

Frequently Asked Questions

Title: Mystic - Recorded AR Binary File Format

Date: 09 Nov 15; updated 11 Dec 19

Card/Board/Module: ARINC-429 family

Operating System: all

Question:

What is the format of the .AR file generated by Mystic when I record to a file ?

Answer:

Format of Mystic generated (*.ar) ARINC Results File

The file is divided into two parts:

1. The header
2. The actual data

Header structure

bytes description

- | | |
|----|-------------------------------------|
| 4 | length of header |
| 6 | ID |
| 6 | internal |
| 4 | channel number |
| 4 | module number |
| 4 | mode |
| 4 | channel type |
| 60 | description |
| 4 | internal |
| 4 | date/time that the file was created |

Explanations of field contents:

- | | |
|---|------------------|
| 4 | length of header |
|---|------------------|

The length of the header is the sum of all the field lengths in the header. The length of the header is always equal to 100. This value is the same for all *.ar files.

- | | |
|---|----|
| 6 | ID |
|---|----|

The ID contains the constant string "MySt" with 2 terminating null characters

6 internal

This is a floating point value containing the Mystic version used to create this AR file

4 channel number

The channel number is stored as a 4 byte integer value. In merge mode, this value is the number of channels on the board. For example, if a board contains 4 channels (numbered 0, 1, 2, 3), then the merge channel number is 4.

4 module number

The module number is stored as a 4 byte integer value.

4 mode

Mode legal values are either CH_ARINC_429_RECEIVE (1) or CH_ARINC_429_TRANSMIT (2)

4 channel type

Here are #define strings for the legal values:

```
#define CH_NOT_INSTALLED 0
#define CH_ARINC_429_RECEIVE 1
#define CH_ARINC_429_TRANSMIT 2
#define CH_ARINC_561_RECEIVE 3
#define CH_ARINC_561_TRANSMIT 4
#define CH_ARINC_568_RECEIVE 5
#define CH_ARINC_568_TRANSMIT 6
#define CH_ARINC_575_RECEIVE 7
#define CH_ARINC_575_TRANSMIT 8
#define CH_ARINC_582_2_RECEIVE 9
#define CH_ARINC_582_2_TRANSMIT 10
#define CH_ARINC_582_6_RECEIVE 11
#define CH_ARINC_582_6_TRANSMIT 12
#define CH_RS_232 16
#define CH_RS_422 17
#define CH_RS_485 18
```

60 description

The description is a free form 60 byte null-terminated string. Nothing needs to be filled in.

4 internal

This value should be zero.

4 date/time that the file was created

The date/time is in the C language "time_t" format which is the number of seconds elapsed since 00:00:00 GMT, January 1, 1970

Data Structure

All received data are stored in consecutive blocks. The structure of each block is as follows:
[The blocks are contiguous and all have the same size (12) which is the sum of the bytes.]

bytes description

4 data word- the ARINC data word

2 time tag- high- the first 16 bits of the time tag (most significant bytes)

2 time tag- low- the last 16 bits of the time tag (least significant bytes)

To calculate the time tag in milli-seconds, perform the following operation:

In C: $(\text{time_tag_high} \ll 16 + \text{time_tag_low}) / 100$

Or, multiply the first 2 bytes by 2 to the power of 16, add the second 2 bytes and divide the sum by 100. This will give the time tag in milli-seconds.

Note that the **timetag resolution** is **10 micro-seconds**.

The **timetag listed in Mystic** when displaying a .AR file is in **milliseconds.microseconds** .

4 status word (see below for more bits)

The status word contains 2 relevant bits:

bit 03- Parity error: Indicates that an even parity error was detected in the word.

bit 07- Indicates that the received ARINC-429 word was valid in all respects.

The order of the bits is as follows:

15-14-13-12-11-10-09-08-07-06-05-04-03-02-01-00

Standard Mode Status Word for ARINC Channels

WORD_RECEIVED 0x1 (bit00) Indicates that a Status Word has been written

HI_BIT_CNT_ERR 0x2 (bit 01) A Hi Bit Count or Null Bit Error was detected in the ARINC word

LO_BIT_CNT_ERR 0x4 (bit 02) A Lo Bit Count or Null Bit Error was detected in the ARINC word

PARITY_ERROR 0x8 (bit 03) Parity error inserted in every word in message

INVALID_CODING_ERR 0x10 (bit 04) A bit level decoding error was detected in the ARINC word

GAP_TIME_ERR 0x20 (bit 05) A Gap (sync) Time Error occurred between words

VALID_WORD 0x80 (bit 07) Global bit – the received ARINC word was valid in all respects

NOTE: If the file is a Merge of a number of channels, the channel number for each message is extracted as follows:

Each record in the .ar file contains a status word. The structure of the status word is:

```
typedef struct rx_merge_status_word
{
  uint word_received :1;
  uint hi_bit_count_error :1;
  uint lo_bit_count_error :1;
  uint parity_error :1;
  uint invalid_coding_error :1;
  uint gap_time_error :1;
  uint reserved1 :1;
  uint valid_word :1;
  uint merge_channel_code :4;
  uint reserved2 :4;
} rx_merge_status_word;
```

Use merge_channel_code for the channel.