

## Matrix ARINC818/FCAV Header Function Usage

The Matrix ARINC818/FCAV card uses a new method for loading header data that is significantly different from the method used with the Gravity series cards. This document describes this method and the usage of the Matrix header functions.

### Why is a ARINC818 or FCAV header important ?

ARINC818 and FCAV have headers or “**Object0**” information at the beginning of the Image payload. This is typically used by the receiving end to determine the characteristic of the Video Stream. Loading a header is just as important as loading the image itself.

Refer to your ICD on the header information you require

### How do the Matrix Header functions work?

#### Receive Mode

In receive mode the Matrix header works the same as with the Gravity cards. The header data is simply the first part of the receive data, and is fetched with `GRTM_GetRawData` for A818/FCAV. As with the Gravity series cards, fetching of received header data with flow ON must be synchronized with the card’s usage of the memory buffers (usually in the callback).

#### Transmit Mode

In transmit mode on the A818/FCAV version of the Matrix card, the header works differently. The Matrix card has a special Transmit Header area in the card’s software core. The Matrix API (`GRTM_API`) provides functions for writing header data to this area. While this Transmit Header area cannot be read directly, the API driver keeps a “mirror” of the header data that can be read at any time. The advantage of this new way of loading transmit header data in the Matrix card is that it can be written to when transmit flow is ON, without the need to synchronize with buffer usage by the card. The card simply picks up new transmit header for transmission with the next frame sent. There is no distinction between MEM1 and MEM2 transmit header data and the same header data is sent with both frames. In addition, the API provides functions to change individual header words or bits, which can be useful for applications that control device parameters such as brightness or contrast. This method of handling the transmit header is currently implemented in the A818 (a.k.a. FCAV) version of the Matrix card.

**NOTE :** The image is still loaded at the header offset even though the header data in the MEM1 or MEM2 buffer is not transmitted. Always load image information starting at the Header offset. The header offset is also stored in the first 16 bits of PLD register at offset 0CH, This value is 1000H as defined in `GRTM_API.H` as `GRTM_IMAGE_OFFSET`

## ARINC818/FCAV Frame RX/TX and memory usage

The RX information including header is written to memory.

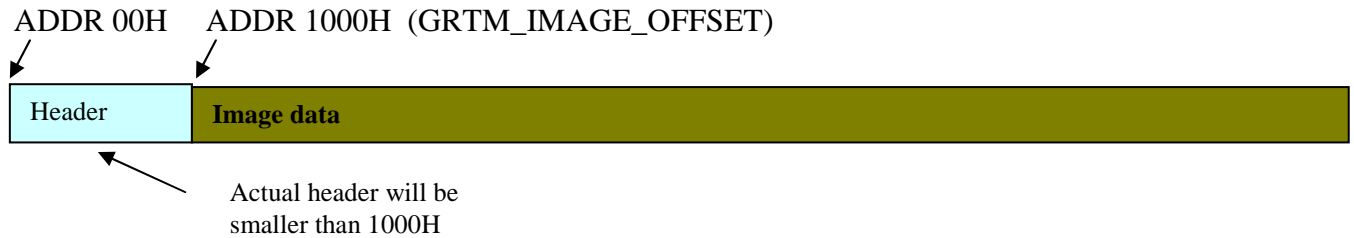
The header always gets stored at address 00H

The image always gets stored at address 1000H

The Image offset 1000H is stored in 0CH by the driver at OpenDriver time for ARINC818/FCAV cards

The actual header offset is stored in FPGA register 8CH.

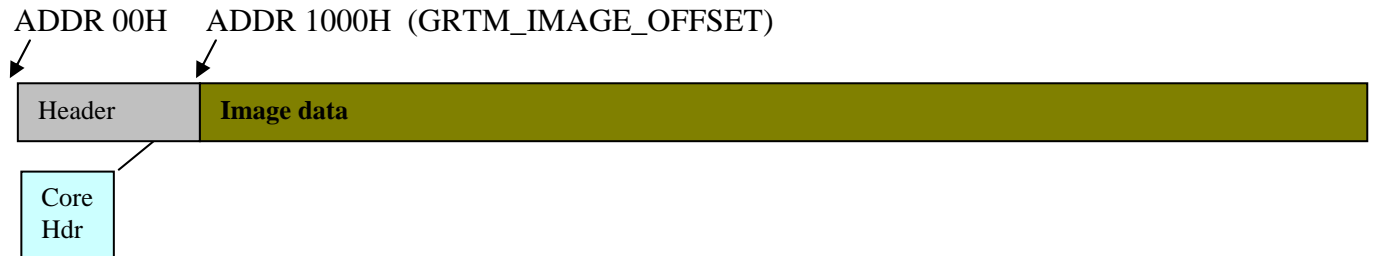
Example received frame



The TX image will be transmitted memory starting at 1000H.

The header is always located in the core itself.

Example transmitted frame



### Memory Map

Byte Addr	Area
00000000H	MEM1 Hdr memory
00000FFFH	
00001000H	MEM1 Image memory
01FFFFFFH	
02000000H	MEM2 Hdr memory
02000FFFH	
02001000H	MEM2 Image memory
03FFFFFFH	

## Summary of the Matrix Header functions

The following is a list of the Matrix Header Functions

GRTM_LoadHeader	High level load a header from a file on disk
GRTM_GetHeader	High level Store a header from the card to file on disk

GRTM\_LoadTxHeaderData  
GRTM\_CopyTxHeaderData  
GRTM\_WriteTxHeader  
GRTM\_WriteTxHeaderBits

For applications that need to simply load header data from a file for transmission once and do not need to change the transmitted header data, use GRTM\_LoadHeader.

For applications that have the transmit header data in a local buffer (either read from a file or created in the application itself) and want to load the entire header at once, use GRTM\_LoadTxheaderData (for A818/FCAV). If the application needs to change the header data, these functions can be used to reload the entire header. However, if only individual words or bits of the transmit header data need to be changed on a A818/FCAV card, use GRTM\_WriteTxHeader or GRTM\_WriteTxHeaderBits (see below)

For applications that have the transmit header data in a local buffer (either read from a file or created in the application itself) and want to load individual 32-bit words of the header, use GRTM\_WriteTxheader (for A818/FCAV). This usage would be appropriate for an application that needs to change header data words or bits to control receiving device brightness or contrast, for example.

For applications that have the transmit header data in a local buffer (either read from a file or created in the application itself) and want to load bits of individual 32-bit words of the header, use GRTM\_WriteTxHeaderBits (for A818/FCAV). This usage would be appropriate for an application that needs to change header data words or bits to control receiving device brightness or contrast, for example.

## Matrix Header Functions Description

This is only a summary of the functions for writing and reading transmit header data. Please see the SDK User's Manual for details on function parameters and usage.

### GRTM\_LoadHeader

This function operates much like the Gravity card's EFC\_LoadHeader function.

This function loads the file data directly into the cards' Transmit Header area (the function's u32MemSelect parameter is ignored). This function is convenient for loading an entire header from a disk file.

### GRTM\_GetHeader

This function operates much like the Gravity card's EFC\_GetHeader function.

This function retrieves header data from the memory area on the card. This header data is usually in memory from a capture of a frame that contains a header. This header data will be saved to the file specified. The header memory area accessed will be per the u32MemSelect parameter value.

### GRTM\_LoadTxHeaderData

This function loads full header data into a card for transmission. This function loads the file data directly into the cards' Transmit Header area. This function is convenient for loading an entire header from an application buffer. This function is for A818/FCAV cards only

### GRTM\_CopyTxHeaderData

This function copies header data from the driver's Transmit Header buffer (mirror), which always contains a copy of the current transmit header loaded into the card. While an application could keep a copy of the header data that it loaded for transmit, this function is convenient for getting a copy of the loaded data. Note that this function only gets a copy of the header data that was loaded, and will not contain any header data that the card changes during transmission (sequence numbers, etc.). This function is for A818/FCAV cards only

### GRTM\_WriteTxHeader

This function writes a single 32-bit data value directly to a card's Transmit Header location, using a byte address (offset) for the data value within the header. This function is useful for updating a single word in the header during transmission. This function is for A818/FCAV cards only

### GRTM\_WriteTxHeaderBits

This function writes bits in a single 32-bit data value directly to a card's Transmit Header location, using a byte address (offset) for the data value within the header. A mask is used to select the word bits to write. This function is useful for updating bits in a word in the header during transmission, such as brightness control bits. This function is for A818/FCAV cards only.