Over the Air Firmware Updates Using ANT-FS

ABSTRACT

This application note describes a communication mechanism based on ANT-FS for updating an Application image, Bootloader and/or Wireless Protocol Stack on an ANT enabled device.

application, use or sale of your designs and/or products that use ANT, ANT+, ANT+ Documents, ANT+ Design Tools, or any other products or services supplied under this Agreement.

If you are not an ANT+ Adopter, please visit our website at www.thisisant.com to become an ANT+ Adopter. Otherwise you must destroy this document immediately and have no right to use this document or any information included in this document.

The information contained in this document is subject to change without notice and should not be construed as a commitment by Dynastream Innovations Inc.

Products sold by DYNASTREAM are not designed for use in life support and/or safety equipment where malfunction of the Product can reasonably be expected to result in injury or death. You use or sell such products for use in life support and/or safety applications at your own risk and agree to defend, indemnify and hold harmless DYNASTREAM from any and all damages, claims, suits or expense resulting from such use.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

ANT-FS can be used to transfer application, bootloader and/ or wireless protocol stack images from an update host (e.g. PC, mobile, tablet) to an update target. The update target is typically an embedded device capable of receiving and activating images received.

This document is focused on the transport mechanism to transfer firmware images between devices; implementation specific details for particular platforms are out of scope. For implementation specific details, refer to the ANT DFU/Bootloader reference design and accompanying documentation.

2 Relevant Documents

Refer to current versions of the listed documents. To ensure you are using the current versions, check the ANT+ website at www.thisisant.com

- ANT Message Protocol and Usage
- ANT-FS Technical Specification
- ANT AN Device Pairing

3 Update Process Overview

The OTA firmware update mechanism is based on ANT-FS for transferring firmware images to the update target device. The “ANT-FS Technical Specification” must be implemented in its entirety to support OTA firmware updates. This section provides an overview on the update process as defined by ANT-FS.

Figure 1 provides an overview of the update process. The update target acts as an ANT-FS client, and the update host as an ANT-FS host.

The update target transmits the ANT-FS link beacon by default. The update target broadcasts the ANT+ managed manufacturer ID in the ANT-FS Link Beacon. The current list of manufacturer ID values can be found in the FIT.xls profile (available within the FIT SDK at www.thisisant.com). New manufacturers are required to be members of the ANT+ Alliance in order to be added to this list; please contact the ANT+ Alliance at antalliance@thisisant.com for details. The value 255 (0x00FF) has been reserved as a development ID and may be used by manufacturers that have not yet been assigned a value

The device type in the ANT-FS Link Beacon is used to reflect the manufacturer specific model number of the update target. Additional device identification can be provided through the ANT-FS friendly name, which can be used as a device descriptor string. The OTA Update Information File, described in Section 3.2, can also be used to provide additional information about the device. Care must be taken to ensure that the update target is the intended device before proceeding with a firmware update; it is strongly recommended that all device identifying information (manufacturer ID, model number, hardware version, etc) is checked to determine that the firmware image(s) to be sent to the device are intended for that particular device. Proper pairing techniques are strongly encouraged; for more information, refer to the document “ANT AN Device Pairing”.

Once the update host receives the ANT-FS link beacon, both devices progress through authentication and reach transport state. The host can optionally download the directory and see if an OTA Update Information file is present; if present, this file can provide additional information to identify the update target, as well as information regarding the current firmware versions present on it. The update host may disconnect from the ANT-FS session after retrieving the OTA Update Information file, if no firmware updates are to be performed (e.g., device is already running the latest firmware version).

If the host intends to perform a firmware update, it will proceed to upload the new image. After the upload is completed successfully, the host sends the disconnect command. The update target will end the ANT-FS session and validate and activate the new image.

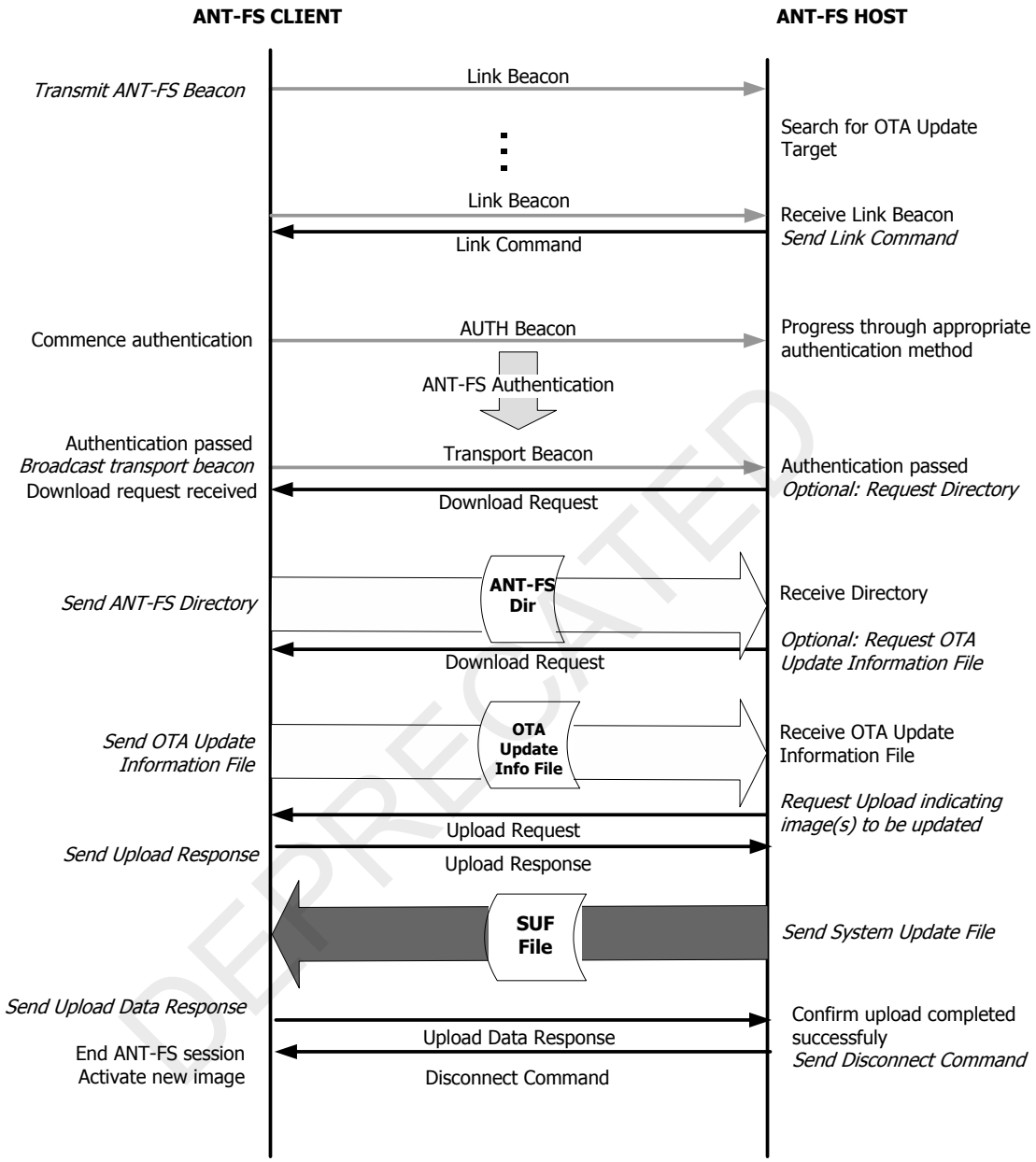

Figure 1. Update Process Overview

3.1 Channel Configuration

The update host expected to communicate with an update target configures an ANT channel with its channel parameters set as listed in Table 1. Please see the ANT Message Protocol and Usage document for more details on these.

Table 1. ANT Channel Configuration for an OTA Update Host

Parameter	Value	Comment
Channel Type	Slave (0x00)	The OTA update target is a master device; therefore, the update host must be configured as the slave. Bidirectional communication is required.
Network Key	ANT-FS Managed Network Key	The ANT-FS Managed Network Key is governed by the ANT+ Managed Network licensing agreement.
RF Channel Frequency	50 (0x32)	RF Channel 50 (2450MHz) is used.
Transmission Type	0 for pairing	The transmission type must be set to 0 for a wildcard search. Once the transmission type is learned, the receiving device should remember the type for future searches.
Device Type	16 (0x10)	Device Type corresponds to OTA Firmware Updates over ANT-FS.
Device Number	0 for pairing	Set the device number parameter to 0 for a wildcard search. Once the device number is learned, the receiving device should remember the number for future searches.
Channel Period	8192 (4Hz)	Channel period of 4Hz is used.

The OTA update target shall establish its ANT channel as shown in Table 2.

Table 2. ANT Channel Configuration for an OTA Update Target

Parameter	Value	Comment
Channel Type	Master (0x10)	Bidirectional master
Network Key	ANT-FS Managed Network Key	The ANT-FS Managed Network Key is governed by the ANT+ Managed Network licensing agreement.
RF Channel Frequency	50 (0x32)	RF Channel 50 (2450MHz) is used.
Transmission Type	Set MSN to 0 (0x0) or MSN of extended device number. Set LSN to 5 (0x5)	The most significant nibble of the transmission type may optionally be used to extend the device number from 16 bits to 20 bits. In this case, the most significant nibble of the transmission type becomes the most significant nibble of the 20 bit device number.
Device Type	16 (0x10)	Device Type corresponds to OTA Firmware Updates over ANT-FS.
Device Number	1-65535	This is a two byte field that allows for unique identification of an update target. It is imperative that the implementation allow for a unique device number to be assigned to a given device. NOTE: The device number shall not be 0x0000.
Channel Period	8192 (4Hz)	Channel period of 4Hz is used.

The device number of the OTA update target needs to be as unique as possible across production units. An example of achieving this specification is to use the lowest two bytes of the serial number of the device for the device number of the ANT channel ID; ensure that the device has a set serial number. The next significant nibble (ie., bits 16-19) of the serial number can be used in the MSN of the transmission type.

The device number of the update target shall not be 0x0000. Care should be taken if the device number is derived from the lower 16-bits of a larger serial number. In this case, ensure that serial numbers that are multiples of 0x10000 (65536) are handled correctly such that the device number is not set to 0.

3.2 Querying Current Version Information

The OTA Update Information file provides information about the OTA update target, including details about the software versions currently running on the device. This is an optional file, and if available, is listed in the ANT-FS directory with file type 0x0E, and file identifier 0x001.

The format of the OTA Update Information File is described in Table 3. All multi-byte numeric fields use little endian encoding.

Table 3. OTA Update Information File Format

Parameter	Description	Size (bytes)
File Structure Version	Indicates the version of the OTA Update Information file format. The upper nibble represents a major revision, while the lower nibble represents a minor revision. Set to 0x10 (v1.0)	1
File Structure Version	Indicates the version of the OTA Update Information file format. The upper nibble represents a major revision, while the lower nibble represents a minor revision. Set to 0x10 (v1.0)	1
Hardware Version	Indicates the hardware version of the update target. Managed by the manufacturer.	1
Region/Product Identifier	Manufacturer specific identifier, used to provide additional information to the update host on which image to send the device. For example, this field can be used to identify that a device requires a version of application firmware for a specific region.	1
Free Upload Space	Amount of free space available for buffering uploaded images, i.e., the maximum size of the images included in an SUF file (excluding header, version descriptor and CRC) that can be uploaded to the device. If the combined size of the images in a SUF file uploaded to the device exceeds this size, the update will fail.	4
Wireless Protocol Stack Version Identifier	Numeric version identifier for the Wireless Protocol Stack.	4
Wireless Protocol Stack Version String Length	Length of the Wireless Protocol Stack version string (max. 255)	1
Wireless Protocol Stack Version String	UTF-8 encoded version string, formatted for display.	0-255
Bootloader Version Identifier	Numeric version identifier for the Bootloader.	4
Bootloader Version String Length	Length of the Bootloader version string (max. 255)	1
Bootloader Version String	UTF-8 encoded version string, formatted for display	0-255
Application Version Identifier	Numeric version identifier for the Application, suitable for numeric comparisons. Format is managed by the manufacturer.	4
Application Version String Length	Length of the Application version string (max. 255)	1
Application Version String	UTF-8 encoded version string, formatted for display	0-255

3.3 Performing an OTA Update

To initiate an OTA update, the host will request an upload to the file index corresponding to the image(s) that it intends to update. These predefined indexes are specific to the OTA firmware updates ANT-FS mechanism, and do not need to be listed in the ANT-FS directory. The specific indexes supported may vary between implementations; an upload request response indicating the index is not valid is used to inform a host that updating that particular image(s) is not supported.

Table 4. Image Update File Indexes

File Index	Image
0xFB01	Application
0xFB02	Bootloader
0xFB03	Wireless Protocol Stack
0xFB06	Bootloader + Wireless Protocol Stack

The image transmitted to the update target follows a custom file format, SUF (System Update File), with a header and optional version descriptor block that allows encoding multiple images in a single file, as well as storing version information about the included images in the file itself to help identify the file at the update host. The file format for SUF files is described in Table 5, Table 6, and Table 7. All multi-byte numeric fields use little endian encoding.

Note that while the file structure supports transmitting the Wireless Protocol Stack, Bootloader and Application images in a single file, support for combined images is dependent on the implementation.

Table 5. SUF File Format

Block	Description	Size (bytes)
Header	Header that identifies this file as compatible with OTA firmware updates over ANT-FS, and specifies which images are included in this file. The format of the header is described in Table 6.	32
Wireless Protocol Stack Image	If this block is present, as specified in the header, it contains the Wireless Protocol Stack image, in binary format.	As specified in header.
Bootloader Image	If this block is present, as specified in the header, it contains the Bootloader image, in binary format.	As specified in header.
Application Image	If this block is present, as specified in the header, it contains the Application image, in binary format.	As specified in header.
CRC	Two byte CRC-16, in little endian format, padded with two leading zeroes. The CRC calculation includes the header, image blocks and the two leading padding zeros in this field. The CRC calculation excludes the version descriptor block.	4
Version Descriptor	If this block is present, as specified in the header, it includes version information about the image(s) included in this file. This field is intended for local identification of the image(s) present in a SUF file on the OTA update host, and shall not be transmitted over the air to the OTA update client. The format of this block is described in Table 7.	As specified in header.

The CRC is computed as follows:

```
UINT16 CRC_Get16(UINT16 crc, UINT8 byte)
{
   static const UINT16 crc_table[16] =
   {
      0x0000, 0xCC01, 0xD801, 0x1400, 0xF001, 0x3C00, 0x2800, 0xE401,
      0xA001, 0x6C00, 0x7800, 0xB401, 0x5000, 0x9C01, 0x8801, 0x4400
   };
   UINT16 tmp;
   // compute checksum of lower four bits of byte
```

```
    tmp = crc_table[crc & 0xF];
    crc = (crc >> 4) & 0x0FFF;
    crc = crc ^ tmp ^ crc_table[byte & 0xF];
    // now compute checksum of upper four bits of byte
    tmp = crc_table[crc & 0xF];
    crc = (crc >> 4) & 0x0FFF;
    crc = crc ^ tmp ^ crc_table[(byte >> 4) & 0xF];
    return crc;
}
```

Table 6. SUF File Header

Parameter	Description	Size (bytes)
Header Size	Size of the header structure. Set to 32.	1
File Structure Version	Indicates the version of the SUF file format. The upper nibble represents a major revision, while the lower nibble represents a minor revision. Set to 0x11 (v1.1)	1
Architecture	Set to 0x01	2
Identifier String	Set to ASCII ".SUF".	4
Reserved	Set to 0x00	10
Wireless Protocol Stack Image Size	Size of the Wireless Protocol Stack image, if included. Set to 0 if a Wireless Protocol Stack image is not present.	4
Bootloader Image Size	Size of the Bootloader image, if included. Set to 0 if a Bootloader image is not present.	4
Application Image Size	Size of the Application image, if included. Set to 0 if an Application image is not present.	4
Version Descriptor Block Size	Size of the version descriptor block (max. 780) Set to 0 if the version descriptor block is not present. This field is ignored by the OTA update target. This field is used by the OTA update host to locate and display the version information of the images included in the file to the user. This field is also used by the OTA update host to identify the number of bytes that should be removed from the end of a SUF file before uploading to the target device, to ensure that the version descriptor block is not transmitted to the OTA update client. This field shall not be modified by the OTA update host after removing an existing version descriptor block, since this would require recalculation of the CRC.	2

Table 7. SUF File Version Descriptor Block

Parameter	Description	Size (bytes)
Wireless Protocol Stack Version Identifier	Numeric version identifier for the Wireless Protocol Stack.	4
Wireless Protocol Stack Version String Length	Length of the Wireless Protocol Stack version string (max. 255)	1
Wireless Protocol Stack Version String	UTF-8 encoded version string, formatted for display.	0-255
Bootloader Version Identifier	Numeric version identifier for the Bootloader.	4
Bootloader Version String Length	Length of the Bootloader version string (max. 255)	1
Bootloader Version String	UTF-8 encoded version string, formatted for display	0-255
Application Version Identifier	Numeric version identifier for the Application, suitable for numeric comparisons. Format is managed by the manufacturer.	4
Application Version String Length	Length of the Application version string (max. 255)	1
Application Version String	UTF-8 encoded version string, formatted for display	0-255

The numeric version identifiers in the descriptor block are intended to allow for numeric comparisons between versions. Their encoding is manufacturer specific; version strings are recommended for display purposes. Choice between use of a numeric version identifier or a version string, or both, is left to the manufacturer.

The update host must ensure that the file index specified in the upload request corresponds to the contents of the SUF file uploaded to the device. That is, if the host requests an upload for file index 0xFB01, which corresponds to an application update, the SUF should contain an application image. If there is a mismatch in these, the update may fail.

4 Closing Remarks

This application note describes a mechanism using ANT-FS for transferring firmware images for updating Application, Bootloader and Wireless Protocol Stack on an ANT enabled device. This mechanism can be customized by manufacturers to support the features that are relevant for their use case and particular implementation.