stationary interference sources such as 802.11. In General, 802.11 routers are configured to operate on channels 1, 6 or 11. These channels are 22MHz wide and have center frequencies of 2.412GHz, 2.437GHz and 2.462, respectively (Figure 1).

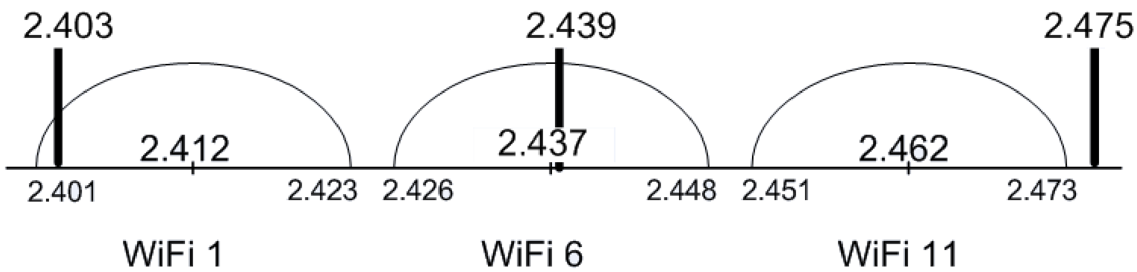

Figure 1. Common Wi-Fi Channels and ANT Frequency Agility Default Channels

ANT devices may experience high levels of interference if they are operating on frequencies that are within, or near, an active 802.11 channel. This is due to the high transmit power and bandwidth of most 802.11 devices while they are actively transferring data. For an ANT device to coexist with an 802.11 signal, it has to operate on a frequency distal to the 802.11 channel.

The ANT frequency channels selected for this application are 3, 39, and 75. These correspond to 2.403GHz, 2.439GHz and 2.475GHz, respectively. These frequencies were selected as default values as each one is sufficiently far away from the two of the common three 802.11 channels (Figure 1).

5 ANT Frequency Agility

For those ANT devices that have internal frequency agility, it is a simple matter of enabling frequency agility when assigning the channel. Operating frequencies may be left at the default values of 3, 39 and 75, or set as desired.

Figure 2 outlines the steps to configure and establish a frequency agility channel. Boxes with dashed outlines are optional steps; if not specifically set, default values for these parameters will be used.

Boxes shaded in grey refer to parameters that are required to establish any ANT channel. White boxes relate to parameters that should be considered when using an ANT frequency agility channel. First, use the *Assign Channel* (0x42) command and assign a bi-directional channel type, and set the extended assignment byte to 0x04 to enable frequency agility. This must be done on both master and slave devices.

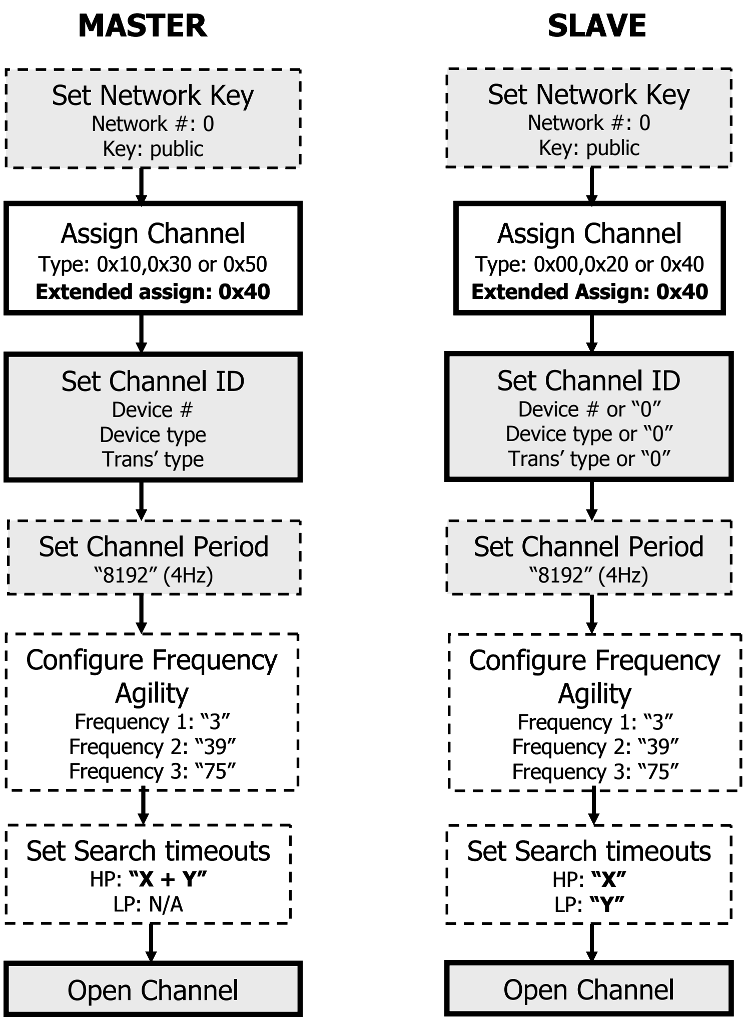

Figure 2. Establishing a Frequency Agility Channel

If frequencies other than the default are desired, use the *Configure Frequency Agility* (0x70) command to designate the three agility frequencies. Make sure these are set to the same values, in the same order, on both master and slave.

Search timeout values should be considered carefully, as they influence the amount of time the devices will take before changing frequencies. It is important that these values match on both master and slave. **Take care to ensure that the master's high priority search timeout equals the sum of the slave's high priority, and low priority search timeouts** (if applicable to the device you are using).

Configure the remainder of the channel (channel ID, channel period etc) as desired, ensuring the master and slave message periods also match, then open both master and slave channels to establish communication. The internal ANT frequency agility will perform as described in section **Error! Reference source not found.**.

Note: frequency agility should only be used in a point to point communication use case. As the master relies on acknowledged messages to monitor the link, multiple slave devices will create a race condition and result in undesired behavior.

6 Application Level Implementation Example

For ANT devices that do not internally include frequency agility (refer to data sheets for capabilities), the attached source files detail the implementation of a PC-based application that can be configured as either master or slave of the frequency agility channel.

Two USB interface boards and two ANT modules are required in this example application. Both modules can be connected to the same or separate computer(s).

Open the application and select the USB port number that the master device is connected to. Click the 'Connect Device' button. Follow the same procedure for the slave device (i.e. under the 'Slave' tab).The application automatically detects the type of ANT device used, and displays the device's capabilities.

Figure 3 illustrates how to connect the ANT devices.

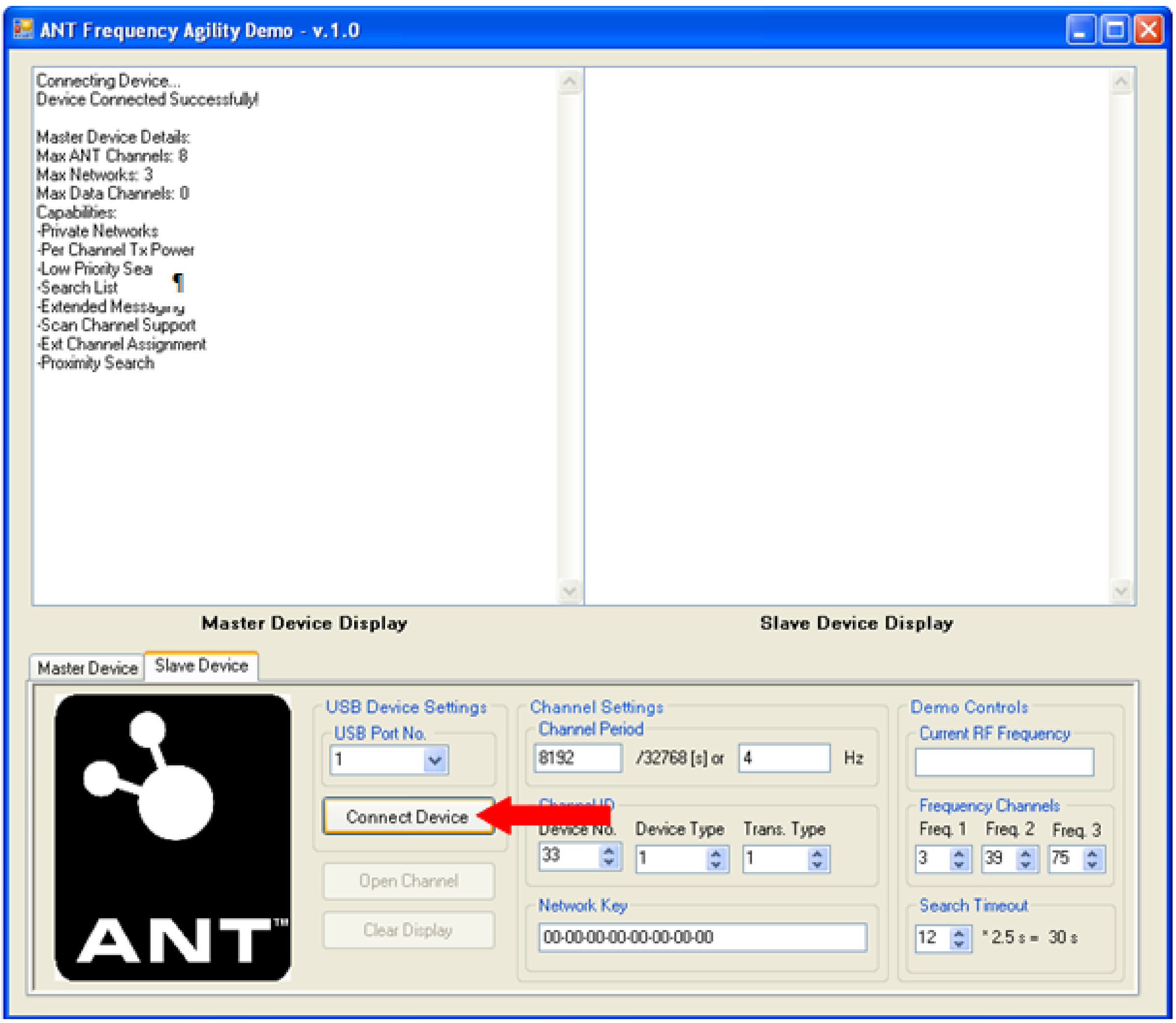

Figure 3. Connecting the ANT Modules

Ensure that the channel period, search timeout, and network key parameters are the same for both devices.

The default channel period is set to 8192/32768 seconds (4 Hz). The network key is set to the default public network key. The network key must be entered as a series of hexadecimal bytes. The default search timeout is set to 30 seconds on both master and slave. This application uses only high priority search timeout for both the master and slave devices. Figure 4 shows the location of the channel period, network key, and search timeout controls in the application window.

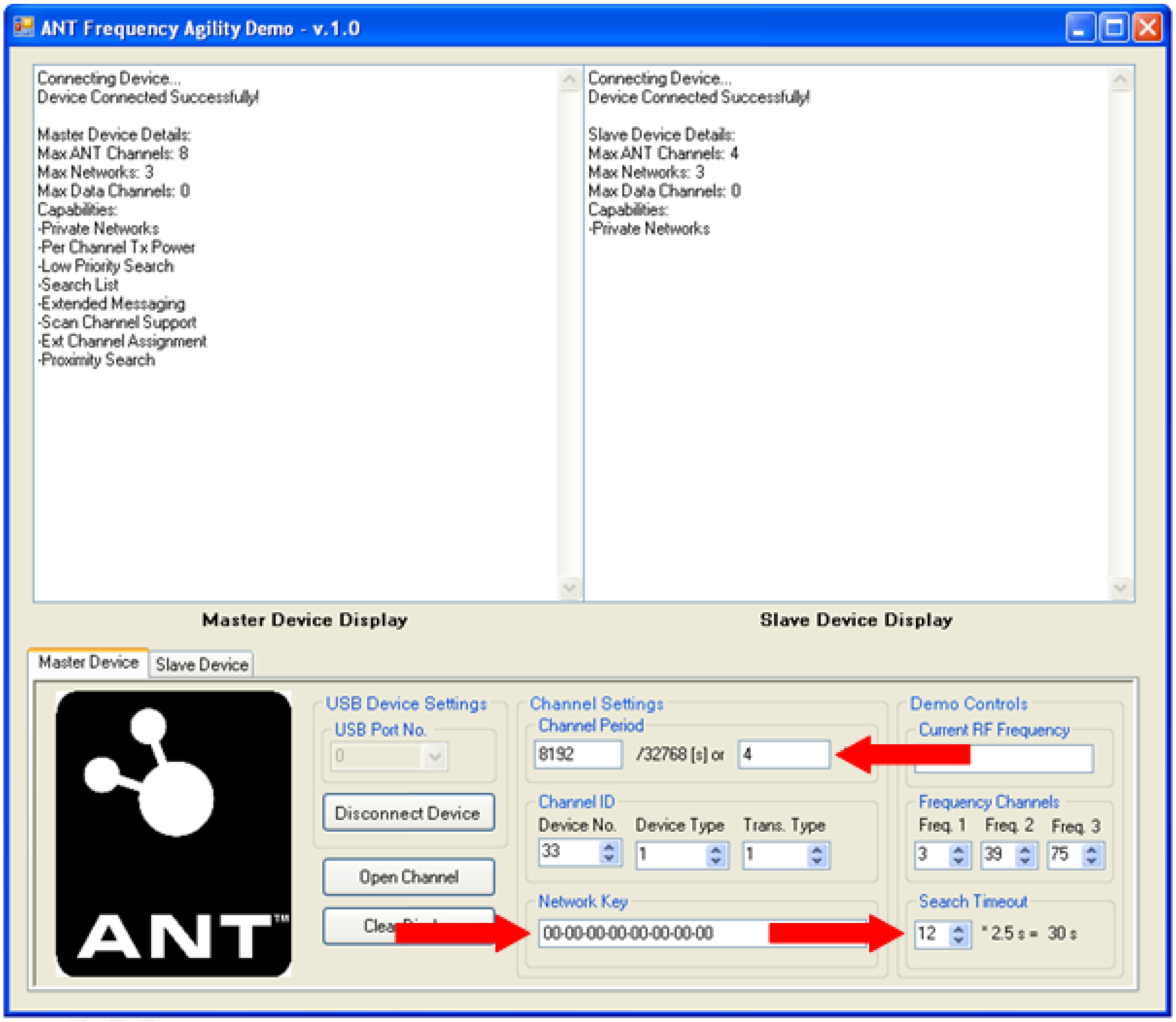

Figure 4. Channel Period, Network Key, and Search Timeout Controls

The slave's channel ID (i.e. device number, device type, and transmission type) should be set to match the master device, or use 'wildcard' values (0) to find any master device transmitting on the desired RF space.

The channel period should be selected to suit the specific application's desired data throughput, latency and power consumption requirements. The channel period impacts the performance of the frequency agility channel in that the faster the master channel period, the faster the slave can acquire it. This improved performance comes at the cost of increased power consumption.

Additionally, the search timeout impacts the performance of the frequency agility channel. If the search timeout is long, there is a potentially long acquisition period in the case of discovering a device. A fast search timeout may not provide sufficient time for the devices to acquire one another. Note that this application makes use of high priority search timeouts only (i.e. low priority search timeout set to zero).

The three RF frequencies configured for the frequency agility channel can be set as desired (Figure 5).

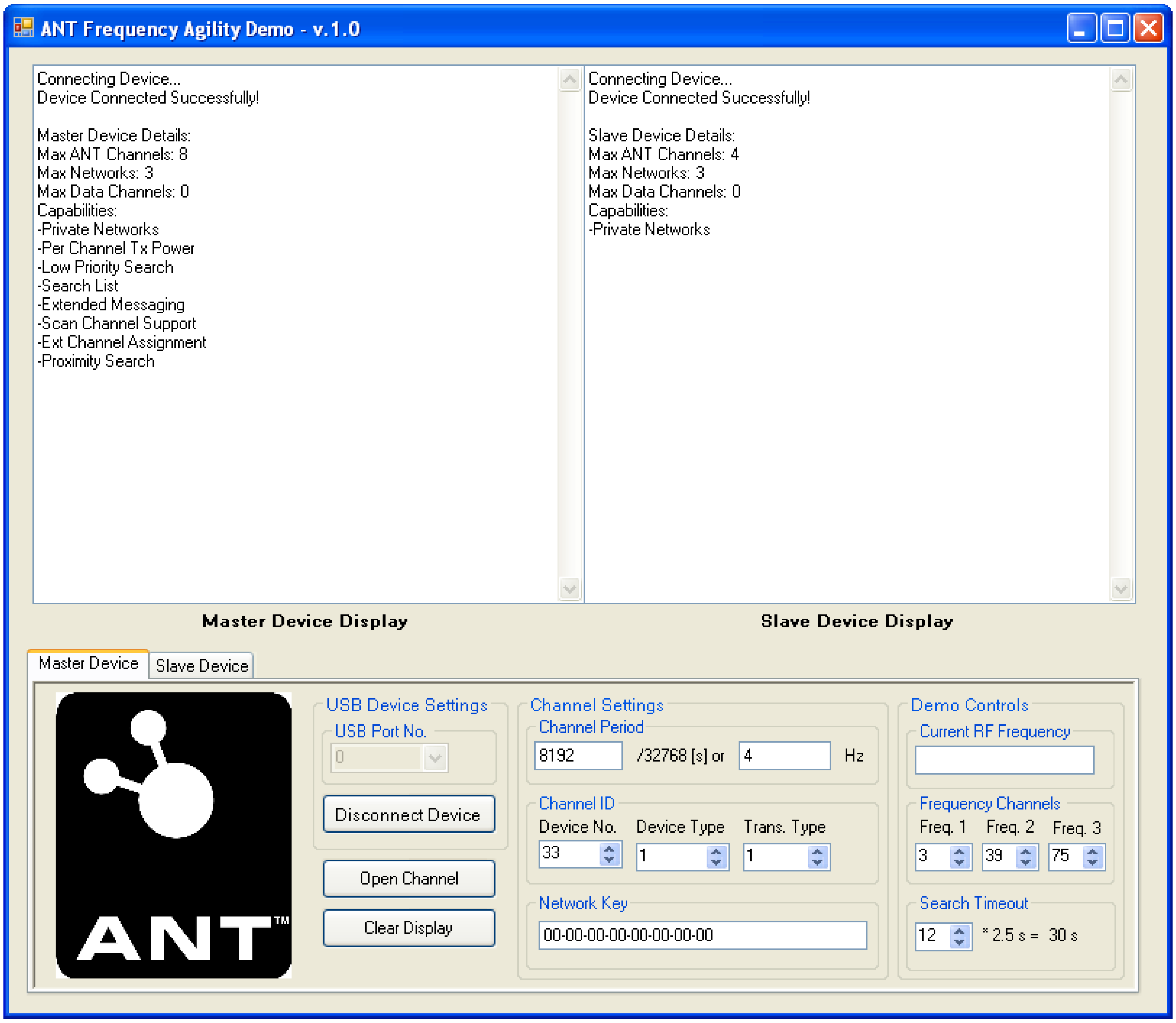

Figure 5. Setting RF Frequencies

To start the communication, click the 'Open Channel' button on both devices. The master device starts by sending acknowledge messages on the RF channel frequency set in "Freq 1". The slave device starts searching on RF channel frequency "Freq. 1". Figure 6 illustrates the application window on opening the channels.

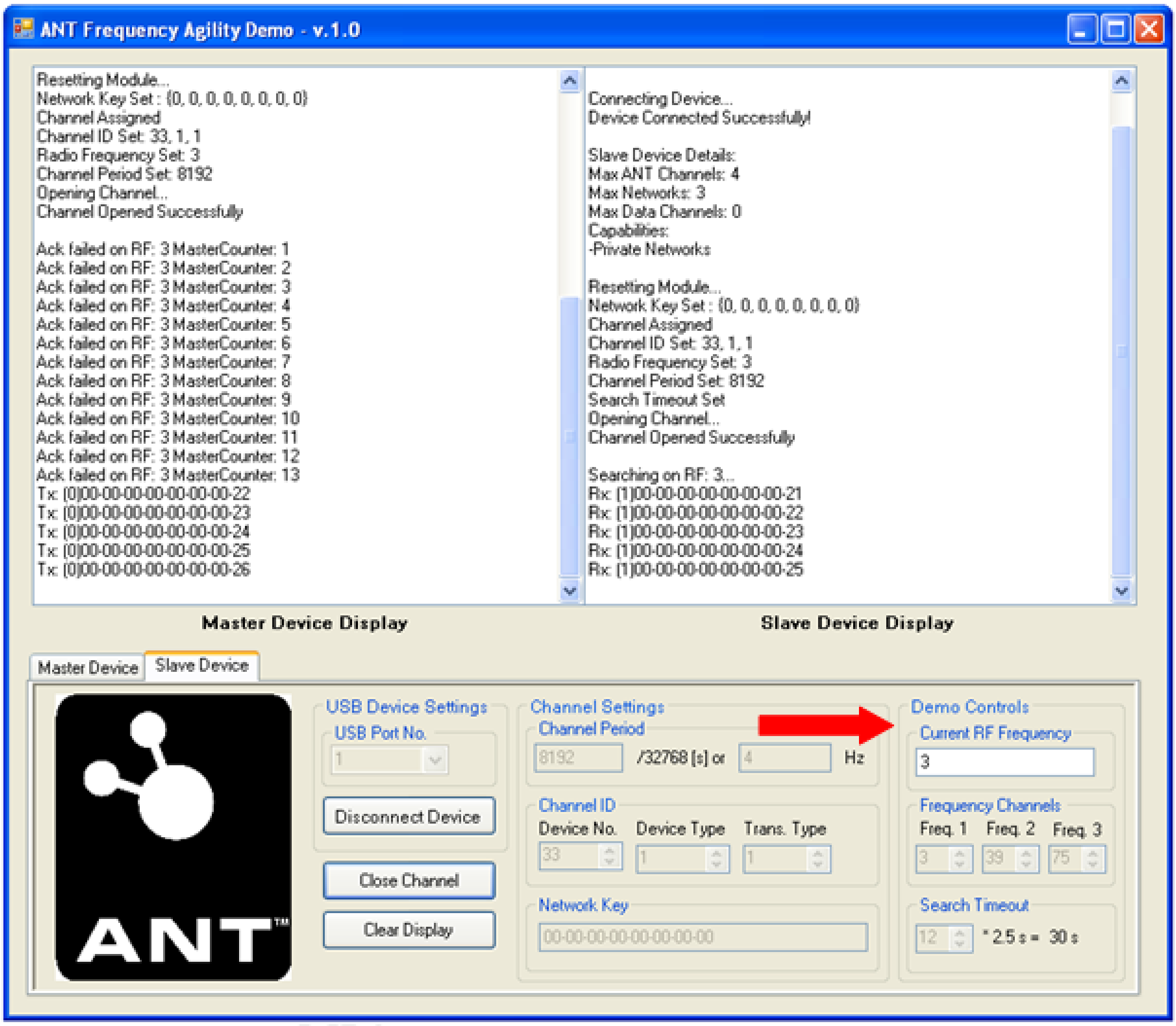

Figure 6. Master and Slave Behavior When the Channel is First Opened

To observe the frequency agility behavior of the master, click the 'Close Channel' button under the slave device tab. In the "Master Device Display" the MasterCounter value will increment with each unsuccessful message, and eventually the master changes RF channel frequencies.

The current RF channel frequency is displayed in the "Current RF Frequency" box under "Demo Controls".

Figure 7 illustrates a master device transmitting acknowledged messages, without the presence of a slave. The MasterCounter increments to the maximum value determined by the channel period and search timeout, then the master changes RF channel frequencies.
The process repeats until a slave responds with an acknowledgement, or until the channel is closed.

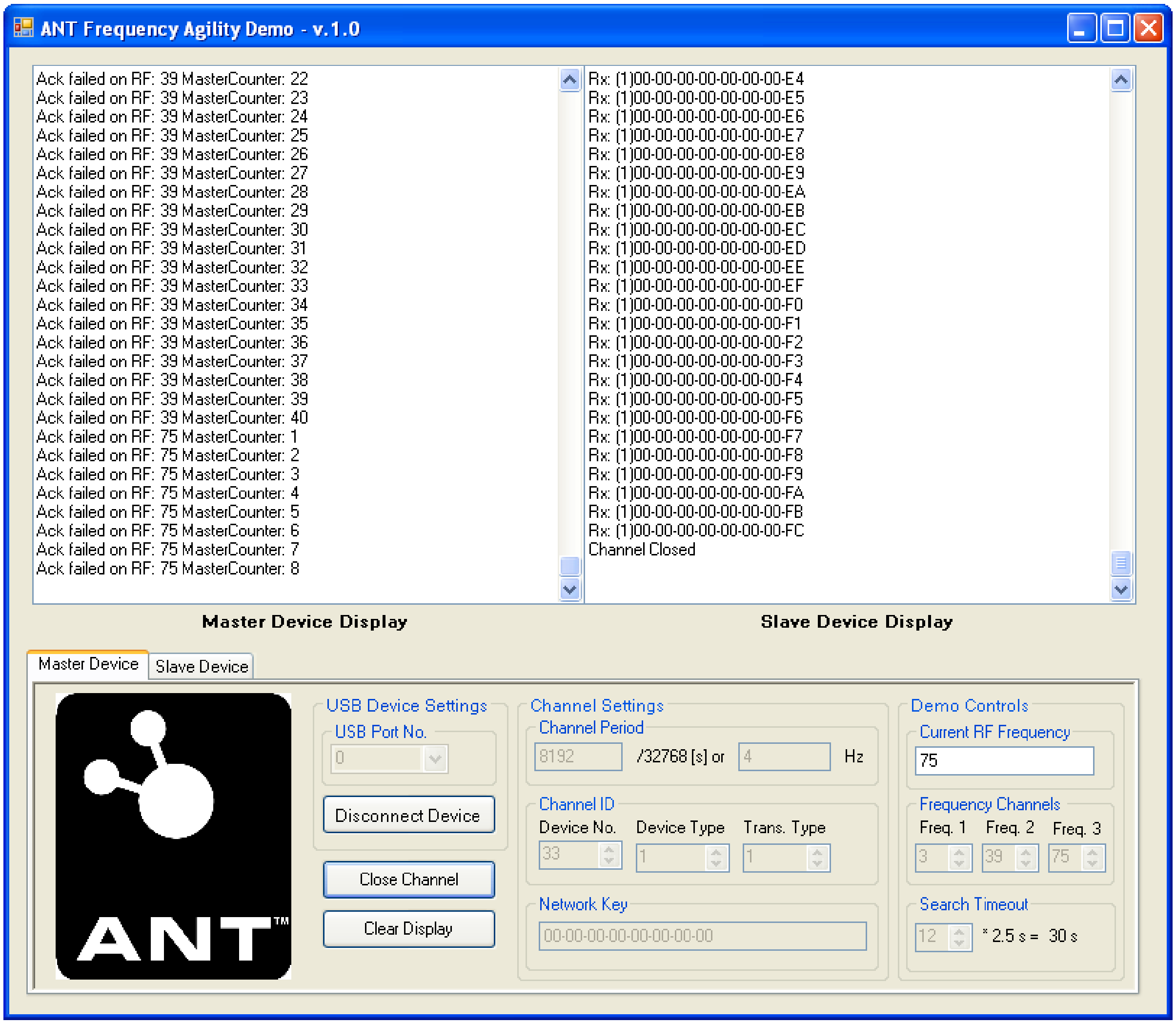

Figure 7. Master Displaying Frequency Agility Configuration

Re-opening the slave channel demonstrates how the slave searches over the three frequencies and acquires the master.

The slave device's frequency agility behavior can similarly be observed by closing the master channel once there is an established connection between master and slave. The slave drops into a search after receiving the maximum number of EVENT_RX_FAIL's (as calculated in Section 4.3). After the search duration exceeds the search timeout, the slave then changes the RF channel frequency and searches again. If an appropriate master is not found, the slave will continue to change frequencies and search as described in Section 4.1. This process is repeated a maximum of 6 times (i.e. twice in each frequency) and is illustrated in Figure 8 below.

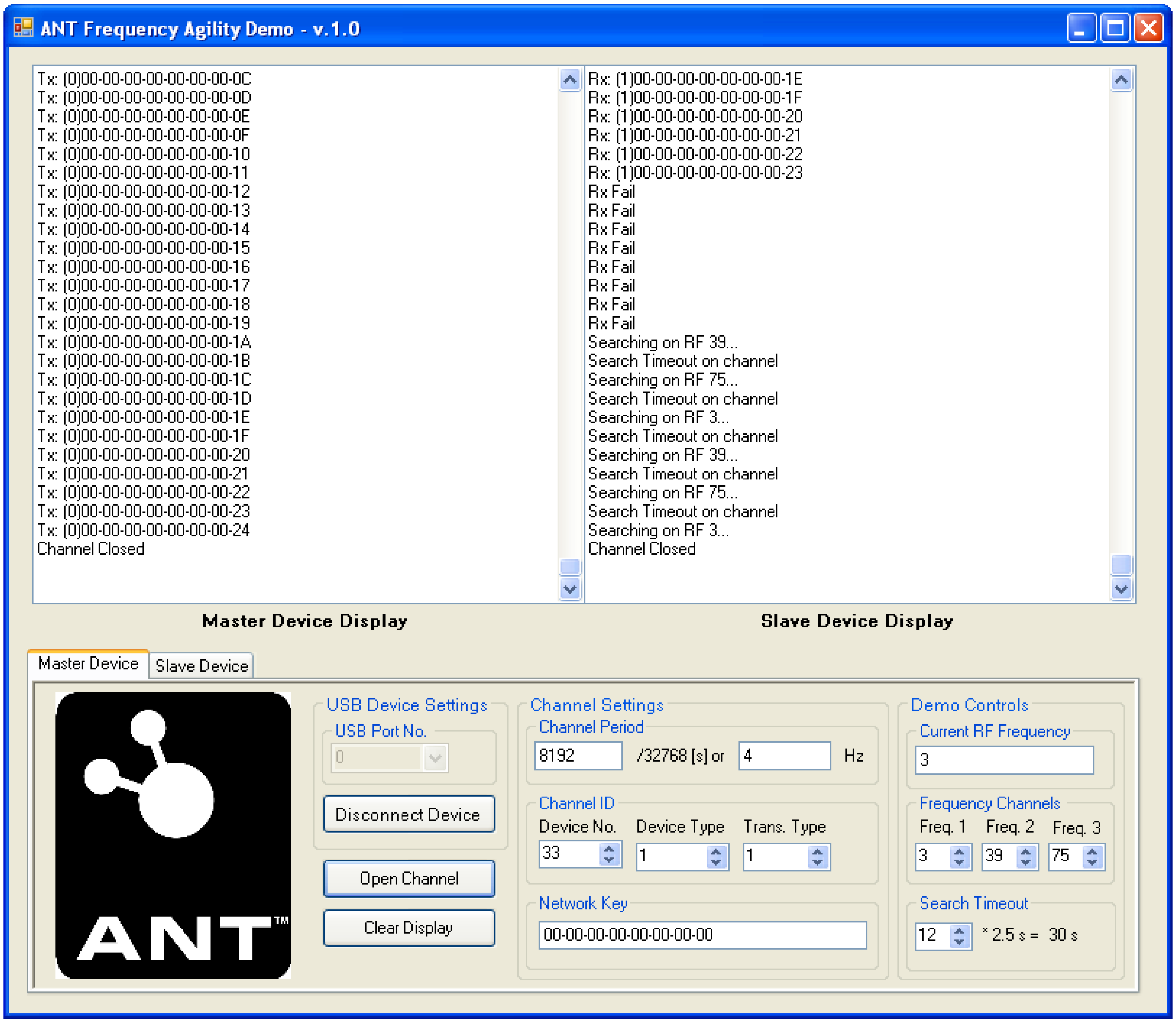

Figure 8. Slave Displaying Frequency Agility Configuration

Note that, in this example, the slave device is an AP1 module. At 4 Hz, eight EVENT_RX_FAIL's are received before the slave drops back into search. For devices that support the EVENT_RX_FAIL_GO_TO_SEARCH message, only 7 EVENT_RX_FAIL messages would be displayed (at 4 Hz) followed by the EVENT_RX_FAIL_GO_TO_SEARCH.

7 Closing Remarks

This application is aimed at providing a detailed description of how ANT frequency agility works, and how to establish a frequency agility channel whether it is included internally in the ANT device, or implemented at the application level. Note that frequency agility is meant for a point to point case, with a single master and slave. Frequency agility is not meant for use in multiple slave networks.

The primary goal of the system described is to allow ANT devices to co-exist in the same environment with very active 802.11 networks. The example source files show how to implement a host controller application for a frequency agility channel.