TP 13140E

FLIGHT RECORDER CONFIGURATION STANDARD

VERSION 2.0

Submitted to
Transportation Development Centre
Transport Canada

by
xwave

June 2003

TP 13140E

FLIGHT RECORDER CONFIGURATION STANDARD
VERSION 2.0

by:

Ian Smith and Robert Aubé
xwave
65 Iber Road
Stittsville, Ontario
K2S 1E7

June 2003

Document Revision History

Revision	**Reason for Issue**	**Origin Date**
Version 1.0	Revised document issued which addresses comments from international Field Trial	25 January 1999
Version 1.1	Revisions to front matter and Introduction; minor changes to File Format, References, and Sample FRCS File	16 October 2002
Version 2.0	Revised document introduces the ability to specify multiple parameter sources and indicate the presence of SDI and overlapping bits.	27 June 2003

Un résumé français se trouve à la page iv.

Transport Canada Transports Canada

PUBLICATION DATA FORM

1. Transport Canada Publication No.

TP 13140E

2. Project No.

8637

3. Recipient's Catalogue No.

4. Title and Subtitle

Flight Recorder Configuration Standard, Version 2.0

5. Publication Date

June 2003

6. Performing Organization Document No.

7. Author(s)

Ian Smith and Robert Aubé

8. Transport Canada File No.

ZCD1455-270

9. Performing Organization Name and Address

xwave
65 Iber Road
Stittsville, Ontario
K2S 1E7

10. PWGSC File No.

XSD95-00048-(621)

11. PWGSC or Transport Canada Contract No.

T8200-5-5505

12. Sponsoring Agency Name and Address

Transportation Development Centre (TDC)
800 René Lévesque Blvd. West
Suite 600
Montreal, Quebec
H3B 1X9

13. Type of Publication and Period Covered

Standard (June 2003 edition)

14. Project Officer

Howard Posluns

15. Supplementary Notes (Funding programs, titles of related publications, etc.)

This document is a revised edition of the publication entitled *Flight Recorder Configuration Standard, Version 1.1* (TP 13140E) published October 2002.

16. Abstract

This document describes Version 2.0 of the Flight Recorder Configuration Standard (FRCS). This Standard defines the content and format for electronic files that describe the flight data stored on a Flight Data Recorder System (FDRS). At the time this standard is being written, multiple formats exist for the different manufacturers and airline configurations. The maintenance of this information in a common, standard format should facilitate and expedite deciphering the recorded flight data. The intended users of the FRCS are airlines, manufacturers, and air safety investigators.

This document is intended for software systems designers and developers who wish to support the FRCS through enhanced or new software systems.

The FRCS content has been logically grouped into header, subframe (record), and parameter information. Parameter information is further grouped into identification, location, conversion, accuracy, sensor and signal, and digital information transfer system items.

The FRCS file format follows the logical groups defined in the FRCS content. The file is further defined as being a single ASCII (American Standard Code for Information Interchange) file, described in an extended Backus-Naur Form (BNF) notation. Interpretation of the file contents is achieved through the use of key words and the positional relationship of field items.

17. Key Words

Flight Recorder Configuration Standard (FRCS),
Flight Data Recorder (FDR), Subframe Layout,
Flight Data Acquisition Unit (FDAU), Flight Parameters

18. Distribution Statement

Limited number of copies available from the Transportation Development Centre

19. Security Classification (of this publication)

Unclassified

20. Security Classification (of this page)

Unclassified

21. Declassification (date)

—

22. No. of Pages

x, 29, app

23. Price

Shipping/ Handling

Transports Canada Transport Canada

FORMULE DE DONNÉES POUR PUBLICATION

1. N° de la publication de Transports Canada TP 13140E	2. N° de l'étude 8637	3. N° de catalogue du destinataire

4. Titre et sous-titre Flight Recorder Configuration Standard, Version 2.0	5. Date de la publication Juin 2003
	6. N° de document de l'organisme exécutant
7. Auteur(s) Ian Smith et Robert Aubé	8. N° de dossier - Transports Canada ZCD1455-270
9. Nom et adresse de l'organisme exécutant xwave 65 Iber Road Stittsville, Ontario K2S 1E7	10. N° de dossier - TPSGC XSD95-00048-(621)
	11. N° de contrat - TPSGC ou Transports Canada T8200-5-5505
12. Nom et adresse de l'organisme parrain Centre de développement des transports (CDT) 800, boul. René-Lévesque Ouest Bureau 600 Montréal (Québec) H3B 1X9	13. Genre de publication et période visée Norme (édition de juin 2002)
	14. Agent de projet Howard Posluns

15. Remarques additionnelles (programmes de financement, titres de publications connexes, etc.)

Ce document se veut une édition révisée de la publication du CDT portant le titre de *Flight Recorder Configuration Standard, Version 1.1* (TP 13140E) publiée en octobre 2002.

16. Résumé

La Norme de configuration des enregistreurs de vol (NCEV), version 2.0, précise le contenu et le format des fichiers électroniques contenant les données de vol stockées dans un Système d'enregistrement des données de vol (SEDV). Au moment où cette norme a été rédigée, de multiples formats étaient utilisés dans les diverses configurations employées par les fabricants et les lignes aériennes. Le fait de structurer l'information selon un format normalisé devrait faciliter et accélérer la traduction des données de vol enregistrées. La NCEV a été établie à l'intention des lignes aériennes, des fabricants et des enquêteurs dans le domaine de la sécurité aérienne.

Ce document s'adresse aux développeurs de logiciels qui souhaitent mettre au point des logiciels nouveaux ou de nouvelles versions de logiciels adaptés à la norme NCEV.

Le contenu des fichiers électroniques conformes à la NCEV se décompose en trois groupes logiques de données, soit l'en-tête, la sous-trame (enregistrement) et les données paramètres. Les données paramètres se subdivisent à leur tour en groupes d'articles (identification, localisation, conversion, exactitude, capteur et signal, système de transfert de données numériques).

Le format de fichier NCEV s'aligne sur les groupes logiques présentés ci-dessus. Ce format est défini en plus comme un fichier ASCII (*American Standard Code for Information Interchange*) décrit dans une notation étendue de la forme Backus Naur (BNF). Le contenu du fichier est interprété au moyen de mots clés et d'après la position relative des articles.

17. Mots clés Norme de configuration des enregistreurs de vol (NCEV), enregistreur de vol, disposition d'une sous-trame, bloc d'acquisition de données de vol (FDAU), paramètres de vol	18. Diffusion Le Centre de développement des transports dispose d'un nombre limité d'exemplaires.

19. Classification de sécurité (de cette publication) Non classifiée	20. Classification de sécurité (de cette page) Non classifiée	21. Déclassification (date) —	22. Nombre de pages x, 29, ann.	23. Prix Port et Manutention

Acknowledgements

xwave gratefully acknowledges the contribution of the following member organizations of the Project Review Committee.

Transport Canada, Civil Aviation Directorate and Transportation Development Centre
National Transportation Safety Board (U.S.)
Transportation Safety Board of Canada
Federal Aviation Administration (U.S.)
National Research Council Canada
Teledyne Controls
Flightscape Inc.
Delta Air Lines Inc.

The support of the other individuals and organizations who participated in the development of this standard is also appreciated.

TABLE OF CONTENTS

1. INTRODUCTION 1
  1.1 Background 1
  1.2 Scope 2

2. FILE CONTENTS 3
  2.1 Header Information 4
  2.2 Record Information 6
  2.3 Parameter Information 8
    2.3.1 Identification Items 8
    2.3.2 Location Items 9
      2.3.2.1 Sample Locations 10
      2.3.2.2 Superframe Cycles 12
    2.3.3 Conversion Items 13
    2.3.4 Accuracy Items 17
    2.3.5 Parameter Source 18
      2.3.5.1 Sensor and Signal Items 18
      2.3.5.2 Digital Information Transfer System (DITS) Items 19

3. FILE FORMAT 21

REFERENCES 29

APPENDIX A – STANDARD EQUATION CONVERSIONS

LIST OF FIGURES

Figure 1 Parameter Sample Concatenated from Three Components 10

GLOSSARY

ADAU	Auxiliary Data Acquisition Unit
ARINC	Aeronautical Radio Incorporated
ASCII	American Standard Code for Information Interchange
BCD	Binary Coded Decimal
BNF	Backus-Naur Form
BNR	Binary
DADS	Digital Air Data System
DIS	Discrete
DITS	Digital Information Transfer System
DME	Distance Measuring Equipment
EU	Engineering Units
FAA	Federal Aviation Administration
FDAU	Flight Data Acquisition Unit
FDR	Flight Data Recorder
FDRS	Flight Data Recorder System
FRCS	Flight Recorder Configuration Standard
GMT	Greenwich Mean Time
IEEE	Institute of Electrical and Electronics Engineers, Inc.
RMS	Root Mean Square
SDI	Source/Destination Identifier

1. INTRODUCTION

1.1 Background

xwave and the Transportation Development Centre of Transport Canada have developed the Flight Recorder Configuration Standard (FRCS) as an improved means of documenting flight data recorder systems (FDRS).

Since the introduction of digital flight data recorders (FDRs) in the 1960s, accident investigators have struggled with the documentation, which is required for the recovery of FDR data. The documentation provided by carriers is often inconsistent, inaccessible, incomplete, inaccurate, and difficult to interpret. This problem has become more significant as FDRSs have become more sophisticated and as more parameters are being recorded.

Difficulties in recovering the data from FDRs often delay the determination of the cause of accidents and consequently the identification of serious safety issues.

This documentation problem may also be experienced by carriers with other recorders (e.g., quick access recorders) that are being used for FDM/FOQA (flight data monitoring / flight operational quality assurance) programs. While the objectives of FDM/FOQA programs are to prevent accidents and to improve the safety and efficiency of the carrier's operation, the recorders used for these programs are also valuable to accident investigators.

The FRCS has been developed to help solve flight recorder documentation problems. The FRCS is an international standard that facilitates the exchange of FDR documentation by defining the content and format of electronic files, which document FDRs.

FRCS files:

- are in electronic format to facilitate their access and input to ground data replay and analysis systems,
- can be maintained by the carriers,
- can be provided by carriers to safety boards and other authorized users immediately upon request, and
- could be stored directly onto the FDRs.

The FRCS is applicable to commercial, military, FDM/FOQA, and maintenance recorders. Accident investigators, airlines, aircraft manufacturers, equipment manufacturers, maintenance organizations, regulators, flight data service providers, and others can benefit from using the FRCS.

The FRCS has been developed under the direction of an international project review committee, with members representing Transport Canada's Civil Aviation Directorate, National Research Council Canada, the U.S. Federal Aviation Administration, the U.S. National Transportation Safety Board, and the Transportation Safety Board of Canada. These organizations continue to maintain and support the FRCS.

Since its first release in March 1998, the FRCS has been endorsed by many safety boards, the FAA in Advisory Circular 20-141 [1], Transport Canada in Airworthiness Manual Advisory 551.100 [2], and EUROCAE in its Minimum Operational Performance Specification ED-112 [3]. Some airlines and aircraft manufacturers have produced FRCS files. Some manufacturers of ground stations and acquisition/recording systems have developed (or are developing) FRCS file import/export utilities.

xwave has developed an FRCS Editor and Tester to help create and validate FRCS files. The Editor and Tester both run on a standard PC.

The FRCS (Transport Canada Publication Number TP 13140E) and the Editor are available free of charge from **xwave**'s Web site (frcs.cbu.xwave.com) and from the Transportation Development Centre Web site (www.tc.gc.ca/tdc/projects/air/e/8637.htm).

This document describes Version 2.0 of the Flight Recorder Configuration Standard and reflects changes regarding the ability to specify multiple parameter sources and to indicate the presence of SDI and overlapping bits.

1.2 Scope

This document is intended for software systems designers and developers who wish to support the FRCS through enhanced or new software systems.

A variety of commercially available tools support the FRCS, including the FRCS Editor and Tester available free of charge from **xwave**'s Web site.

2. FILE CONTENTS

Electronic files that are compliant with the FRCS contain header, subframe (*record*), and parameter information. The header describes the aircraft, the on-board data acquisition and recording equipment, and the subframe (*record*) information that applies to all subframes (*records*). Separate, detailed information describes each recorded parameter.

The terminology employed in this standard is that used by Aeronautical Radio Incorporated (ARINC). Non-ARINC specific wording, such as may be applicable to military recorders, is noted in italics.

The following conventions are used for defining the type of data for particular items:

{Constrained Text}	Indicates an item that is constrained in some way (e.g., parameter names must be unique).
{Free Text}	Unrestricted text. Constrained to printable characters and may be constrained to a single line of text (see Section 3).
{Optional}	Indicates an optional (i.e., non-mandatory) item.
{**Mandatory**}	Indicates a mandatory item for which information must be entered.
{Numeric (Integer)}	Indicates a numeric integer value.
{Numeric (Unsigned Integer)}	Indicates a numeric integer value greater than or equal to zero.
{Numeric (Positive Integer)}	Indicates a numeric integer value greater than zero.
{Numeric (Real)}	Indicates a numeric real value.
{Numeric (Unsigned Real)}	Indicates a numeric real value greater than or equal to zero.
{Numeric (Positive Real)}	Indicates a numeric real value greater than zero.
{Numeric (Fraction)}	Indicates a numeric fractional or mixed fractional value.

{Binary, Octal} Numeric value constrained to be in binary or octal format.

{Boolean} Either True or False.

The convention pertaining to the "start" entries, as they apply to bits, words and subframes (*records*), is that they start at 1 (not 0). Bit 1 is the least significant bit of an FDR word.

The FRCS does not constrain the precision of floating point numbers. It is recommended that the minimum implementation allow for Real values specified to double precision as defined by the IEEE Standard 754 floating point standard which consists of 15 decimal digits accuracy and decimal exponents ranging between –308 and 308 [4].

For a complete definition of the data types and restrictions applied, please refer to Section 3, File Format.

2.1 Header Information

The following information is stored once in the header of the FRCS file.

A. FRCS Version Number {Constrained Text, **Mandatory**}

The version of the FRCS with which this file complies. Example: 2.0

B. File Version Identification (Number) {Free Text, Optional}

The version of the file. It may correspond to the version number of the paper documentation utilized to produce the file.

C. Aircraft Make and Model {Free Text, **Mandatory**}

The make and model of the aircraft to which this file applies. Example: Boeing 747-400.

D. Aircraft Registration Number(s)

{Free Text, Optional}

The registration number of the aircraft to which this file applies. Example: CGABC.

E. Aircraft Tail Number(s)

{Free Text, Optional}

The tail number of the aircraft to which this file applies.

F. Aircraft Serial Number(s)

{Free Text, **Mandatory**}

The serial number of the aircraft to which this file applies.

G. FDR Make and Model/Part Number(s)

{Free Text, Optional}

The Flight Data Recorder (FDR) Make and Model/Part Number on board the aircraft to which this file applies.

H. FDAU Make and Model/Part Number(s)

{Free Text, Optional}

The Flight Data Acquisition Unit (FDAU) Make and Model/Part Number on board the aircraft to which this file applies.

I. Sequential Subframes Flag

{Boolean, **Mandatory**}

Indicates whether the subframes (*records*) are stored in time sequence (ARINC) or not (e.g., MIL-STD-1553 recording devices). Example: True (for ARINC)

J. User-Definable Header Fields

{(Constrained Text, Free Text), Optional}

User-definable field names and values relevant to the entire file. The field names should be unique and consist of printable characters. Example: Field Name of "DFDAU Input Port" with Field Value of "B12".

K. User-Definable Parameter Field Names {Constrained Text, Optional}

User-definable field names relevant to each parameter. The field names should be unique and consist of printable characters. Example: Parameter Display Length.

L. Subframes per Frame[1] {Numeric (Positive Integer), **Mandatory**}

Number of unique subframes (*records*). There must be at least 1 subframe (*record*) per frame. Example: 4

M. File Modification Date and Time {Free Text, Optional}

Date and Time the FRCS file was last modified.

N. Comments {Free Text, Optional}

Comments relevant to the entire file. The file revision history should be noted here and should include, at a minimum, the date the file was created.

2.2 Record Information

The following items are stored for each subframe (*record*). If the Record Information for each subframe (*record*) is identical, this information is only stored once.

A. Bits per FDR Word {Numeric (Positive Integer), **Mandatory**}

Subframes (*records*) are broken up into smaller fragments referred to as FDR Words. FDR Words are not necessarily the same as computer words that are usually 16 bits long. There must be at least one bit per word.
Example: 12

[1] The Teledyne miniframe structure is accommodated by documenting each miniframe as a subframe. The Miniframe Id and Sync words at the beginning of each miniframe are concatenated to form unique 24-bit sync words. For example, in the case of 8 Miniframes/Subframe, 4 Subframes/Frame, and 1.0 Second/Subframe, the resulting FRCS file would show 32 Subframes/Frame, with 0.125 Seconds/Subframe.

B. FDR Words per Subframe

{Numeric (Positive Integer), **Mandatory**}

There must be at least one word per subframe (*record*). Examples: 32, 64, 128, ...

C. Leading Bits

{Numeric (Unsigned Integer), Optional}

The number of non-data bits at the beginning of the subframe (*record*). Should be left blank if unknown. Usually 0.

D. Trailing Bits[2]

{Numeric (Unsigned Integer), Optional}

The number of non-data bits at the end of the subframe (*record*). Should be left blank if unknown. Usually 0.

E. Seconds per Subframe

{Numeric (Positive Real or Fraction), **Mandatory**}

Indicates the time duration of one subframe (*record*) of data. Must be greater than zero. Examples: 1.0, 0.25, 1/3, 2 1/6

[2] The following equation defines the size of a subframe:
Bits per Subframe = (Bits per Word) x (Words per Subframe) + (Leading Bits) + (Trailing Bits)
Bit 1 of a subframe refers to the least significant bit of the first data word (i.e., Leading and Trailing bits are ignored).

2.3 Parameter Information

The following items are stored for each parameter.

2.3.1 Identification Items

The identification items for each parameter include a parameter name, mnemonic code, and parameter identification.

A. Parameter Name {Constrained Text, **Mandatory**}

The parameter name identifies the recorded parameter. The name shall be unique relative to all other parameters in the FRCS file and cannot have leading or trailing white space (tab or space) characters.

B. Mnemonic Code {Constrained Text, Optional}

This is a common abbreviation for the parameter name (e.g., IAS for Indicated Air Speed). The mnemonic shall be unique relative to all other parameters in the FRCS file.

C. Parameter Identification {Constrained Text, Optional}

This optional field may be used to assist in identifying the parameter within the file. It may be cross-referenced back to the identification number used by manufacturers. The identifications should be unique relative to all other parameter identifications in the FRCS file.

D. Record Identifier {Boolean, **Mandatory**}

This field is set to True if this parameter is used to uniquely identify the subframe (*record*). The Sync code for a subframe (*record*) is an example of such a parameter. There must be one Record Identifier parameter for each subframe (*record*). Where a parameter is the identifier for the subframe (*record*), it must only have one sample location within the subframe (*record*) and must have a Parameter Range of a single unique value (i.e., value distinct from all other Record Identifier parameters).

E. User-Definable Parameter Field Values {Free Text, Optional}

Values for user-definable parameter fields. The parameter field values are specified in the same sequential order as the parameter field names are specified in the Header (i.e., see Section 2.1 K).

F. Parameter Modification Date and Time {Free Text, Optional}

Date and Time the parameter description was last modified.

G. Comments {Free Text, Optional}

This field contains comments about the parameter. The parameter comments should include a modification history and the date the parameter was created.

2.3.2 Location Items

Each parameter may have one or more samples within a frame of data. In addition, if the parameter is a superframe parameter, the associated cycle parameter and counts are required. A superframe parameter is a parameter recorded less than once a frame (typically every 16, 32, 64 or 128 frames). The superframe counter indicates the cycle number for the superframe parameters recorded in the associated subframe (*record*) /word location.

2.3.2.1 Sample Locations

Each parameter sample consists of one or more components and timing items. Usually there is only one component per sample. For samples with more than one component, the components are ordered from least significant to most significant. The sample value is the concatenation of the components.

For example, three components may be defined as:

- Component 1: Subframe (*Record*) 1, Word 5, Overlap Bits 0, Bits 1-4;
- Component 2: Subframe (*Record*) 1, Word 2, Overlap Bits 0, Bits 2-3; and
- Component 3: Subframe (*Record*) 1, Word 4, Overlap Bits 0, Bits 7-12.

The resulting raw value of the sample is the concatenation of the three components as shown in Figure 1.

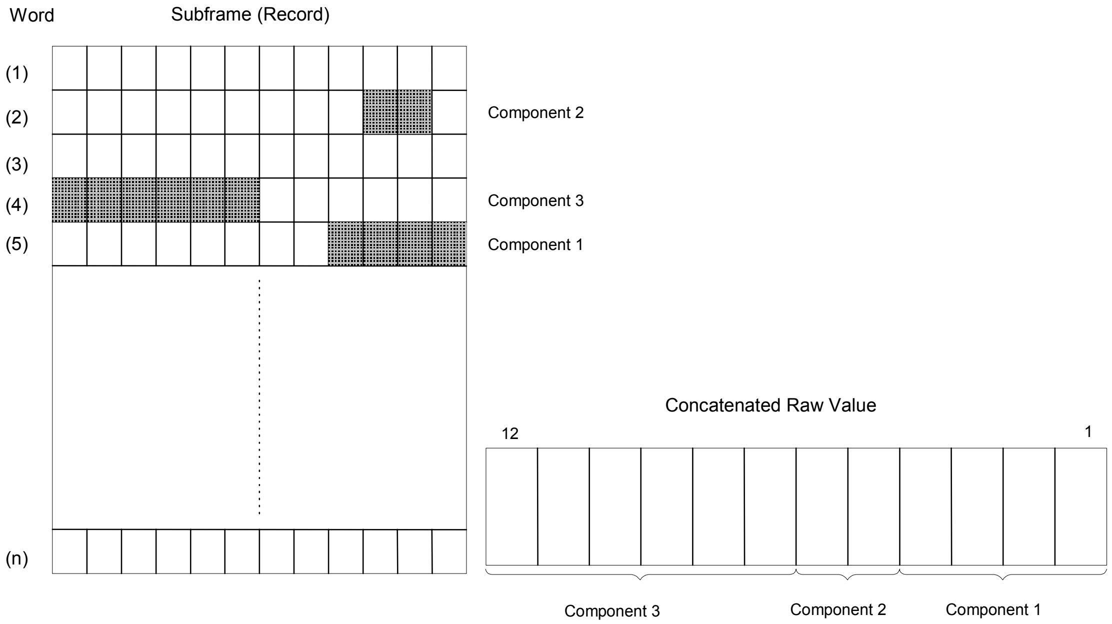

Figure 1 Parameter Sample Concatenated from Three Components

Each sample location for a specific parameter must have the same total number of bits (e.g., if the first sample location for Airspeed uses 12 bits, all other Airspeed sample locations must use 12 bits).

The following items are defined for a sample location.

A. Components

Each component is located by a subframe (*record*) number, word number, and a range of bits within the word.

A.1 Subframe Number

{Numeric (Positive Integer), **Mandatory**}

The subframe (*record*) in which a parameter sample component is stored. Subframe (*record*) numbers start at 1 and end at the number of subframes (*records*) in a frame.

A.2 Word Number

{Numeric (Positive Integer), **Mandatory**}

The subframe (*record*) word slot where a parameter sample component is stored. Word numbers start at 1 and end at the number of words in the subframe (*record*).

A.3 Overlap Bits

{Numeric (Unsigned Integer), **Mandatory**}

The number of bits at the most significant end of the component that overlap the next component. This value will always be 0 for parameters with only one component.

Overlap Bits are used to detect value rollover conditions when a parameter is stored in multiple components.

Example: An altitude parameter is made up of two components. The two components combined occupy 18 bits. Bits 1-12 of the sample are placed in subframe 1, word 105. Bits 12-18 of the sample are placed in subframe 3, word 125. Since bit 12 is stored in both components, the value 1 would be stored in the Overlap Bits field.

Components of multi-component parameters are typically recorded at different points in time. Using the above example, if, for a particular sample, the value stored at bit 12 in each of the components was not equal, the sample would be considered invalid.

A.4 Bits

{Numeric (Unsigned Integer, Unsigned Integer), **Mandatory**}

The range of one or more bits within a word in which a sample component is stored. Bit numbers start at 1 and end at the number of bits in a word.

B. Time Offset

{Constrained Text or Numeric (Unsigned Real), Optional}

For each sample location, this item refers to the time offset (in seconds) from the base subframe (*record*) sample time. It may be documented with:

- a specific value,
- an indication that the time is based on the Word Offset within the subframe (*record*), or
- an indication that the samples are Equally Spaced[3] within the subframe (*record*) time duration.

Most commonly this time is based on the Word Offset. It should be left blank if unknown. There must be more than one sample within the subframe (*record*) when Equally Spaced is specified.

2.3.2.2 Superframe Cycles

When superframe parameters are defined, at least one superframe cycle counter must be documented as a separate parameter. The items included when defining a superframe parameter are the following.

A. Cycle Counter Name

{Constrained Text, **Mandatory** for superframe parameters}

The parameter name of the associated superframe cycle counter parameter.

[3] The time offset of the first sample for an Equally Spaced parameter is always zero. For example, a parameter with four equally spaced samples in a subframe implies time offsets of 0, 0.25, 0.5, and 0.75 (for a subframe of one-second duration).

B. Cycle Numbers

{Numeric (Integer), **Mandatory** for superframe parameters}

A list of superframe cycle numbers for which this parameter is defined.

2.3.3 Conversion Items

Conversion items provide all the documentation necessary to convert from raw data values to Engineering Units (EU).

A. Signed Value

{Boolean, **Mandatory**}

Indicates whether the value contains a sign bit. The sign bit is assumed to be the most significant bit. This implies an initial two's complement conversion. Example: True for signed values.

B. Conversion

Each parameter may have zero or more Conversions. A conversion is defined by a unique raw data range and an associated list of Conversion Steps. If there is no Conversion, the raw data values are considered EU.

B.1 Raw Data Range

{Numeric (Unsigned Integer, Unsigned Integer), or Constrained Text, **Mandatory** for parameters with Engineering Unit conversion}

A range of raw data values to which the conversion steps apply. Raw data ranges for different conversions must not overlap. For a conversion that applies to the entire range of the parameter, "ALL" may be specified. Raw data ranges and corresponding conversions can be used to define piece-wise equations.

B.2 Conversion Steps

A list of one or more sequential steps required to convert a raw value to an EU value.

Each step is defined by one of the following four techniques:

- Polynomial Coefficients;
- Tabular Data;
- Predefined Equation; and
- Conversion Description.

a) Polynomial Coefficients

{Numeric (two or more Real), **Mandatory** if applicable}

Coefficients A_0, A_1, A_2, A_3, ... used in the following equation to convert a raw data count or intermediate value to an EU value.

$$EU = A_0 + A_1 * x + A_2 * x^2 + A_3 * x^3 + \ldots$$

Specifying A_0 and A_1, means Linear conversion.
Specifying A_0, A_1, A_2, A_3 ..., means Polynomial conversion.

b) Tabular Data

{Numeric (Real, Real), **Mandatory** if applicable}

Tables contain raw data counts or intermediate values and their corresponding EU values. For non-discrete parameters, unspecified values in the table can be found using a suitable interpolation method.

c) Predefined Equation

{Constrained Text, **Mandatory** if applicable}

This item refers to one of the following predefined equations: Teledyne Synchro, Fairchild Synchro, Binary Coded Decimal (BCD). Appendix A describes each of the predefined equations in detail.

d) Conversion Description

{Free Text, **Mandatory** if applicable}

A free text description of the method of conversion to EU.

C. Conversion Accuracy

{Numeric (Unsigned Integer, Unsigned Integer, Real) or Numeric (Real), Optional}

This is the accuracy of the mathematical conversion algorithms used to convert from raw data counts to EU. It may be expressed as a single correlation coefficient (e.g. 1.0 represents perfect correlation) or a table of absolute values (representing plus/minus uncertainty associated with the engineering unit value) over the valid operational raw data range of the parameter. The correlation coefficient shall be calculated as follows:

$$r = \frac{N*\Sigma\,(x*y) - \Sigma x*\Sigma y}{\{\,[N*\Sigma x^2 - (\Sigma x)^2] * [N*\Sigma y^2 - (\Sigma y)^2]\,\}^{1/2}}$$

Where: r = correlation coefficient
N = number of data points
x = reference data (EU)
y = derived data (EU)
Σ indicates the sum of the values that follow
(e.g. Σx equals the sum of all x values)

D. Units

{Free Text, **Mandatory** for parameters with units of measure}

These are the EU of measure of the parameter (e.g., Degrees, Radians, Feet, Knots, G).

E. Interpretation

{Numeric (Real, Real), Free Text, **Mandatory** for discrete and signed parameters}

For discrete parameters, this constitutes a table of discrete codes and their corresponding interpretations for a given parameter. Descriptions of all states should be entered.

For example:

Engineering Units[4]	**Discrete Status**
96	"OFF"
97	"RETARD"
98	"CLAMP"

For non-discrete parameters, this constitutes a table of EU parameter ranges with corresponding interpretations. For signed parameters this table must describe the sign convention (meanings for both positive and negative values should be entered).

For example:

Engineering Units Range	**Interpretation/Sign Convention**
<0	Aircraft Nose Down
0	Aircraft Level
0>	Aircraft Nose Up

[4] If there is no parameter conversion, the raw counts are considered Engineering Units.

2.3.4 Accuracy Items

These items pertain to the accuracy of the parameter data.

A. Parameter Range

{Numeric (Real, Real), **Mandatory** for parameters that are Record Identifiers (sync. code parameters)}

This is the valid operational range of the parameter (minimum and maximum) expressed in EU appropriate to the aircraft in question (e.g. the total angular movement of a control surface) irrespective of the range available from the associated transducer or the magnitude of the word on a data-bus.

B. Parameter Accuracy

{Numeric (Real, Real, Real), Optional}

This is the total accuracy of the input data before recording as described in Section 4.1.1 of ARINC 573 [5] and ARINC 717 [6]. Some factors included in the accuracy figure are:

- Sensor Tolerances;
- FDAU Tolerances; and
- Non-linearities of the Analog to Digital Conversion Process.

The total accuracy for a parameter may be expressed as a plus or minus root mean square (RMS) error value in engineering units or as a percentage error. It may consist of a single value or for varying accuracies a table of values over the valid operational range of the parameter.

For example:

Parameter Range	**Accuracy (±)**
x < 10000 ft	20 ft
10000 = x < 30000 ft	25 ft
x = 30000 ft	30 ft

C. Parameter Resolution {Free Text, Optional}

This refers to the resolution of the recorded data in engineering units. It is also documented in terms of raw data counts and is usually based on a number of bits. It generally represents the minimum (or worst-case) resolution over the parameter range. For parameters where the resolution is not linear over the full range, it may also be specified in terms of the minimum, intermediate, and maximum resolutions for the parameter range. Should be left blank if unknown.

D. Transport Delays {Numeric (Unsigned Real), Optional}

This refers to acquisition channel delays which are defined as the time (in milliseconds) between the input sampling and the output for recording. Normally zero. Should be left blank if unknown.

2.3.5 Parameter Source

A Parameter Source identifies the data source for a particular parameter. Multiple sources (e.g., bus switching) may be identified using multiple Parameter Sources. Each Parameter Source contains one Sensor and Signal Items (to identify the bus/physical location) and optionally one DITS Items.

2.3.5.1 Sensor and Signal Items

The sensor and signal items are described below.

A. Sensor Type {Free Text, Optional}

This is text indicating the type of sensor used to measure the parameter (e.g., Gyroscope, Accelerometer).

B. Signal Type {Free Text, Optional}

These signal types are as defined in Sections 4.2 through 4.4 of ARINC 573 [5] and ARINC 717 [6].

These include the following:

- Analog Input Signals. For example: Synchro, AC Voltage Ratios, Potentiometer, Strain Gauge, and Thermocouples.
- Digital Inputs. For example: Digital Air Data System (DADS), Auxiliary Data Acquisition Unit (ADAU), and Greenwich Mean Time (GMT).
- Discrete Inputs. For example: On/Off signals, series, shunt, AC sensitive, Ident, and marker beacon.

C. Signal/Data Source {Free Text, Optional}

This field identifies the aircraft subsystem (e.g., Air Data Computer, Cockpit Display, Inertial Reference Unit), dedicated transducer, or signal conditioners, which is the primary signal source for this parameter. If the source may be switched, this should be documented.

2.3.5.2 Digital Information Transfer System (DITS) Items

The Digital Information Transfer System items are listed below.

A. SDI Flag {Boolean, Optional}

Bits 9 and 10 normally provide a Source/Destination Identifier (SDI). In some cases, these bits are used for data. The SDI flag is TRUE when bits 9 and 10 represent the SDI and FALSE when these bits represent data. Should be left blank if it is not known what bits 9 and 10 represent.

B. DITS Label

{Numeric (Octal)}

The Digital Information Transfer System label is the octal value (0-1777) that identifies the type of data (e.g., Distance Measuring Equipment (DME), Static Air, etc.). Should be set to 0 if unknown.

C. DITS Bit Numbers

{Numeric (Unsigned Integer, Unsigned Integer), Optional}

As defined in ARINC Characteristic 429 [7], the standard DITS format is a 32-bit BCD, binary (BNR), or discrete (DIS) word, with bits (11-29) generally assigned as the data bits. The remaining bits contain labeling, parity, source/destination codes, and sign/status information. Documentation shall indicate all ARINC 429 data bits as they are assigned or mapped to the FDRS data bits.

D. DITS Coding

{Free Text, Optional}

DITS coding identifies the type of data (e.g., binary, BCD, or discrete).

3. FILE FORMAT

The FRCS electronic file format consists of a single ASCII (American Standard Code for Information Interchange) file. Interpretation of the file contents is achieved through the use of key words and the positional relationship of field elements. The file is divided into series and sub-series containing one or more items.

The FRCS format is described using an extended Backus-Naur Form (BNF) notation. The differences from pure BNF pertain to notational conventions. The notational conventions are as follows:

::=	The left-hand-side of the symbol "consists of" elements on the right hand-side. This symbol may be used to define series and subseries.
< >	Angle brackets enclose a series or subseries name.
{ < > / < > }	Series separated by a backslash and enclosed in brackets denote multiple alternatives.
*	An asterisk preceding an element indicates a sequence of zero or more elements.
+	A plus sign preceding an element indicates a sequence of one or more elements.
*x or +x	‘x’ represents the upper limit on the number of elements (e.g., +3 indicates one to three elements)
[]	Square brackets enclose optional elements. There are zero or one of these elements. In the case of zero elements the blank field should still be comma-delimited.
()	Round brackets enclose multiple elements grouped together.
""	Literal text such as for key words are enclosed in double quotation marks.

Also, note that individual field elements are written in capitalized, hyphenated format and are not enclosed in angle brackets < >.

When producing an FRCS file, the following rules apply:

- Extra white space characters (space and tab) are allowed.
- Extra new line characters are not allowed.

The syntax for the FRCS file is shown below. Clarifying remarks and additional constraints (not explicitly represented by the grammar) are included in the Notes. Software utilizing FRCS format files should allow input and output of files as constrained by the syntax only. Subsequent manipulation of the content should adhere to the constraints listed in the Notes and described in Section 2. Software developers must not make specialized use of free text fields (e.g., Comments) to indicate software dependent flags, etc.

Rules	Grammar	Notes
FRCS_File_Format ::=	<Header_Information> +<Record_Information> <Parameter_Information>	
<Header_Information> ::=	"HEADER:" NL FRCS_Version_Number "," File_Version_Identification "," Aircraft_Make_Model "," Aircraft_Registration_Numbers "," Aircraft_Tail_Numbers "," Aircraft_Serial_Numbers "," FDR_Make_Model_Part_Numbers "," FDAU_Make_Model_Part_Numbers "," Sequential_Subframes_Flag "," *<User_Def_Header_Fields> "," *Parameter_Field_Names "," Subframes_per_Frame "," Date_Time_Stamp "," Comments NL	See Section 2.1 for mandatory and optional text fields.
<Record_Information> ::=	"RECORD:" NL Bits_per_FDR_Word "," FDR_Words_per_Subframe "," [Leading_Bits] "," [Trailing_Bits] "," Seconds_per_Subframe NL	The number of <Record_Information> items required is equal to Subframes_per_Frame except in the case where all <Record_Information> items are identical, in which case only one <Record_Information> item is stored.
<Parameter_Information> ::=	{"NONE" / +<Parameters>}	For FRCS files which are in progress, it is possible to save a file with no parameters defined. However, the completed file should describe all recorded parameters.
<User_Def_Header_Fields> ::=	"[" Field_Name SPACE Field_Value "]"	

Rules	Grammar	Notes
<Parameters> ::=	"PARAMETER:" NL <Identification_Items> <Location_Items> <Conversion_Items> <Accuracy_Items> +<Parameter_Source> NL	
<Identification_Items> ::=	Parameter_Name "," Mnemonic_Code "," Parameter_ID "," Record_Identifier "," <User_Def_Field_Values> "," Date_Time_Stamp "," Comments NL	See Section 2.3.1 for mandatory and optional text fields. Parameter_Name must be unique relative to all other parameters in the FRCS file and cannot have leading or trailing white space characters. Mnemonic_Code and Parameter_ID must be unique relative to all other parameters in the FRCS file. There must be Subframes_per_Frame parameters with Record_Identifier as True. The number of User_Def_Field_Values (n) should correspond to the number of Parameter_Field_Names specified in the header and they should be in the same sequential order.
<Location_Items> ::=	+<Sample_Locations> [<Superframe_Cycles>] NL	
<Conversion_Items> ::=	Signed_Value "," *<Conversions> "," [<Conversion_Accuracy>] "," Units "," [<Interpretation>] NL	
<Accuracy_Items> ::=	[Parameter_Range] "," [<Parameter_Accuracy>]"," [Parameter_Resolution] "," [Transport_Delays] NL	For parameters with Record_Identifier set to true or TRUE, Parameter_Range is mandatory and must consist of a single value.
<Parameter_Source> ::=	<Sensor_Signal_Items> [<DITS_Items>] NL	

Rules	Grammar	Notes
<Sensor_Signal_Items> ::=	Sensor_Type "," Signal_Type "," Signal_Source NL	
<DITS_Items> ::=	[SDI_Flag] "," DITS_Label "," [DITS_Bit_Number] "," DITS_Coding NL	
<User_Def_Field_Values> ::=	Field_Value SPACE	
<Sample_Locations> ::=	+<Components> Time_Offset NL	
<Components> ::=	Subframe_Number "," Word_Number "," Overlap_Bits "," Bits NL	Subframe_Number must be less than or equal to Subframes_per_Frame. Word_Number must be less than or equal to FDR_Words_per_Subframe for the given subframe. Overlap_Bits is always 0 for a parameter sample with only 1 component.
<Superframe_Cycles> ::=	Cycle_Counter_Name"," Cycle_Numbers	Cycle_Counter_Name must equal one Parameter_Name. Values in the Cycle_Numbers must be within the Parameter_Range of the parameter indicated by the Cycle_Counter_Name.
<Conversions> ::=	Raw_Data_Range"," +<Conversion_Step> NL	The Raw_Data_Range must not overlap other Raw_Data_Ranges for the Conversions for a particular parameter.
<Conversion_Step> ::=	{"POLYNOMIAL:" <Polynomial>/ "EUTABLE:" <EU_Table> / "STANDARD:" <Predefined_Equation> / "DESCRIPTION:"<Conversion_Description>} NL	
<Conversion_Accuracy> ::=	{<Int_Table> / EU_Accuracy} NL	Correlation coefficient is specified using a single EU_Accuracy value. Int_Table is used to represent EU accuracies over a range of raw values.
<Parameter_Accuracy> ::=	<Accuracy_Type> <Real_Table> NL	
<Accuracy_Type>::=	{"RMS"/"Percent"}	
<Interpretation> ::=	+(<Real_Range> free_text) NL	Individual Real_Ranges for a parameter Interpretation must not overlap.

Rules	Grammar	Notes
<Polynomial> ::=	Coefficient +(SPACE Coefficient) NL	
<EU_Table> ::=	+(real_number SPACE real_number) NL	
<Real_Table> ::=	+(<Real_Range> EU_Accuracy) NL	
<Int_Table> ::=	+(raw_range SPACE EU_Accuracy) NL	
<Real_Range> ::=	Range_Start Real_Range_Value SPACE Real_Range_Value Range_End	
<Predefined_Equation> ::=	{"FairchildSynchro" / "TeledyneSynchro" / BCD} NL	
<Conversion_Description> ::=	Comments NL	
<BCD> ::=	"BCD" [SPACE +non_zero_digit]	

The data types used by the FRCS Grammar are defined below:

Data Types	Grammar	Notes
Aircraft_Make_Model ::=	free_text	
Aircraft_Registration_Numbers ::=	free_text	
Aircraft_Serial_Numbers ::=	free_text	
Aircraft_Tail_Numbers ::=	free_text	
Bits ::=	bit_range	
Bits_per_FDR_Word ::=	positive_integer	
Coefficient ::=	real_number	
Comments ::=	{free_text / multi_line_free_text}	
Cycle_Counter_Name::=	free_text	
Cycle_Numbers::=	digit_list	
Date_Time_Stamp::=	free_text	
DITS_Bit_Number ::=	bit_range	
DITS_Coding ::=	free_text	
DITS_Label ::=	octal_value	
EU_Accuracy ::=	real_number	
FDAU_Make_Model_Part_Numbers ::=	free_text	
FDR_Make_Model_Part_Numbers ::=	free_text	
FDR_Words_per_Subframe ::=	positive_integer	
Field_Name ::=	text	Field_Name must be unique relative to all other header field names in the FRCS file.
Field_Value ::=	free_text	
File_Version_Identification ::=	free_text	
FRCS_Version_Number ::=	DOUBLE_QUOTE "2.0" DOUBLE_QUOTE	
Leading_Bits ::=	unsigned_integer	
Mnemonic_Code ::=	free_text	
Overlap_Bits ::=	unsigned_integer	
Parameter_Field_Names ::=	text	Parameter_Field_Names must be unique relative to all other parameter field names in the FRCS file.

Data Types	**Grammar**	**Notes**
Parameter_ID ::=	free_text	
Parameter_Name ::=	free_text	
Parameter_Range ::=	EU_range	If Signed_Value is FALSE or false, the Parameter_Range cannot include negative values.
Parameter_Resolution ::=	free_text	Because up to three optional values may be entered, this field is stored as free_text. The text should only contain real numbers.
Range_End ::=	{ "]" / ")" }	The end of a range of real numbers can be inclusive of the end value as represented by "]", or exclusive as represented by ")".
Range_Start ::=	{ " [" / " (" }	The start of a range of real numbers can be inclusive of the start value as represented by "[" or exclusive as represented by "(".
Raw_Data_Range ::=	{raw_range / "ALL"}	
Real_Range_Value::=	{real_number / "MIN"/"MAX" }	The minimum and maximum parameter values are represented using the keywords "MIN" and "MAX". For example, the range representing parameter values less than zero is "[MIN 0) ".
Record_Identifier ::=	boolean	
SDI_Flag ::=	boolean	
Seconds_per_Subframe ::=	{positive_real_number / fraction / mixed_fraction}	
Sensor_Type ::=	free_text	
Sequential_Subframes_Flag ::=	boolean	
Sign_Bits ::=	unsigned_binary_digit_sequence	
Signal_Source ::=	free_text	
Signal_Type ::=	free_text	
Signed_Value ::=	boolean	
Subframe_Number ::=	positive_integer	
Subframes_per_Frame ::=	positive_integer	

Data Types	Grammar	Notes
Time_Offset ::=	{"WORD_OFFSET"/ "EQUAL_SPACED"/ "NOT_SPECIFIED"/ unsigned_real_number }	If EQUAL_SPACED is specified, there must be more than one sample for the parameter within the given subframe and each sample must also have a Time_Offset of EQUAL_SPACED. Time_Offset must be less than Seconds_per_Subframe.
Trailing_Bits ::=	unsigned_integer	
Transport_Delays ::=	unsigned_real_number	
Units ::=	free_text	
Word_Number ::=	positive_integer	

The data types are further described as follows:

Data Type	Grammar	Notes
bit_range ::=	unsigned_integer SPACE unsigned_integer	Bits range values must be from low bit number to high bit number and within the Bits_per_FDR_Word for the given subframe.
boolean ::=	{"true" / "false"/ "TRUE"/ "FALSE"}	
digit ::=	{"0"/"1"/"2"/"3"/"4"/"5"/"6"/"7"/"8"/"9"}	
digit_list ::=	unsigned_integer *(SPACE unsigned_integer)	
EU_range ::=	real_number SPACE real_number	
fraction ::=	positive_integer "/" positive_integer "	
free_text ::=	DOUBLE_QUOTE *(qtext) DOUBLE_QUOTE	
mixed_fraction ::=	unsigned_integer SPACE fraction	
multi_line_free_text ::=	DOUBLE_QUOTE *{qtext / NL} DOUBLE_QUOTE	
non_zero_digit ::=	{"1"/"2"/"3"/"4"/"5"/"6"/"7"/"8"/"9"}	
octal_digit ::=	{"0"/"1"/"2"/"3"/"4"/"5"/"6"/"7"}	
octal_value ::=	+4(octal_digit)o |in range 000-1777|	
positive_integer ::=	non_zero_digit *(digit)	
positive_non_zero_real::=	{positive_integer / positive_integer "." unsigned_integer / positive_integer "."}	
positive_real_number	{positive_non_zero_real/ positive_non_zero_real {"E"/"e"} ["-"] unsigned_integer}	
qtext ::=	|any printable ASCII character except DOUBLE_QUOTE, or NL|	
raw_range ::=	unsigned_integer SPACE unsigned_integer	The raw_range must be from low value to high value and within the constraints of the number of bits in the samples.

Data Type	Grammar	Notes
real_number ::=	{["-"] unsigned_real_number / ["-"] unsigned_real_number {"E"/"e"} ["-"] unsigned_integer}	
text ::=	DOUBLE_QUOTE +(qtext) DOUBLE_QUOTE	
unsigned_integer ::=	+(digit)	
unsigned_real_number::=	{unsigned_integer / unsigned_integer "." unsigned_integer / "." unsigned_integer / unsigned_integer "."}	
CR ::=	|ASCII carriage return|	
CRLF ::=	CR LF	
DOUBLE_QUOTE ::=	|ASCII double quote mark|	
LF ::=	|ASCII line feed|	
NL ::=	{CRLF / CR / LF}	
SPACE ::=	|ASCII space character|	

Key words are utilized as described in the file syntax to facilitate automated parsing. The following is the list of key words:

- HEADER:
- RECORD:
- PARAMETER:

REFERENCES

1. Federal Aviation Administration, Advisory Circular 20-141, *Airworthiness and Operational Approval of Digital Flight Data Recorder Systems*, October 5, 1999 (AIR-130).

2. Transport Canada, Airworthiness Manual Advisory 551.100, *Flight Data Recorder Installation Calibration And Correlation Procedures*, December 1, 1998.

3. European Organisation for Civil Aviation Equipment, ED-112, *Minimum Operational Performance Specification for Crash Protected Airborne Recorder Systems*.

4. Institute of Electrical and Electronics Engineers, *IEEE Standard for Binary Floating-Point Arithmetic*, ANSI/IEEE Standard 754-1985, August 1985.

5. Aeronautical Radio Incorporated*, MARK 2 Aircraft Integrated Data System (AIDS MARK 2) ARINC Characteristics 573-7*, December 2, 1974.

6. Aeronautical Radio Incorporated, *Flight Data Acquisition And Recording System ARINC Characteristic 717-3 (and Supplements 1-8 January 15, 1988)*, March 27, 1981.

7. Aeronautical Radio Incorporated, *Mark 33 Digital Information Transfer System (DITS) ARINC Specification – 429-10*, March 16, 1987.

APPENDIX A – STANDARD EQUATION CONVERSIONS

Binary Coded Decimal (BCD) Conversion

BCD is a method of encoding decimal (base 10) numbers in binary form typically using 4 bits to store each decimal digit. The lower order bits represent the lower order digits. For example, the decimal number 42 is encoded as 0100 0010 as opposed to 101010 in binary. In some cases, fewer than 4 bits may be used to represent a restricted range of digits. For example, two bits may be used to represent the Tens of Days digit in a date (i.e., digits 0 through 3).

BCD numbers are defined as BCD followed by the number of bits per group. For example, a 12 bit BCD stored as 4 groups of 3 bits would be defined as "BCD 3333". When the number of bits per group are not specified, the default "BCD" only, refers to 4 bits per group.

Teledyne Synchro Conversion

The following table describes the conversion of a raw count value to an angle value using the Teledyne Synchro conversion. The appropriate conversion equation is shown in the right-hand column for the different raw count value ranges shown in the left-hand column.

Abbreviations used in the table:

count: raw count value

n: number of bits used to represent raw count value (e.g., 10 or 12)

θ: resulting angle from the conversion

base: $\frac{2^n}{8}$

Raw Count	**Angle**
$0 \le count < base$	$\theta = Tan^{-1}\left(\frac{count}{base}\right)$
$base \le count < 2 \cdot base$	$\theta = Tan^{-1}\left(\frac{1}{2 - \frac{count}{base}}\right)$
$2 \cdot base = count$	$\theta = \frac{\pi}{2}$
$2 \cdot base < count < 3 \cdot base$	$\theta = Tan^{-1}\left(\frac{1}{2 - \frac{count}{base}}\right) + \pi$
$3 \cdot base \le count < 5 \cdot base$	$\theta = Tan^{-1}\left(\frac{count}{base} - 4\right) + \pi$
$5 \cdot base \le count < 6 \cdot base$	$\theta = Tan^{-1}\left(\frac{1}{6 - \frac{count}{base}}\right) + \pi$
$6 \cdot base = count$	$\theta = 3 \cdot \frac{\pi}{2}$
$6 \cdot base < count < 7 \cdot base$	$\theta = Tan^{-1}\left(\frac{1}{6 - \frac{count}{base}}\right) + 2 \cdot \pi$
$7 \cdot base \le count < 2^n$	$\theta = Tan^{-1}\left(\frac{count}{base} - 8\right) + 2 \cdot \pi$

Fairchild Synchro Equation

The following equations describe the conversion of a raw count value to an angle value using the Fairchild Synchro conversion.

count: raw count value

n: number of bits used to represent raw count value (e.g., 10 or 12)

θ: resulting angle conversion

$$high = \left\lfloor count \cdot \frac{1}{2^{(n-2)}} \right\rfloor \cdot 2^{(n-2)}$$

$$low = count - high$$

$$\theta = \left(\tan^{-1}\left(\frac{low}{2^{(n-2)} - low} \right) \cdot \frac{2^n}{2 \cdot \pi} + high \right) \cdot \frac{360}{2^n}$$

in the case of $low = 2^{(n-2)}$,

$$\theta = \left(2^{(n-2)} + high\right) \cdot \frac{360}{2^n}$$