Cresnet Connected Host Upgrade Protocol

Addendum to

CONNECTED-CHIP-C2N-DEMO

Cresnet® Connected Demo Chip

Reference Guide

5/11/2025

Contents

Scope

This document details an extension to the existing *serial bridge mode* protocol described in Cresnet Connected Demo Guide.

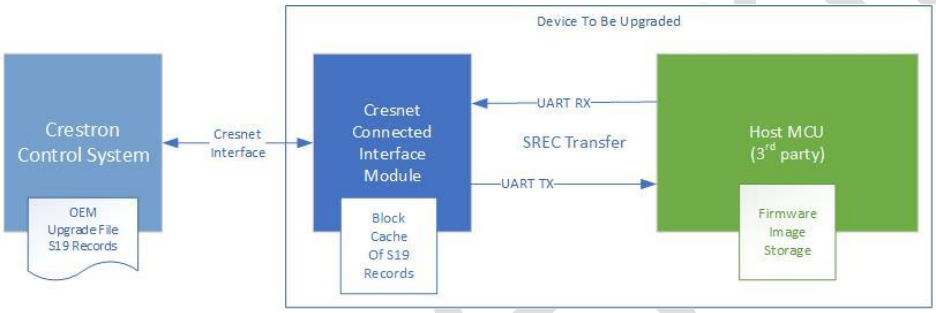
Purpose

This document describes the update to the CCM firmware that provides the ability to upgrade host processors using Crestron tools.

Glossary

CCM	Cresnet-connected-Module or Cresnet-Connected-IC
Host	OEM processor linked to CCM by UART.
HU	Host Upgrade

Introduction



As indicated in the diagram above, an OEM upgrade file containing the new HOST MCU image must be created and transmitted over Cresnet using the UPLOAD DATA command. Transfer to the Host MCU is done in two stages with the Cresnet Connected Interface Module (CCM) acting as a intermediary device that accepts the upgrade image encapsulated in Cresnet packets, stripping the Cresnet encapsulation and forwarding them to the Host MCU for handling.

The *CCM* uses the standard Cresnet file transfer command and protocol when communicating with the Control System (CS). The CCM uses the Cresnet Connected Serial Interface Protocol transferring ASCII data over the existing UART interface to the Host MCU.

A correctly structured OEM Upgrade File containing Motorola S19-style records is supplied to the CS when the upgrade is initiated. The CS will send this file to the CCM in blocks using a block size negotiated by the CCM. This block size is set by the **HOST** to ensure that it has sufficient memory to cache the full block as it is received. The CCM will then strip the Cresnet encapsulation and serially transmit each S19 record to the Host MCU for processing and acknowledgement. When all the information contained in a block is successfully transmitted the CCM will request the next block of the OEM Upgrade file from the CS until the final block is transferred and the upgrade is completed. At this point the CCM will initiate a reboot of the Host MCU so that it will begin executing the new code.

Key Technical Points

- The transfer of the OEM Upgrade File from the CS to the CCM uses existing Cresnet technology and is **independent and asynchronous** of the transfer of the received S19 records from the CCM to the HOST MCU, which uses a serial transfer of ASCII information over a UART link.
- The HOST MCU is responsible for **accepting, assembling and updating its operating image in local Flash memory.** The mechanism for doing this is beyond the scope of this document. The system image above indicates that the HOST MCU may assemble a full process image and then update it's image - it may also update the image in stages, as information is received and then switch to the new image on reboot.
- The OEM user is responsible for creating the OEM Upgrade Image in correct S19 format and **validating** the image at the HOST prior to upgrade.

Host Upgrade Protocol Details

New Messages

Host -> CCM Messages

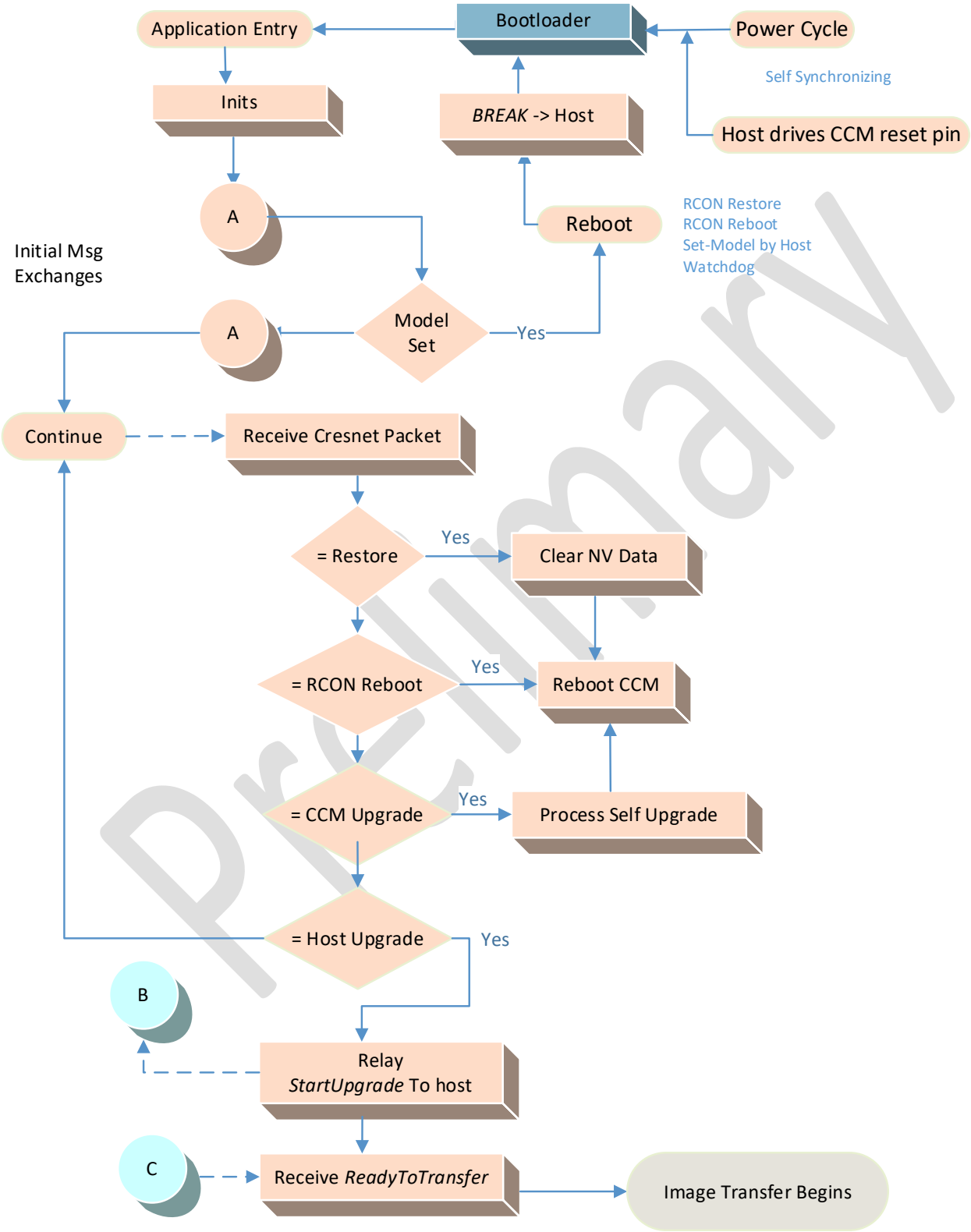
BootLoaderVersion	STRING: ***B****|< length>|<1-32 chars>*
AppVersion / Status	STRING: ***V****|< length>|<1-32 chars>|Status* *length* **must** be 2 chars, leading 0 if necessary *Status* = 1 : Host FW corrupted, Host board in BL *Status* = 0 : Host FW OK
ReadyToTransfer	*BINARY: 02 04 01* ***xx*** *0x0D* *(Block Size = 2^**xx**)* Example: 02 04 03 01 05 0d (block size = 2^5 = 32) 2^<xx> must be a multiple of the length of the S-records to be sent, thus <nn> will normally be >=5.
BlockFail / Abort	*BINARY:* 02 02 04 06 0x0d This **aborts** host firmware update, and the CCM resets itself
BlockOK	*BINARY:* 02 02 04 01 0x0d
Reset_CCM	This represents the actual reset pin on the CCM. The Host processor uses this to synchronize with the CCM or reset the CCM in case of persistent error communicating with the CCM.

CCM -> Host Messages

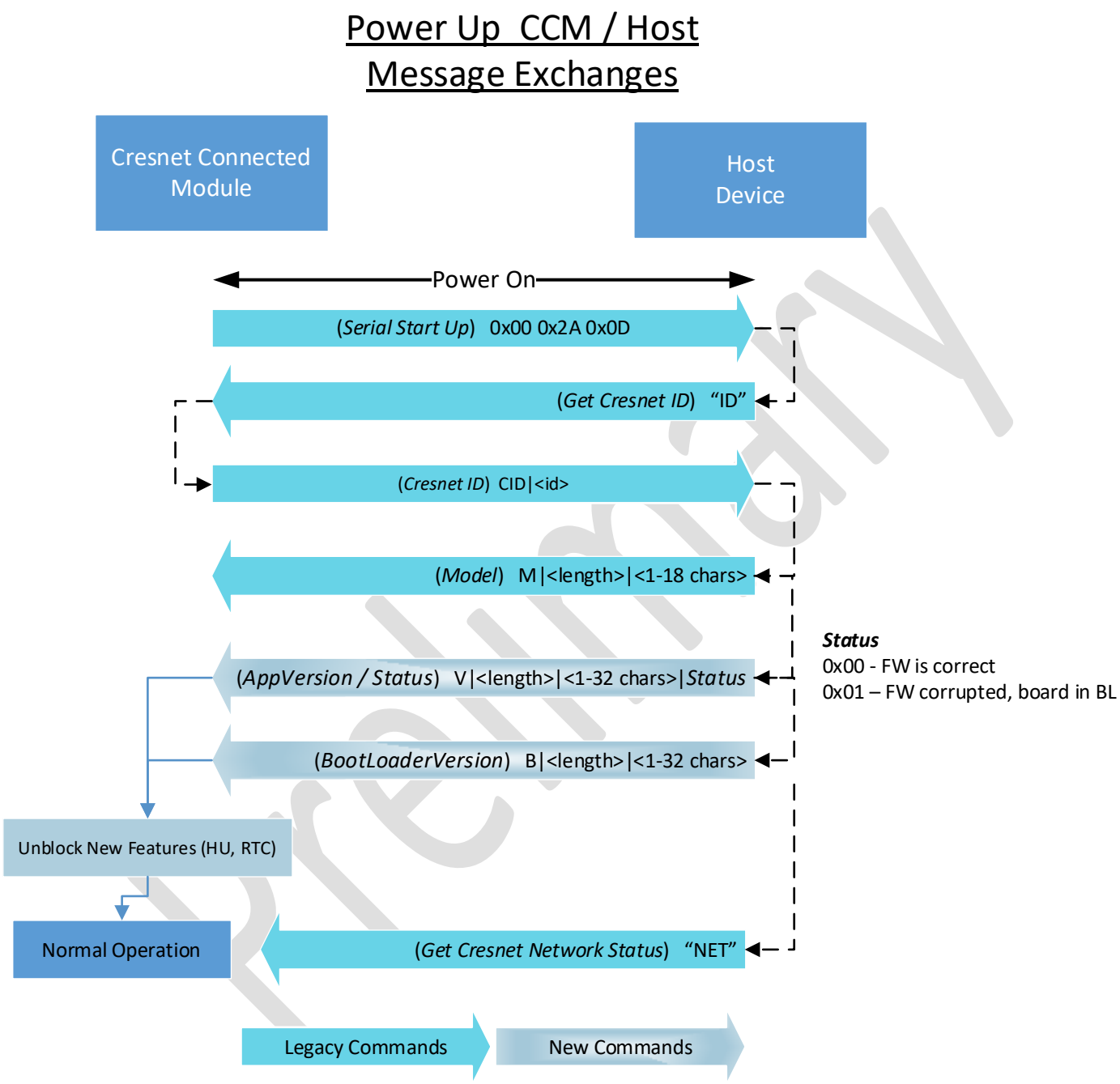
Start CCM Upgrade	STRING*: SYNC*
Start Host Upgrade	*BINARY: 0x00 0xAF 0x0D*
DataBlock	*BINARY: <Id> <cnt> 0x0D <addr> [data] <cs>* *<addr>: 2 bytes, <cs>: checksum*

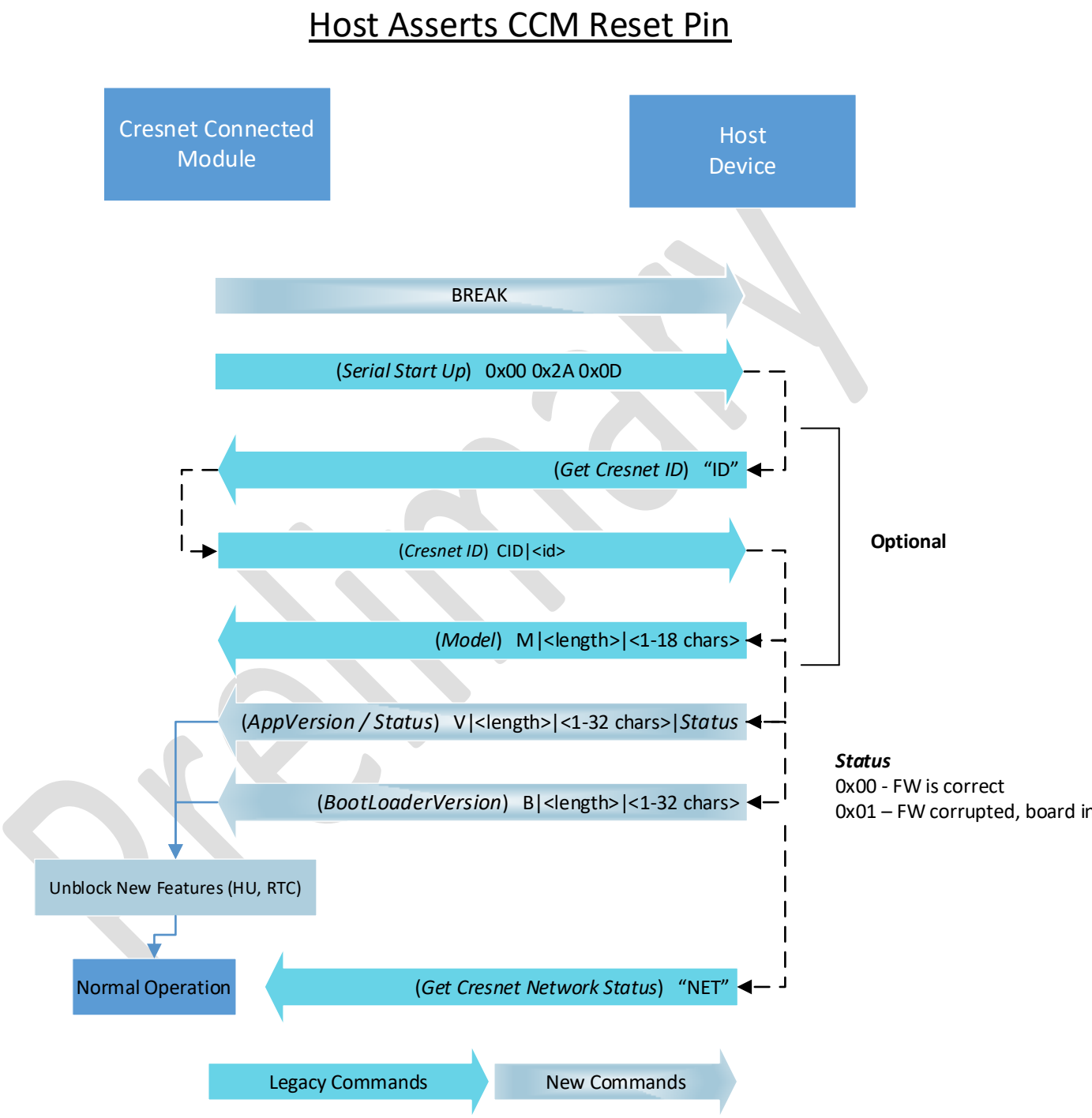
EndOfBlock	*BINARY: 0x02, 0x04, 0x05*
DownloadComplete	*BINARY: 0x02, 0x04, 0x02*
Break	*This is an RS-232 type break sent by the CCM to resynchronize the protocol. It is an out-of-band message that should always be decodable regardless of the host's protocol state.* *To signal a break the CCM Tx line is driven low for 10 bits followed by a stop bit.*
RTC	*T|hhmmssMMddyy* *"T|": beginning of RTC update packet* *"hh": Hours (24), "mm": Minutes, "ss": Seconds* *"MM": Month, "dd": Day of month, "yy": yearSince2000, "\r": 0x0d* *Example:* *Current Time/Date: 16:47:20 11-24-2023* *RTC update packet = "T|164720112423\r"* *The control system sends the local time to the CCM/host under the following conditions:* *The device connects/reconnects to the control system (update request)* *There is a shift of local time on the control system for any reason.*

CCM Startup Logic



Initial CCM / Host Message Exchange)





Sample Host Startup Logic

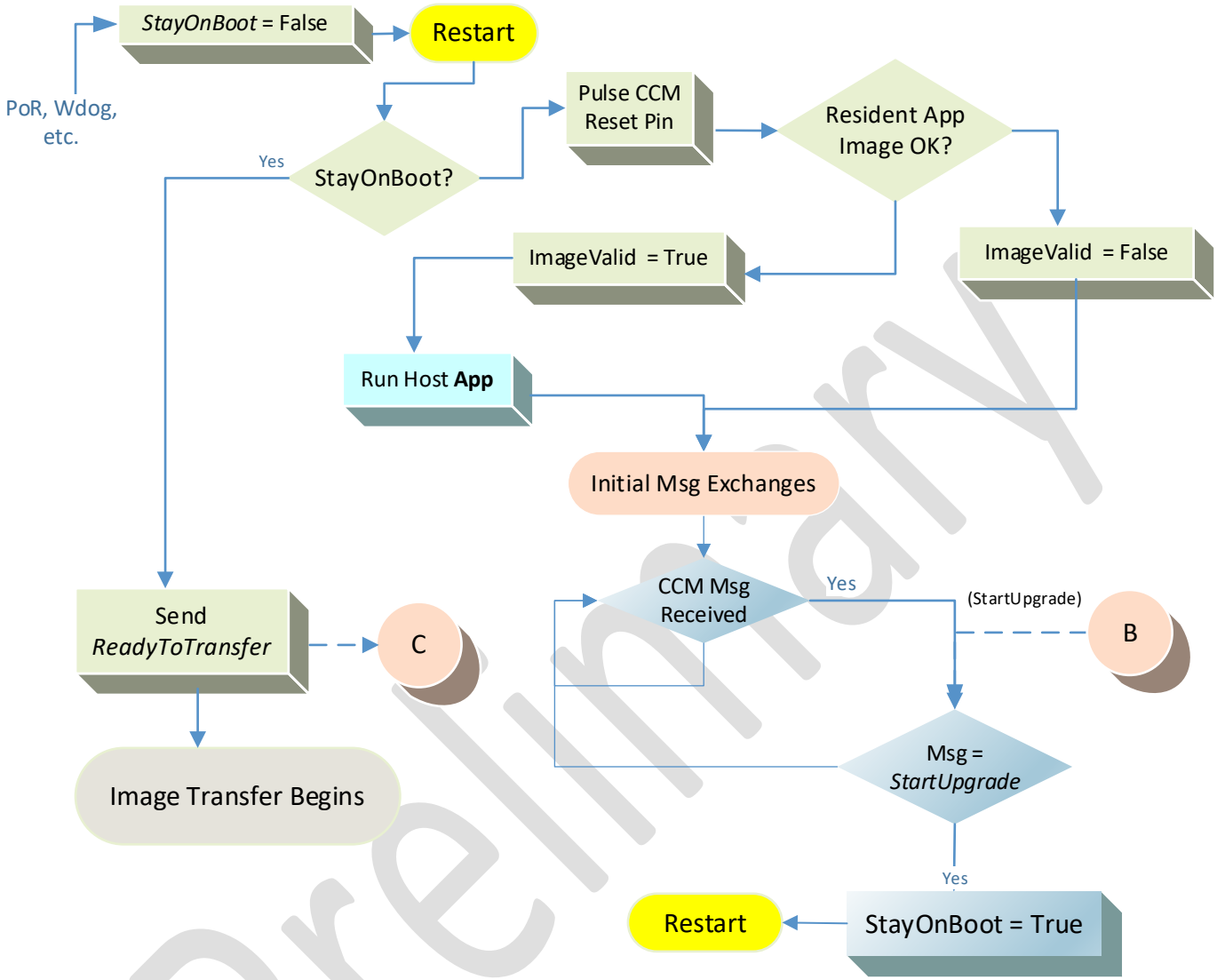


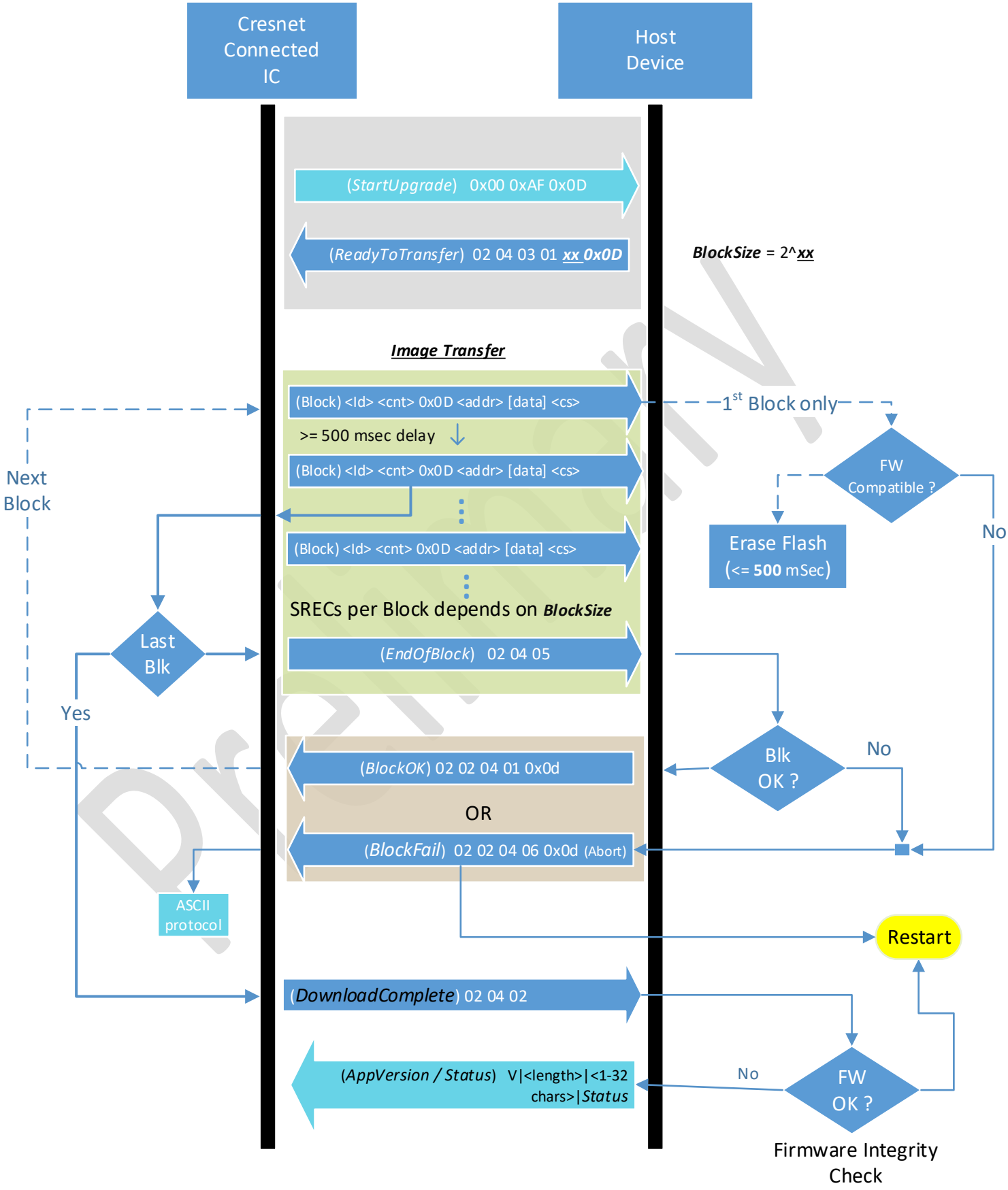
Image Transfer

General

- supports a wide range of host controllers
- can stream multiple S19 packets sequentially if the host have storage space for them
- uses the existing ASCII protocol with extensions noted above
- uses a more efficient binary method to transfer the information and reverts to the ASCII protocol when the transfer is completed successfully or has failed.
- makes use of the host's bootloader code (although a bootloader is not strictly necessary)

IMPORTANT:

- The **BLOCK size** is key to SREC operation - you can set the block size to only support single SREC blocks for Hosts that do not have resources to store more.



OEM Image File handling

Step 1: Preparing the OEM Upgrade File

- The OEM must create an OEM Upgrade file that is formatted in Motorola SREC format for transfer to the device. See https://en.wikipedia.org/wiki/SREC_(file_format) using S19 Style 16 Bit Records.

Step 2: OEM File Transfer From the CS to the CCM

Low level Transfer Method

The user initiates the transfer of the OEM Upgrade File to the CCM in the device using the UPLOAD XX DATA command, where "XX" is the Cresnet ID of the OEM device. **This command prepares the device to receive the file. Next, the file transfer needs to be initiated by selecting "XMODEM Send" under the Communication pull-down. It can also be done using Alt-U keys. Select the file to be sent using the presented dialog. After pressing Open the file is transferred block by block over Cresnet**. The sequence diagram below shows the Cresnet transfer in detail.

Notes:

- the Abort messages also go to Control System - showed separately to be handled as a failure.
- once the CCM begins receiving S19 Records it can begin transferring them to the OEM HOST (Step 3 Below). The HOST firmware is responsible for ensuring a complete valid image is provided before upgrading it's MCU.
- once each BLOCK complete message occurs the CCM must respond within 52 seconds with the Ready for Transfer to the CS to get the next block

HI Level Transfer Method – Package Upgrade File

Upgrading of both the Host and CCM can be achieved using a PUF file and the Crestron PUF Tool. The PUF file contains both images. The PUF tool inspects the host and CCM versions and upgrades each target, only as required. Normally, the PUF approach is desired.

The PUF tool comes as part of Crestron Toolbox software, available at Crestron.com.

Additional Details

Flash Erase Time

The flash device requires ~500 milliseconds to erase prior to programming. The flash device will be erased after the first SREC is received and confirmed to represent valid firmware for the device. Since handshaking occurs at the block level, the HOST device block size must be setup to allow the host to receive and store additional SRECs during flash erasure. If the host can only buffer 1 SREC, then 1 SREC per block needs to be used. For one SREC/block, the configuration is:

- X=5
- S19 file with 48 bytes per SREC

Modified S19

The S19 data used for file transmission over Cresnet is modified in the following manner:

- Converts every two hex characters to one byte of binary data
- No “S3”
- No CR/LF at end of records
- Format: Len, 0x0D, Adr, Data (includes CS)
- Sample data will be provided.

S-Record Checksum

```
// returns: TRUE for valid, FALSE for invalid checksum
BOOL cnupg_ProcessOneRecord( UINT8 *record )
{
   UINT8 checksum, *ptr, *dest;     UINT32 address, count;

   BYTE0( address ) = record[6];  BYTE1( address ) = record[5];
   BYTE2( address ) = record[4];  BYTE3( address ) = record[3];

   if ( address < g_CnetUpg.BlockAddress )
         return ( FALSE );

   count = record[1] - 6;

   g_CnetUpg.LastAddress = address + count - 1;

   if (( g_CnetUpg.LastAddress <= g_CnetUpg.BlockAddress ) ||
      ( g_CnetUpg.LastAddress >= g_CnetUpg.BlockAddress + g_CnetUpg.write_size ))
         return ( FALSE );

   if ( g_CnetUpg.LastAddress >= cnupg_CbGetAppMemoryEnd() )
         return ( FALSE );

   dest = g_CnetUpg.page_buffer + ( address & ( g_CnetUpg.write_size - 1 ));    // set offset into buffer
```

```
    checksum = record[1] - 1;          // original count from S3 record is this count less 1
    checksum += record[3] + record[4] + record[5] + record[6];  // add address field to checksum

    for ( ptr = &record[7]; count; count-- )     {
            checksum += *ptr;
            *dest++ = *ptr++;
    }

    checksum = ~checksum;

    if ( checksum != *ptr )
            return ( FALSE );

    return ( TRUE );
}
```

Timeouts

Where a response is expected, the receiving side will time-out after about 52 seconds. This applies to the host as well as the control system.

Version Display

The “version” or “ver” command is used to print the application version.

- “ver –v” will be used to print both the application and host versions. The host’s application and bootloader versions are printed.

Error Codes

Currently, Crestron does not provide error codes for upgrade failures. Success or failure is determined by checking the version of the target device.

Serial Number

OEMS do not to need to program serial numbers. The CCM-IC sets its own S/N based on a unique number contained in the CCM silicon. The CCM reports the S/N to the control system whether a host is used or not.

Legacy Devices

New CC⇔Host messages were created for host upgrade and *RTC* support. Products with legacy Host firmware could be harmed by these unanticipated messages. Uniquely, hosts with HU support send a firmware version string to the CC. If

the CC receives a version string it will not send any newly defined messages to the Host. These messages include *RTC* and all HU related messages. See New Messages section.

- Three categories of CC products from multiple vendors:
    - No Host – CC only – so no handshake, code will operate in legacy mode
    - Host without HU support – initial host handshake will fail – operate in legacy mode
    - Host with *HU* – host handshake will pass – operate in HU mode and provide RTC/alarm support