TABLE OF CONTENTS

APPENDICES

FIGURES

TABLES

1.0 INTRODUCTION

Document Title: *Software Loading in the Shop Environment*

This document defines information required to load data into aircraft components while not installed aboard an aircraft. The objective is to address all levels of data as related to component maintenance and repair within a shop environment.

Many legacy aircraft systems use the ARINC Report 615 data loading communications protocol for loading components. Such systems use one common protocol for shop loading and aircraft loading. Shop loading is performed in a maintenance environment while the component is not installed on an aircraft. Aircraft loading is performed while the component is installed on an aircraft. The same data loading equipment can be used at the aircraft as well as in the off-aircraft shop environment.

With the advent of networked avionics systems integrated into aircraft (such as Aircraft Data Networks that use ARINC Specification 664, Part 7), it has become apparent that many of the avionics units cannot be loaded independent of the aircraft network system in which they are designed to operate. Furthermore, the information required to understand and develop solutions for loading data directly into networked components has not been available to the operators.

Airlines recognize the need for this guidance since they operate aircraft that use Aircraft Data Networks (ADN). Airlines, Maintenance Repair Organizations (MRO), and other parties need the ability shop loading of all types of data into aircraft components which includes both Aircraft Controlled Software (ACS) and Hardware Controlled Software (HCS).

1.1 Scope and Purpose

Airlines require the capability to load software into avionic components within the component shop maintenance environment. They need the ability to load all types of software as required to perform maintenance of components at all three Component Maintenance Levels (CML) as defined in **ARINC Report 663:** *Data Requirements for Avionics Component Maintenance* which are indicated as CML-1, CML-2, and CML-3.

Operators of avionics components should have the ability to develop a data load solution for their chosen components as an alternative to purchasing supplier produced equipment. History has shown that there can be a large cost to operators associated with acquiring and maintaining ground equipment when its functional fit, or interface characteristics are not documented as an open standard and are instead treated as proprietary. Information and specifications for communications between aircraft components and data loading equipment is essential for having cost effective support during the entire life cycle of an aircraft.

Specifications and requirements for data loading at all component maintenance levels should be included in the Technical Support Data Package (TSDP) as defined in **ARINC Report 625:** *Industry Guide for Component Test Development and Management*. This includes information required to design and implement a data load solution specific to a single component or family of components.

Data loading instructions should also be included in the Component Maintenance Manual (CMM) for each component as required during the performance of shop environment testing and maintenance.

1.0 INTRODUCTION

The loading equipment in the CMM may be specified as specific OEM or third party supplied items, however, each item should always include the words “or equivalent” as a qualifier for each piece of data loading equipment listed. This enables the usage of operator approved data loading equipment.

The focus of this standard is to list and describe the types of information that should be provided as part of the avionics component documentation and test support data package.

The current generations of aircraft usually employ airborne integrated data networks for communications between onboard avionics system components. Components that communicate within switched network architectures require clear definitions for shop environment loading when removed from the aircraft network environment..

There are no current ARINC standards for loading data to components, SRUs, subcomponents, or component parts within at the shop environment.

This document discusses the shop environment processes associated with various types of data loading. The types of software that is commonly loaded into components is indicated. This document defines three levels of load processes performed within the shop environment. Several terms that are associated with data loading are defined in the first chapter; many of these terms should be understood in order to comprehend some of text that follows in later sections.

This document exposes the type of information required to develop solutions for shop environment loading, including component specific bus interface protocol specifications and software communications protocol specifications. If vendors and OEMs choose to adhere to these requirements, it will enable custom shop loaders to be developed by operators or their agents as required. Operators will be able to maintain components throughout the lifecycle of their aircraft fleets.

Figure 1-1 shows how a gap analysis reveals the need for shop environment loading standards within the existing standards for aircraft system design and maintenance.

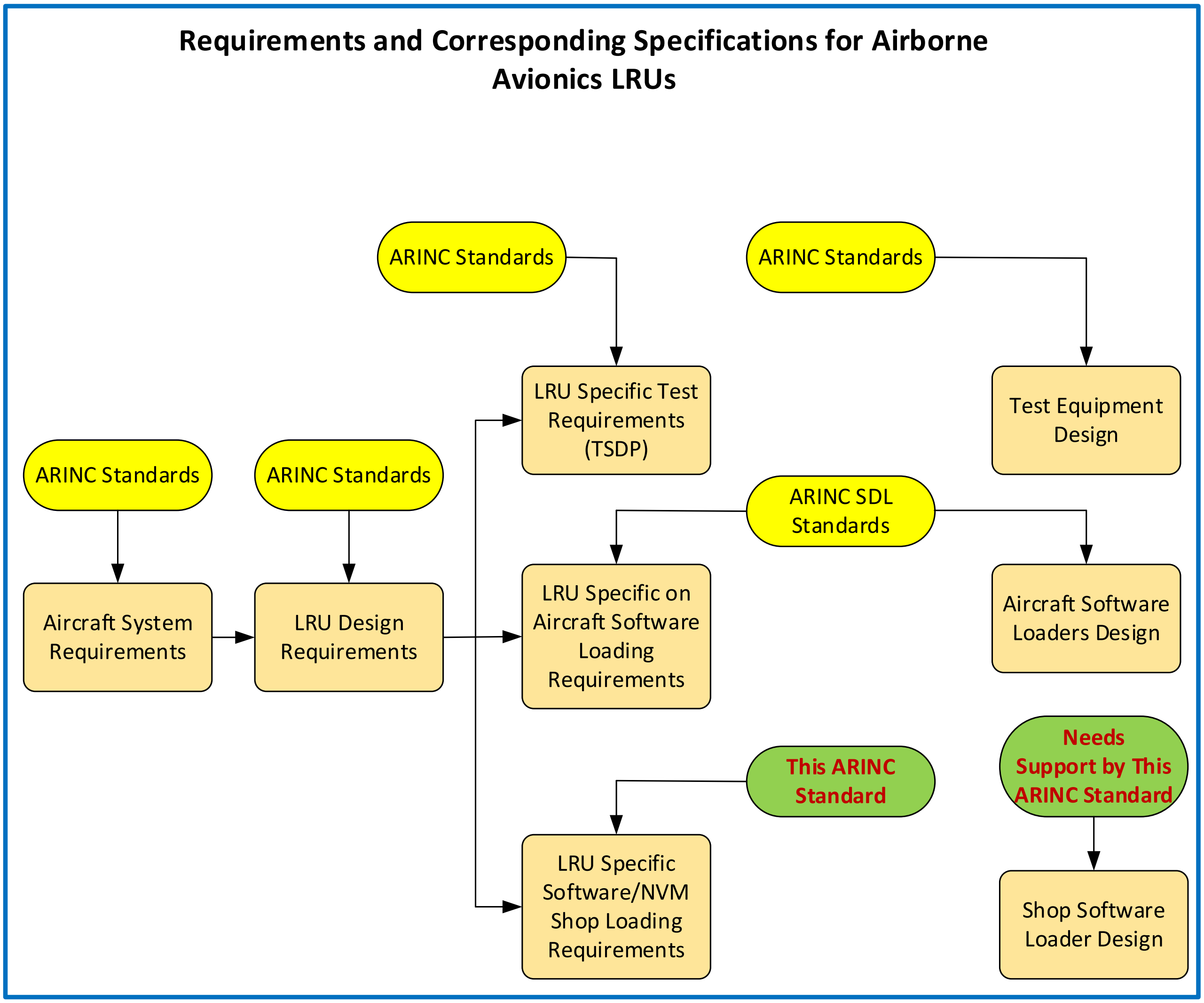


Figure 1-2 – Requirements and Corresponding Specifications

1.2 Term and Phrase Definitions

The following terms and phrases should be understood before reading further sections of this document. Having some familiarity with each of these will provide the reader with the language overview of this subject “Software Loading in the Shop Environment”.

Within the industry, several terms that reference the same object or idea have been historically used. It is important to understand when terms have equivalent definitions in order to avoid confusion while digesting the information presented in this document, and its application.

1.2.1 Shop Environment

Also called shop maintenance environment. The maintenance environment where aircraft avionics components are repaired and tested while removed from the aircraft. The subject of this document is loading software to components withn this environment.

1.2.2 Data versus Software

The words *data* and *software* are used interchangeably, and considered to mean the same thing within this document. While in the state of storage and transfer, they are both alike as compositions of digital information. However, they are sometimes interpreted as different items, each having an exclusive purpose. This has led to some confusion within the industry. The word data is synonymous with the word software within the context and purpose of software distribution, loading, and management, and therefore are used herein to have the same role. Data is digital information that may or may not be used as a sequence of program instructions by the loadable target, component, or part. For example, a digital package such as an Aircraft Controlled Loadable Software Part (ACLSP) usually contains data that is intended to be used as stored information combined with data that is intended to be used as a sequence of program instructions.

Software and data sometimes have other names that imply their usage and volatility within a component. The specific usage of such names is sometimes inconsistent between manufacturers. For example, firmware is a term that often indicates software or data stored in the non-volatile memory of programmable parts. (i.e., retains data when electrical power is removed). Firmware is sometimes executed during power on bootstrap operations for components. Some firmware may only provide services to higher-level software. Firmware such as the program of an embedded system may be the only program that will run on the system and provide all of its functions. In any case, the terms data, software, firmware, or program are sometimes used interchangeable in different electronic industry situations.

1.2.3 Loading

The word *loading* is used to mean the transfer of software (data) from a containment source to the active memory of an aircraft component.

In most cases, when loading is specified in regard to software type, environment type, and levels of loading, is may be assumed that data may flow both ways. That is data may be uploaded to the component or data may also be downloaded from the component.

Since loading to the component is the most common activity in the context of maintaining aircraft systems, loading will be used unless technical aspects are addressed which are specific to uploading or downloading.

1.2.4 Data Loading Verses Software Loading

Several terms are used in the electronics and avionics industry to describe moving or copying data from a source to a destination. In commercial aviation, Data Loading has been the most common term for moving data from a source of data to a component while installed on an aircraft (aircraft environment), or within the shop maintenance environment (shop environment). Software loading is becoming the preferred industry term going forward; however it remains synonymous with the term data loading.

1.2.5 Aircraft Environment Loading

Data loading can occur within the aircraft environment or the shop environment. Loading at the aircraft environment refers to loading data to the component while the component is installed within its intended system on board the aircraft. Aircraft Environment Software Loading is an equivalent term.

1.2.6 Shop Environment Loading

Loading at the shop environment refers to loading data while the component is not installed within any aircraft system or on board an aircraft. This usually occurs within the shop environment. Some load processes may be performed in other ground support areas such as for the support of pre-loading Aircraft Controlled Software into stocked components.

COMMENTARY

The terms *Aircraft Environment Loading* and *Shop Environment Loading* refer to the environment and method in which software is loaded, and should not be confused with the terms Aircraft Controlled Software (ACS) and Hardware Controlled Software (HCS). These terms refer to the process of software type, version, and part number control, irrespective of software loading environment and method.

1.2.7 Loading Aircraft Controlled Software (ACS)

Aircraft Controlled Software (ACS) is assigned an aircraft part number, managed as an aircraft part, and can be loaded at the aircraft environment or the shop environment. Aircraft controlled software is managed as described in ARINC 667A.

It is important to understand the difference between Aircraft Controlled Software and Aircraft environment Software loading. Aircraft Controlled Software is a type of software whereas Aircraft environment Software loading is the type of action which is to load software to components and systems while installed on the aircraft. For example, Aircraft Controlled Software can be Shop Environment Software loading.

1.2.8 Loading Component Test Software (CTS)

CTS is software that is used within the shop environment as part of a component test or diagnostic function. CTS does not remain resident in the component during service.

1.2.9 Loading Hardware Controlled Software (HCS)

Hardware Controlled Software, with few exceptions, is loaded within the shop environment. It has various means of loading and control procedures that are defined by various OEM’s and industry electronic standards, instead of aircraft industry standards like those used for aircraft controlled software loading.

Within the component shop maintenance environment, where shop environment loading occurs, other electronic industry terms have been used in place of loading data or loading software, such as programming a device, upload data, upload software, upload files, download data, copy data, transfer data, etc. These are used to describe copying data to a component, subcomponent, or programmable parts that are used within the component. This document intends to address all forms of airborne software loading that are required for all levels of component maintenance and repair.

1.2.10 Aircraft Support Data (ASD)

Aircraft Support Data is aircraft software and data that is not certified under Part 25 or operationally certified under Part 121. This includes some cabin In-Flight Entertainment software (IFE). For more information regarding aircraft support data, refer to **ARINC 676**.

1.2.11 Airborne Software

Airborne Software is all software and data that are resident in any aircraft system, component, or part within a component that remain aboard an aircraft during flight.

1.2.12 Component

In the context of this document, component refers to 'avionics component' which is a self-contained avionic hardware assembly that can be installed on an aircraft in a manner that facilitates the removal from and replacement to the aircraft.

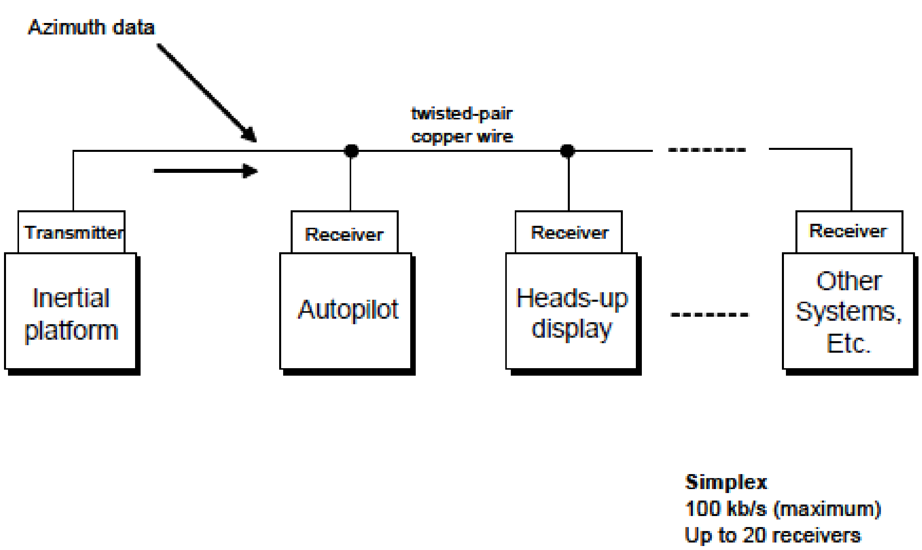
A component is generally synonymous to a Line Replaceable Unit (LRU). It is commonly used interchangeably with LRU which is a component that can be removed from and replaced to the aircraft in a relatively short period of time.

Component shop maintenance facilities require the ability to load data into components as a whole and load data into a component's sub parts such as Shop Replaceable Units SRU's, sub-components, and basic parts or devices.

1.2.13 ARINC 429 Avionics Data Bus

The majority of legacy aircraft use the ARINC 429 bus architecture for flight systems communication. In ARINC 429, a twisted pair must link every LRU, which is to receive a data signal from a source LRU. The point-to-multi-point property of ARINC 429 means that the avionics system must include an ARINC 429 bus for each communication path. In a system with many end points, point-to-point wiring requires much more wire, which can lead large wiring harnesses and adds significant weight to the aircraft. The details of the ARINC 429 bus are discussed in section 7.

Figure x.x is an example of this bus architecture.

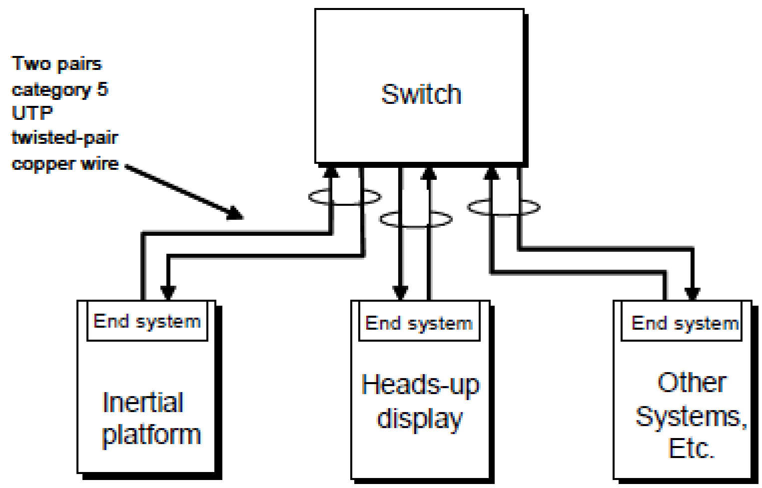


a) ARINC 429

1.2.14 Avionics Full-Duplex Switched Ethernet Network (ARINC 664 P7)

The majority of commercial aircraft have evolved from using the ARINC 429 Bus as the primary flight systems communications bus to ARINC 664 P7. This is the architecture of avionics systems where components share and access data using the ARINC 664 Part 7. The details of the ARINC 664 Part 7 bus are discussed in section 7.

Figure x.x is an example of this bus architecture.



Full duplex
100 Mb/s (maximum)
Number of connections:
governed by number of switch ports

b) AFDX

1.2.15 Published Open Document

In the context of this document, open document implies that the specification document not only exists, but is also available and procurable by stakeholder operators and possibly by designated 3rd part development bodies.

1.2.16 Other Terms

- Portable Data Loader (PDL)
- Automated Test Equipment (ATE)
- Operators – Term for aircraft users such as airlines, freight air carriers, and charter services.
- Component Maintenance Level 1 (CML-1)
- Component Maintenance Level 2 (CML-2)
- Component Maintenance Level 3 (CML-3)
- Component Software Loading (CSL)
- Component Software Loading Level (CSLL)
- Component Software Loading Level 1 (CSLL-1)
- Component Software Loading Level 2 (CSLL-2)
- Component Software Loading Level 3 (CSLL-3)
- Component Software Load Level (CSLL) specification – All technical information required to develop and implement a solution for loading a

component at one or more Component Software Loading Levels (CSLL-1, -2, -3).

- Component Software Load Level (CSL) standard – Standard written against a component specifying one or more CSLL specifications.

1.3 Document Conventions

ARINC reports are voluntary standards intended to ensure interchangeability and interoperability between equipment, independent of manufacturer or airframe.

In this standard, the following terms carry key significance:

Term	Usage
shall	Identifies features required to meet the minimum level of compatibility intended by this standard.
must	Obligation, no other choice
should	Used to recommend approaches to optimize transactions and management.
will/is/does	Used to express a statement of fact based on other requirements.
may	Used to express an optional capability or choice.

1.4 Reference Documents

In all cases, it is intended that the latest version of the referenced document applies. As well as referring to the terms and definitions used in this ARINC Report, the reader should ensure that the most current version of all documents, websites, and contacts are referenced.

This report references other industry guidance documents, including:

- **ARINC Specification 429**: *Digital Information Transfer System (DITS)*
- **ARINC Characteristic 603:** *Airborne Computer Data Loader*
- **ARINC Report 615:** *Airborne Computer High Speed Data Loader*
- **ARINC Report 615A:** *Software Data Loader Using Ethernet Interface*
- **ARINC Report 625:** *Industry Guide for Component Test Development and Management*
- **ARINC Report 663:** *Data Requirements for Avionics Component Maintenance*
- **ARINC Specification 664:** *Aircraft Data Network, Part 7 - Avionics Full-Duplex Switched Ethernet Network*
- **ARINC Report 665:** *Loadable Software Standards*
- **ARINC Report 667A:** *Guidance for the Management of Field Loadable Software*
- **ARINC Report 668:** *Tooling and Test Equivalency*
- **ARINC Specification 825:** *General Standardization of CAN (Controller Area Network) Bus Protocol For Airborne Use*
- **ARINC Report 826:** *Data Loading Using CAN Bus*
- **ARINC Report 827:** *Electronic Distribution of Software*
- **ARINC Report 835:** *Guidance for Security of Loadable Software Parts using Digital Signatures*
- **ARINC Specification 838:** *Loadable Software Part Definition Format*

- **ARINC Report 847:** *Product Development Guidance for Maintainability and Testability (PDMAT)*

2.0 BACKGROUND

2.1 Shop Maintenance Requirements

Airlines need the ability to load all types of airborne and test software within the shop environment. Loading software is integral to testing and repair of components at all three Component Maintenance Levels (CML) as defined in **ARINC Report 663:** *Data Requirements for Avionics Component Maintenance* in which are indicated as CML-1, CML-2, and CML-3.

Therefore, the airline's maintenance processes and functions within shop maintenance require the ability to load all types of airborne software, which include:

- Aircraft Controlled Software (ACS)
- Hardware Controlled Software (HCS)
- Component Test Software (CTS)

There have been cases where the information required to develop solutions for loading software was not openly available to operators.

2.1.1 Firmware procurement

Additionally, hardware controlled software or firmware was not obtainable from the supplier of the component. This was particularly evident in cases of programmed parts. A programmable part (IC) and the firmware that is loaded to it are not procurable as two separate items.

2.2 Component loading technical specification

Specifications should be available for the following:

- Load software to the component as a whole
- Loading software to any subcomponents or sub-assemblies
- Loading software to component programmable parts, devices, and ICs

To date, there has been very few software loading protocols published for loading components and their sub parts. Airlines have had difficulty obtaining the technical information required for some levels of loading. Levels of loading are discussed in chapter 3.0.

2.3 Component Part and Software (Firmware) Availability

Additionally, hardware controlled software or firmware was not obtainable from the supplier of the component. This was particularly evident in cases of programmable parts. A programmable part (IC) and the firmware that is loaded to it have not been procurable as two separate items.

When a programmable component part is replaced, the software (firmware) that is used to program the new part should be procurable as an independent item. The un-programmed hardware part or device should be listed in the CMM IPL. If there is software needed for that one part, it should be listed as a separate IPL item.

Until now, it has been typical for vendors to list a pre-programmed device as one single part number in the IPL. This prevents the shop from using multiple resources to acquire the un-programmed device. The vendors are also tasked with having to stock pre-programmed devices for up to 30 years.

Many circuit designs use the same programmable part for several different functions. A circuit card may use ten parts (ICs) that have the same IC part number. The function definition of each IC is established by the firmware loaded into the IC.

Therefore, by having the software procurable as a separate item, the programmable IC parts may be stocked as one single part number to serve as replacement for several use cases.

Airlines have had much difficulty obtaining pre-programmed parts from vendors in the past, especially for older components that are no longer supported very well.

2.4 Exemplary Case: Shop Environment Loading of ACS and CTS

One notable issue regarding shop environment loading is described below in section 2.3.1 and section 2.3.2. This is a Component Software Load Level 1 (CSLL-1) process, which is defined in section 3.3.

This case is one in which component shops need the ability to load component test software and aircraft controlled software which uses the same loading protocol and component resources within the aircraft environment and the shop environment. This was possible for legacy aircraft using equipment that conformed to the ARINC 615 loading standard using the ARINC 429 bus. However, for current aircraft that use the ARINC 664 P7 bus, there are no ARINC standards for loading the component when removed from the aircraft.

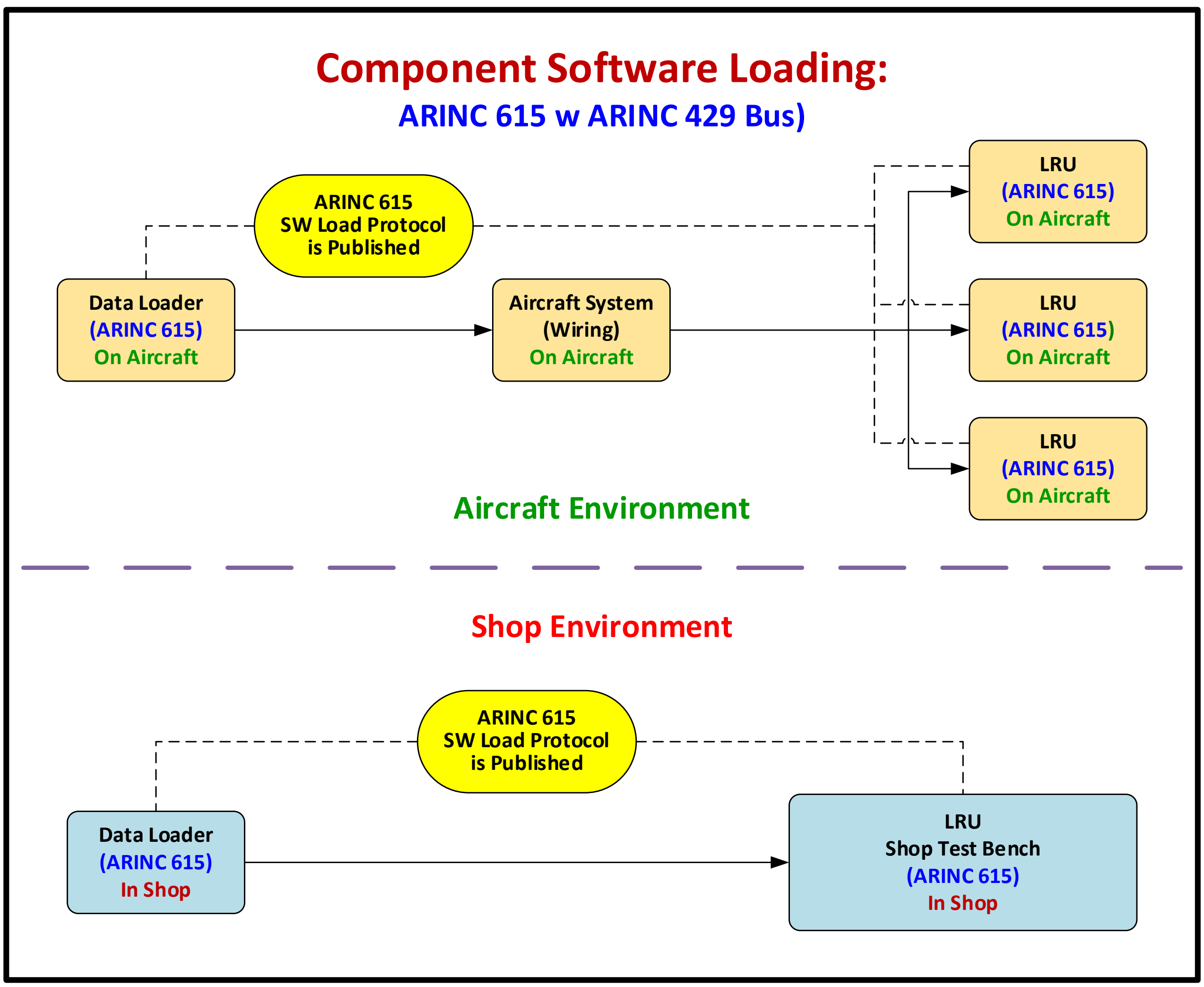


Figure 2-1 – Loading ACS Into LRU: Legacy Aircraft

2.0 BACKGROUND

2.4.1 Shop Environment Loading – ARINC 615 (ARINC 429)

For aircraft that mainly use ARINC Specification 429 bus data loading equipment, the communications protocol is specified within the open source document ARINC Report 615. Many organizations have used the information contained in this document to create custom data loading solutions for shop maintenance.

It is shown in Figure 2-1 that the common ARINC Report 615 protocol follows through the entire aircraft data load path up to the software load target LRU. When the removed component LRU is in the shop maintenance environment, data loading can be accomplished using the same protocol, and often the same Data Loading equipment, that is used in the aircraft environment. It has been common place to use ARINC 615 compliant Portable Data Loaders and Airborne Data Loaders within the shop environment.

Sometimes it is advantageous to embed the ARINC 615 data load protocol and functionality into test equipment such as Automated Test Equipment (ATE) used in the shop environment, thus avoiding the use of external loading equipment intended for the aircraft environment. Shop component test equipment are often required to have data loading capability.

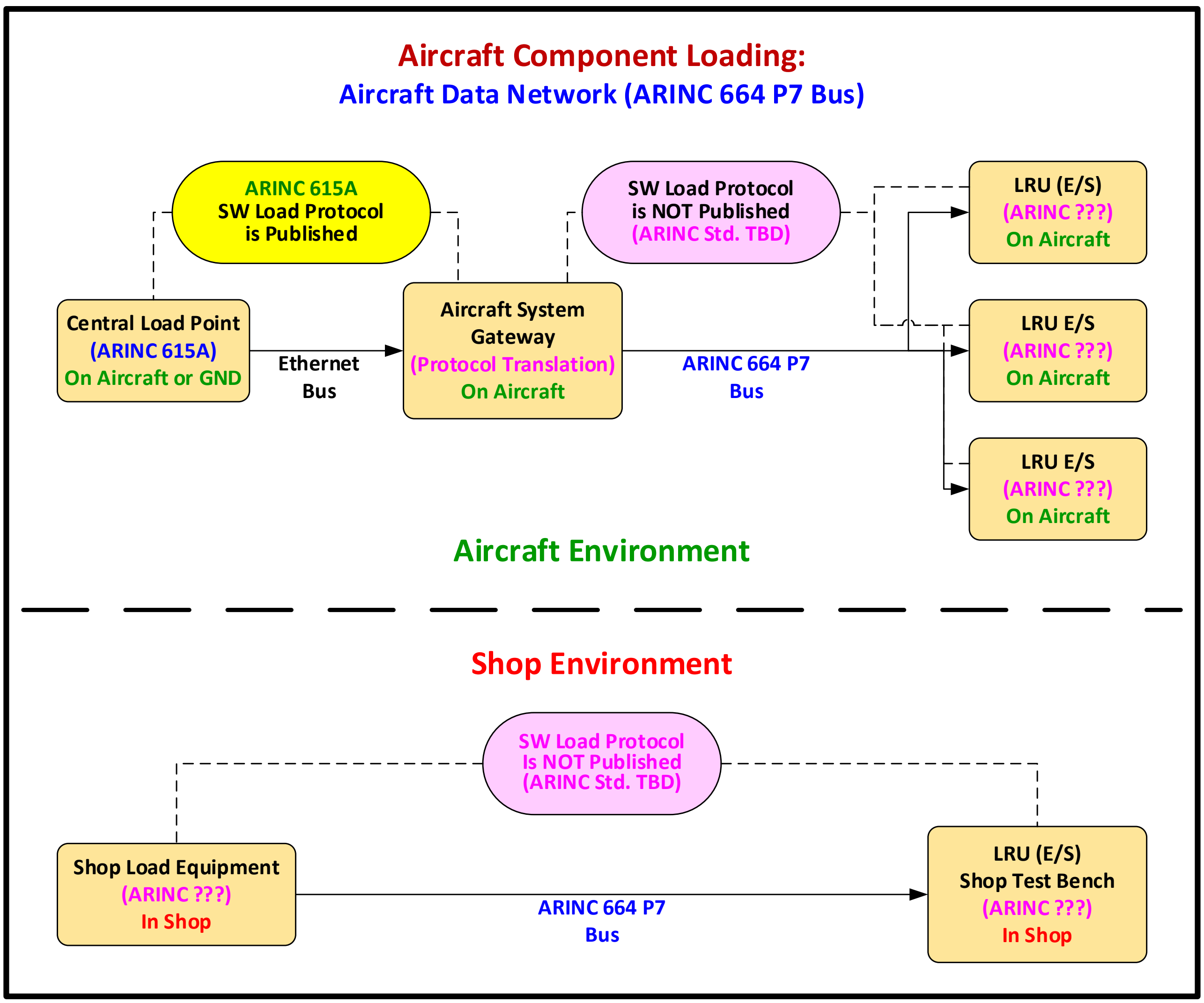


Figure 2-2 – Loading ACS Into LRU: Aircraft with Aircraft Data Network

2.4.2 Shop Environment Loading Issue – ARINC 664 Part 7

As aircraft have moved towards using network architectures for avionics communications, an issue became apparent and of concern regarding component data loading within the shop environment.

Figure 2-2 shows how the ARINC 664 P7 bus is used for communications between components and for software loading.

Within ARINC 664 P7, Virtual Links (VL) are used to provide communications between aircraft components or End-Systems (E/S).

A Virtual Link defines a unidirectional (logical) connection from one source End-System to one or more destination End-Systems using switches.

Some switches incorporate an embedded E/S. The purpose of the switch E/S is to provide a means for functions that are external to the network in order to communicate with the switch. For example, data loading and network management functions communicate with the gateway (translational switch) via the gateway’s embedded E/S.

The communications protocol used between data loading equipment and the switch E/S is specified in an open source document that is available to the end user. A common example is the ARINC Report 615A standard for loading data using the Ethernet bus.

However, communications between the gateway E/S and other E/S components has not been specified in any open source document to the operator. This makes development of data loading equipment for use within the shop environment impossible for operators. Developing data loading solutions to be embedded within shop component test equipment are also precluded.

Solutions that may be available are based on proprietary information that is not openly accessible as an open source document or as part of the ARINC 625 Test Support Data Package (TSDP) for a component.

The practice of using aircraft equipment as shop test equipment, such as a gateway device, would likely be cost prohibitive. Using aircraft equipment also does not provide for the development of embedded data load solutions for test equipment.

This further drives the airline’s need for data loading specifications regarding aircraft data network End-System components.

3.0 LEVELS OF COMPONENT SOFTWARE LOADING

Component software loading processes may be categorized into logical levels that represent the data loading functional access to a component. Different levels of access are required are related to component loading, sub-assembly loading, or component part loading.

CSLL	Component Communications Interface	Aircraft / Shop Environment	Type of Software Loaded	Component Physical State	Hardware Bus Examples
CSLL-1	Primary load Interface	Aircraft Environment, Shop Environment	ACS, ASD	Assembled	ARINC_429, ARINC_664_P7
CSLL-1	Aircraft Maintenance Bus Interface	Aircraft Environment, Shop Environment	ACS, ASD	Assembled	Not Yet Implemented (SDL Wish List)
CSLL-2	Alternate load Interface	Shop Environment, EEC	HCS, ASD	Assembled	ARINC 429, RS-422, RS-232, JTAG
CSLL-3	Alternate load Interface	Shop Environment	HCS	Dissassembled	JTAG, Parallel Bus, Chip Programmer

Table 3-1 - The Levels of Software Loading

3.1 Component Software Loading (CSL)

CSL is a general term that includes all three levels of Component Software Loading.

3.2 Component Software Loading Level (CSLL)

CSLL is a general term for the process level of loading airborne software into the component.

3.3 Component Software Loading Level 1 (CSLL-1)

Component Software Loading Level 1 is the function where software is transferred into a target component using the component's primary load port. This is the bus and port intended for data loading in the aircraft environment.

Software can be loaded in either the aircraft environment or the shop environment using CSLL-1.

The most common type of software loaded using CSLL-1 is Aircraft Controlled Software (ACS), which can be done within either the aircraft environment or the shop environment.

When in the shop environment, Component Test Software (CTS) type of software is often loaded to support shop maintenance functions.

The CSLL-1 level process of loading Aircraft Controlled Software (ACS) into a component can occur under the following conditions:

- The component is in an aircraft environment; installed on an aircraft or within an aircraft system.

- The component is in a shop environment; not installed on an aircraft or within an aircraft system.

The CSLL-1 level process of loading Component Test Software (CTS) into a component can occur under the following conditions:

- The component is in a shop environment

COMMENTARY

The term aircraft system in relation to CSLL refers to a combination of two or more interconnected components that work together. An aircraft system generally exists aboard an aircraft, but may be fully or partially constructed external of the aircraft.

When a component is external of the aircraft, it must be software loadable at the CSLL-1 without the use of any other aircraft system components.

3.4 Component Software Loading Level 2 (CSLL-2)

CSLL-2 is the level of load process in which software classified as Hardware Controlled Software (HCS) is loaded into an aircraft component. An HCS part number reference is controlled within the Component Maintenance Manual (CMM) and Illustrated Parts List (IPL) or Service Bulletin (SB). The process of loading HCS class of software into a component requires a change to the component's part number reference.

No disassembly of the component is required to perform CSLL-2 software loading. The component may facilitate connectivity to loading equipment through dedicated pins within its standard electrical connectors, or by auxiliary maintenance connectors. Some auxiliary connectors may require the removal of a protective cover plate or cap for access during loading.

The CSLL-2 level process of loading an HCS into a component can occur under the following conditions:

- The component is in a shop environment; not installed on an aircraft or within an aircraft system.
- For some exceptions, the component may be installed on an aircraft. For example, some Electronic Engine Controllers (EEC) can be loaded in the shop environment or while installed on the aircraft. HCS of this type is often referred to as Hardware Controlled Loadable SW Part (HCLSP)

COMMENT

The word "Loadable" in HCLSP refers to loading software while the component is installed on the aircraft. It is loaded in the aircraft environment.

In either case, the component's part number is changed by performing the CSLL-2 load procedure. The identifying tag on the component must be changed to reflect the new component part number.

3.5 Component Software Loading Level 3 (CSLL-3)

CSLL-3 is the level of load process in which software classified as Hardware Controlled Software (HCS) is loaded into an aircraft component, its subcomponents, or parts. The HCS part number reference is controlled within the Component

Maintenance Manual (CMM) and Illustrated Parts List (IPL) or Service Bulletin (SB). The process of loading HCS class of software into any part of a component requires a change to the component's part number reference. This class of software is sometimes referred to as firmware or resident software.

In contrast to the CSLL-2 level, the CSLL-3 level load process often requires some disassembly of the component to access required connectivity to the component, sub-component, circuit card, or component part.

Examples of the CSLL-3 level load process are as follows:

- The load process requires accessing Joint Test Action Group (JTAG) connectors or other maintenance connectors on Shop Replaceable Modules or other circuit cards to load software into Integrated Circuits (IC) such as memory chips, Field Programmable Gate Arrays (FPGAs), microcontrollers, and other programmable ICs.
- Shop Replaceable Modules or other circuit cards are completely removed from the component and software is loaded using appropriate card programming equipment.
- Integrated Circuits (IC) such as memory chips, FPGAs, microcontrollers, and other programmable ICs must be removed from the circuit card and inserted into a programmer.

The CSLL-3 process of loading HCS into any part of a component can occur under the following conditions:

- The component is in a shop environment and not installed on an aircraft or within an aircraft system.

In any case, the component's part number is changed by performing the CSLL-3 HCS load procedure.

COMMENT

It should be understood that CSLL-2 and CSLL-3 are defined by the environmental requirements associated with the load process. HCS is loaded in the shop because it is not considered to be an aircraft controlled process. Components are generally not disassembled in the aircraft environment, or anywhere else outside of the shop environment.

The CSLL-2 process in general includes the function of loading the component as a self-contained unit. In some cases, it may be possible to do this in the aircraft environment with specific loading equipment. This means that software loaded at CSLL-2 could potentially be a HCLSP. However, this is usually not standard procedure for flight avionics equipment because doing so changes the component part number which implies a modified component. This changes the aircraft hardware configuration as it was not aircraft controlled software. Changing HCS to modify a component often requires a shop level return to service test as part of certification for service.

The component design could make it possible however, to upload from one accessible connector, software to various subcomponents and parts within the component, without any disassembly, in an aircraft environment. This still changes the component part number. The current process of managing the aircraft software configuration is simplified by not having components that chang identity while in the aircraft environment. This philosophy is reflected in ARINC 667A.

4.0 SCOPE OF DATA LOADING SPECIFICATION REQUIREMENTS

Component data loading specification requirements for the shop environment are described in relation to the CSLL and are dependent on the availability and comprehensiveness of the software loading specification as an open standard or as a document provided by the supplier.

In some cases, the supplier may require operators to comply with a non-disclosure agreement that protects their intellectual property (IP).

The following sections describe requirements that are dependent on the availability of the software loading specification as an open standard or as a document provided by the supplier.

For each component that requires airborne software loading as part of CML-1, CML-2, or CML-3, the software load specifications may be included in the CMM, or within an external document source referenced by the CMM. The load specifications should be included in the ARINC 625 TSDP for the component.

The load procedures should be included within the CMM, or within an SB if applicable. The load procedures are written to use the load tool that conforms to the load specifications.

4.1 CSLL Specific Load Specifications

Each software load specification document, or document section, should be developed in regard to one of the three component software load levels, CSLL-1, CSLL-2, or CSLL-3. The document should clearly state the intended CSLL applicability.

There are cases when …. There are two conditions of consideration

4.1.1 The CSLL Specific Standard is Available

If the CSLL protocol is already fully documented in an open document such as an ARINC standard, then that is the only reference required. If there are any deviations from, or exceptions to the existing standard, then they must be specified, documented and made available by the supplier.

In general, when the CSLL conforms to a standard, there is still additional information required to perform software loading for the specific component. All parameters required for communication with the specific component such as network addressing, timing, and configuration parameters must be made available as a procurable document. All hardware component specific signal requirements for software loading must be indicated.

One example of a published CSLL-1 standard is ARINC Report 615 that is used to load FLS using the ARINC 429 bus. However, in addition to the standard document information, other supplemental parameters must also be included, such as requirements to put the component into the loading state.

Another example would be ARINC Report 615A when it is used to load directly to a component that directly accepts the ARINC 615A Ethernet protocol without any other intermediate special aircraft component such as a gateway. In this case, a CSLL-1 specification would include all necessary component specific software and hardware connection information required to allow component loading using ARINC 615A equipment while not installed aboard an aircraft.

4.1.2 The CSLL Specific Standard is Not Available

If the CSLL specification is not openly available to the end user, then the OEM should develop and prepare a document appropriate for open distribution. The technical specification for a component's CSLL may be generated as a general standard when it applies to a family of components, or as a document specific to a component and its component software load level.

If any part of the data loading communications specification uses another open standard, then that document may be referenced as an external standard within the data load specification. However, care must be taken to include all information regarding any variation from the published specification. When multiple feature and mode selections exist within the external standard, the appropriate selections implemented by the specification must be indicated.

When appropriate, parts of external standards should be embedded into the CSLL specification to clarify the functional context of its use within the load specification.

Examples of openly available supplemental standards used within CSL standards are as follows:

- ARINC 429
- ARINC 664, Part 7
- Ethernet
- Internet Protocol (IP)
- User Datagram Protocol (UDP)
- Trivial File Transfer Protocol (TFTP)
- RS-485 (TIA-485)
- RS-422 (TIA-422)
- RS-232 (TIA-232)
- Joint Test Action Group (JTAG)

5.0 TYPES OF COMPONENT SOFTWARE LOADED IN THE SHOP ENVIRONMENT

The principal types of software that are loaded into component LRUs in the shop environment are as follows:

- Aircraft Controlled Software (ACS)
- Hardware Controlled Software (HCS)
- Component Test Software (CTS)
- Calibration Data
- Aircraft Support Data (ASD)

5.1 Airborne Software

All software, that is installed in any component or device which is installed aboard an aircraft is considered airborne software. In the context of this document, the term *software* includes *data* and *firmware.* The control requirements for airborne software are dependent on the class of software as described in ARINC 667 and the applicable FARs. Each class is defined according to its criticality level, function and method of control.

Airborne software contains two primary types of software, Aircraft Controlled Software (ACS) and Hardware Controlled Software (HCS). This standard describes aspects and requirements regarding the process of loading all types of airborne software into components while in the shop environment.

A brief description of airborne software types is needed for this standard to accurately define shop loading specification requirements. For more information regarding control of airborne software, reference ARINC Report 667A: *Guidance for the Management of Field Loadable Software* (FLS).

Loading airborne software to components within the shop environment is required for maintenance capability at all three component maintenance levels, CML-1, CML-2, and CML-3. Performing at three component maintenance levels in turn require that the component environment have the capability of loading software at all three component software load levels, CSLL-1, CSLL-2, and CSLL-3.

5.2 Aircraft Controlled Software (ACS)

Aircraft Controlled Software (ACS) is also commonly referred to as Field Loadable Software (FLS) and is controlled as an aircraft part which is termed an Aircraft Controlled Loadable Software Part (ACLSP). ACS is identified and distributed as ACLSPs.

Aircraft Controlled Software (ACS) is a category of airborne software that can be loaded into components while they are installed on an aircraft and powered by the aircraft's system. This generally takes place as part of maintenance actions on the flight line or in the maintenance hangar.

ACS is loaded at CSLL-1, and consists of software required for in-flight functionality of the component.

Performing test procedures of a component within the shop environment often requires for component test software (CTS) to be loaded using a CSLL-1 process. This usually overwrites all or part of the ACS currently install in the component.

After performing a complete test of a component within the shop environment, the current configuration ACS must be loaded before the component is returned to service.

Sometimes, components are maintained to have the most current configuration of ACS flight software while in stock as part of operator maintenance process efficiency. Doing this saves valuable aircraft turn time by making it unnecessary to load the component after installation on the aircraft. However, the installed component must always be checked for proper software configuration before aircraft dispatch.

COMMENTARY

An Aircraft Controlled Loadable Software Part (ACLSP) has a part number reference that is controlled as an aircraft software part exclusive from the host component's hardware part number reference. The process of loading an ACS class of software part into a component does not change or affect the component's hardware part number.

Each unit of software has a unique part number that identifies it as an aircraft part, and represents its function and version. Each part is controlled in a similar way to that of all other aircraft hardware parts, using an authority document such as the Illustrated Parts Catalogue (IPC).

ACS generally resides in aircraft system components. Loading ACS into components usually takes place while the component is installed on the aircraft. However, loading ACS into components may also be required when they are not installed aboard an aircraft. Operator's reasons for using ACS level data loading to uninstalled components are as follows.

- The shop environment test and maintenance procedure may require diagnostic software to be loaded. Diagnostic software is often loaded through the same component loading mechanism that is used for operational software loading.
- The component is required to have operational software loaded (pre-loaded) before installation to the aircraft so as to avoid having to load software during time critical operational periods.

5.3 Hardware Controlled Software (HCS)

Hardware Controlled Software, with few exceptions, is loaded within the shop environment. It has various means of loading and control procedures that are defined by various OEM's and industry electronic standards, instead of aircraft industry standards like those used for aircraft controlled software loading.

Loading HCS in the shop environment is a CSLL-2 process, where no disassembly of the component is required, or a CSLL-3 process in which sub components and component parts are loaded with software.

In the context of loading HCS, especially as part of a CSLL-3 process, other electronic industry terms and phrases are used in place of *loading software*. Examples of these term phrases are *programming a device, upload data, upload* *software, upload files, download data, copy data, transfer data*, etc. These are used to describe copying data to a component, subcomponent, or programmable parts that are used within the component.

Avionics component shops require the ability to load HCS as CSLL-2 or CSLL-3 processes in order to be capable of performing maintenance at CML-2 and CML-3.

This document intends to address all forms of airborne software loading that are required for all levels of component maintenance and repair which includes

The method of identification and control of HCS is OEM dependent. The software's version and status is rolled into the component's part number or MOD status of the component. If the software changes, so does the component part number (or MOD status) shown on its ID plate.

COMMENTARY

> Sometimes a component will be identified by two numbers: a hardware part number and a component part number.
>
> The hardware part number represents the current state of the component's hardware without regard to its hardware controlled software.
>
> The component part number represents the combined hardware state and the hardware controlled software state of the component.

HCS is sometimes referred to as firmware or resident software. It includes data that is programmed into the component's basic parts or sub-assemblies. Software loading or programming specifications are required to perform Component Maintenance Level 3 (CML-3) procedures and some CML-2 procedures. For example, as a CML-3 function, basic parts such as Programmable Array Logic (PAL) devices and Field Programmable Gate Arrays (FPGA) must be replaced and then loaded with the correct data to function within the component or component's sub-assembly.

An example of a CML-2 procedure is to remove a Circuit Card Assembly (CCA) from the component and load software using the appropriate CCA programming equipment.

More information can be found in **ARINC Report 663:** *Data Requirements for Avionics Component Maintenance*.

The component's design may facilitate loading of HCS without disassembly by use of an additional maintenance port electrical connector or thru connections through its rack connector.

The special maintenance port connector is may be located within a removable access panel. However, the effort required to access such a port should be minimal.

When HCS is loaded as a CSLL-3 process, it is highly desirable that minimal disassembly of the component is required.

Types and nomenclature for various HCS include:

- Hardware Controlled Loadable Software Part (HCLSP)
- Firmware
- Resident Software
- JEDEC File
- Datasets

Some of the functions that require loading of this type of software are as follows:

- Loading data to individual programmable parts that have been replaced such as FPGAs, PALs, Microcontrollers, and Embedded NVM.
- Loading firmware data to circuit cards or other subassemblies.
- Loading a firmware or resident software as part of a testing maintenance function or to perform a hardware controlled software HCS upgrade to the component.
- Loading data into Line Replicable Modules (LRM) to facilitate testing and diagnostics.

It is highly desirable to load HCS with minimal disassembly of the component.

Typically, requirements for firmware loading are for repair and should include specific programming procedures, instruction steps to get access to the component, pin connections definition for program download.

5.4 Component Test Software (CTS)

In contrast to Airborne Software, Component Test Software is used for component testing in the shop environment. This type of software is never used when the component is in service.

Component test software is often loaded as a CSLL-1 process within the shop environment as an integral part of a component test procedure. Doing this usually overwrites the memory space containing the operational Aircraft Controlled Software. The test software is designed to test various functions of the component hardware.

When testing of the component if complete and successful, the appropriate aircraft controlled software is loaded before the component is returned to service.

Component Test Software (CTS) supports all three levels of component maintenance. CML-1, CML-2, and CML-3, and is defined as ground support test software or datasets that are temporarily loaded into the avionics components for the purpose of component testing, subcomponent testing, diagnostic testing, and fault isolation/troubleshooting.

Therefore, CTS designed for various levels and aspects of testing may be loaded at any one of the three component software load levels, CSLL-1, CSLL-2, or CSLL-3.

During maintenance procedures in the shop test software datasets may be temporarily loaded into the LRU, the fault history memory device may be accessed and read.

ACS Software – When testing of a component is finished, operational software is usually loaded as part of a complete return to service process. ACS software is always loaded at CSLL-1.

5.4.1 CTS for CSLL-1

Component Test Software is often loaded using the CSLL-1. Test software is transferred to the component's memory using the component's primary load interface.

Testing a component often requires that custom test software is first loaded into the component being tested. Test software identification and version is controlled by the Component Maintenance Manual (CMM), Illustrated Parts List (IPL) or Service Bulletin (SB). After testing and approval for return to service, the correct ACS is loaded into the component in the shop environment, or after installation to the aircraft. Sometimes the loading of ACS is required before return to service certification is complete.

Other types of non-flight ground support software are also required to be loaded using the component's ACS Load mechanism and protocol. Such software is used for shop testing and diagnostic troubleshooting, and includes:

- Test Software – Component RTS Testing
- Diagnostic Software – Troubleshooting and sometimes part of RTS Testing

5.4.2 CTS for CSLL-2

During a maintenance process, test software is sometimes required to be loaded using a process similar that used for loading HCS without any disassemble of the component. This can be software required to perform RTS testing, or diagnostic testing used for troubleshooting a malfunctioning component.

Some examples of loading CTS at the CSLL-2 are as follows:

- Loading diagnostic software that enables testing for troubleshooting a malfunctioning component
- ATE loads diagnostic software to component through maintenance port for automatic RTS test support
- Factory BITE Software (FBS)
- Resident Boot Program (RBP): A short program that needs to be loaded in the component for further software data loading
- Loading Diagnostic Software – Troubleshooting
- Functional Test Kernel (FTK): The primary function of the kernel of a component is to serve as mediating access to the avionics component resources including:
    - The Central Processing Unit (CPU) or microprocessor that is responsible for executing internal programs
    - Random Access Memory (RAM) that stores executable scripts and data
    - Input/Output (I/O) devices such as a BITE display and pushbutton input.
    - Kernels also provides methods for synchronization and communication among Interprocess Communication (IPC)

5.4.3 CTS for CSLL-3

It may be necessary to partially disassemble a component to gain access to internal component and sub component data load ports and connectors. This may require only the removal of the outer cover, or may require further disassembly by removing sub components. Sometimes subcomponent part may require removal.

Circuit cards, SRMs, or sub-assemblies may need to be removed in order to facilitate data loading or programming.

Replacement of programmable parts such as microcontrollers, FPGAs, FLASH, or EPROMs may require direct access or removal for programming.

Occasionally a part containing test software, such as a test EPROM, is required to be inserted to perform testing of the component.

5.5 Calibration Data

Calibration Data frequently needs to be loaded during replacement of some components and sub components. Examples of calibration data include:

- Cabin lighting data
- Sensors and transducers
- Radar gain and power data
- Passenger Address configuration data

5.6 Aircraft Support Data (ASD)

Aircraft Support Data is not loaded to components that are part of any aircraft flight control, flight critical, or flight safety systems.

6.0 COMMUNICATIONS PROTOCOL

To have a complete shop maintenance test support data package, it is crucial to have access to a clear definition of the communications protocol used to transfer data to and from an aircraft component as a stand-alone unit. This applies to all modes of loading all types of software. Aircraft components that will require off-aircraft data loading capability include LRUs (self-contained units) and Printed Wired Assemblies (PWAs) that are installed into card file rack assemblies.

Modes of a data loading communications protocol include:

- Aircraft environment – The same component mechanism and protocol that is used for loading data to and from aircraft installed component is also used to load data to and from the uninstalled component. This is generally used for loading Loadable Software Aircraft Parts (LSAP), but is also commonly used to load component test software needed during shop testing and maintenance.
- Resident Level – Software and data is loaded into the component using a communications protocol that is for uninstalled component use only. This includes transferring data to/from the component as a complete unit, a subcomponent such as a pc card, or a basic part such as an FPGA, PAL, or microcontroller IC.

6.1 ARINC Specification 664, Part 7

ARINC Specification 664, Part 7: *Avionics Full-Duplex Switched Ethernet Network,* is widely used for network centric aircraft such as the A350 and B787. It is built around commercial Ethernet standards (MAC, IP, UDP, SNMP) with provisions for deterministic behavior. This provides a high speed commercial Ethernet with provisions for guaranteed deterministic timing and redundancy that is required for avionics applications.

There are three basic types of network elements used within the ARINC 664, Part 7 system.

- End Systems (E/S) which constitute one or more sub-sections of a component used for ARINC 664, Part 7 communications.
- Switches that are used to route communication data packets to their correct destination End System.
- Virtual Links that are communication paths between two or more End Systems via Switches.

6.1.1 ARINC 664, Part 7 ADN Component Data Load Structure

Communications with data loading equipment that does not directly use ARINC 664, Part 7 is required at the aircraft environment. ARINC 664, Part 7 aircraft systems can interface to other communications protocols and functions using a gateway type device as illustrated below. The gateway device translates between the central point data loading protocol and the ARINC 664, Part 7 E/S data loading protocol.

6.1.2 ARINC 664, Part 7 ADN Component Data Load Protocol Translation

The translation required for data loading occurs primarily at the application layer. The four bottom layers of ARINC 664, Part 7 function similar to the standard OSI communication layers.

Ethernet loading protocols such as ARINC Report 615A also implement special functionality at the application layer.

Figure 7-1 shows how common data loading equipment communicates with components while installed on the aircraft.

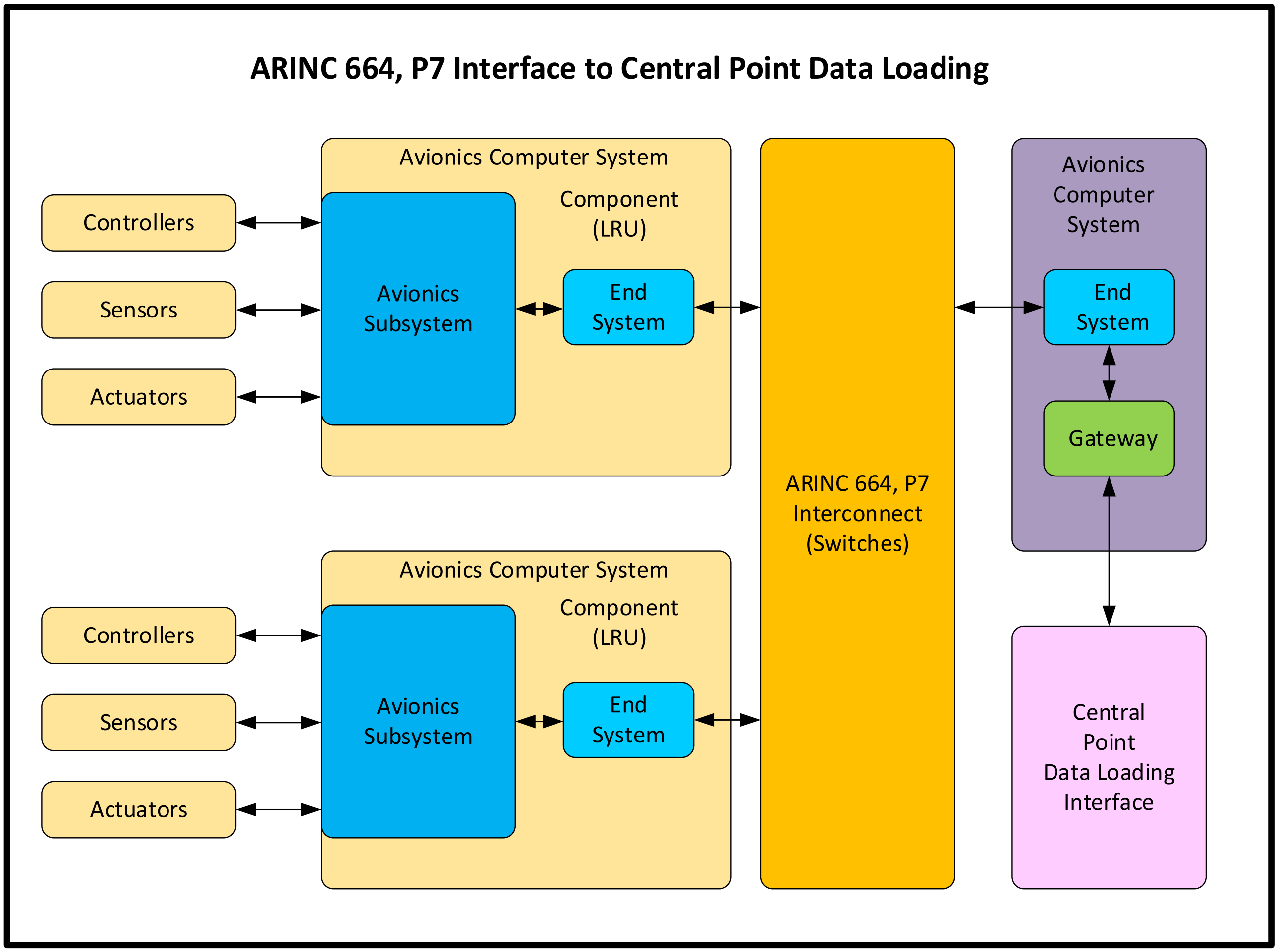


Figure 7-1 – ARINC 664, Part 7 Interface to Central Point Data Loading

6.1.3 End System (E/S) Communications

When a component is installed on the aircraft, the aircraft central point data loading system often requires the use of additional aircraft components to implement data loading to ARINC 664, Part 7 components. The additional components generally perform a bus protocol translation between the software loading source and the target LRU.

When the component is removed from the aircraft system, data loading functionality must be available without the use of additional expensive aircraft system components. Communications between the software loading equipment and the component should be implemented using the ARINC 664, Part 7 end system (E/S) communications stack that is native to the aircraft component.

Figure 7.2 shows the E/S protocol and communications architecture overview. Data loading communications is accomplished using standard communication protocol elements from the Transport layer using a Service Access Point (SAP) port.

Some switches incorporate an embedded E/S. The purpose of the switch E/S is to provide a means for functions that are external to the network in order to communicate with the switch. For example, data loading and network management functions communicate with the gateway (translational switch) via the gateway's embedded E/S.

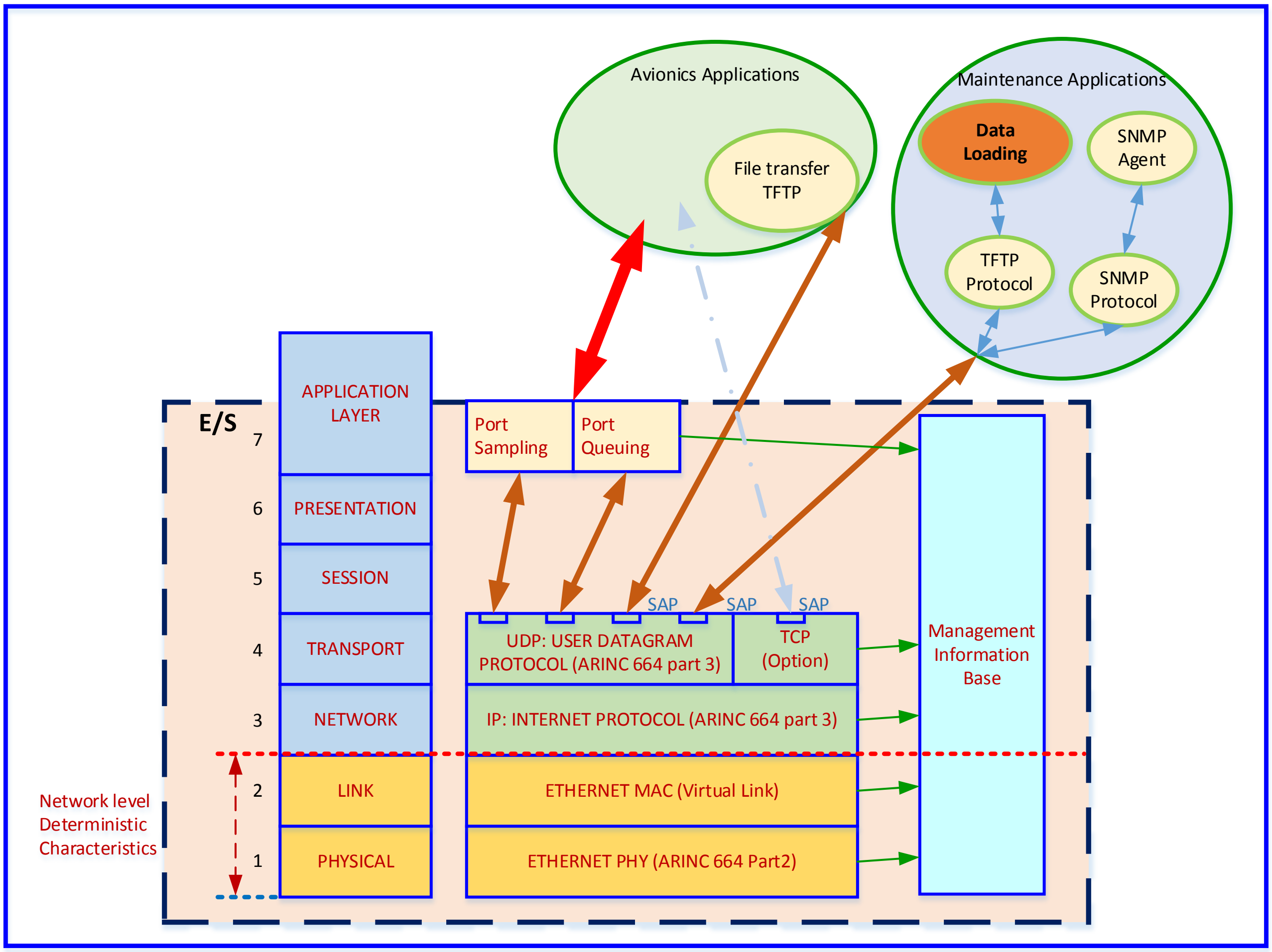


Figure 7-2 – End System Protocol Layers

6.1.3.1 Required E/S Specifications

Text to be added.

6.1.3.2 Virtual Link (VL) Parameters

A VL Parameters include, but are not limited to:

- A component's VL Address information
- Bandwidth Allocation GAP (BAG)
- Frame Size
- Max Allowed Jitter
- Number of Sub-VLs
- Account Type
- Priority
- Network Selector
- Skew Max

6.2 ARINC 615 (ARINC 429 Bus)

For LRUs that are directly loadable using ARINC 615, equipment that complies with the standard may be used. However, there may be other requirements to put the target LRU in the proper state for loading data.

Some of the required information is as follows:

- The ARINC 429 bus speed used by the LRU to both send and receive data. This is either “Low Speed” (12.5 kb/s) or “High Speed” (100 kb/s)
- Any discrete signals that needed to be asserted by the data loader for the LRU to put into a loadable state (“data load mode”)
- Any other data inputs needed to set LRU to load state such a Weight on Wheels, Engines off, etc.
- The ARINC 429 labels that the LRU uses to receive and transmit ARINC 429 data

6.3 ARINC 615A (Ethernet Bus)

For LRUs that are directly loadable using ARINC 615A, the following is required:

- THW_ID + POS (see ARINC 665 for details)
- Loader IP Address: The IP Address the loader should set itself to
- LRU IP Address: The IP address of the LRU that the loader should attempt to communicate with
- IP Subnet Mask: Loader must be on same subnet as LRU
- Gateway Address: If the LRU to be loaded is not on the same network as the dataloader, the gateway address should be set to the IP address of the router/device that can route packets between the two networks
- UDP Client/Server Ports: The UDP ports if they deviate from the 615A standard
- FIND Parameters: If the FIND network discovery command (defined in 615A) is supported by the LRU, then indicate if broadcast is supported or specify unicast FIND IP address required
- ARP IP Resolution: If ARP (which associates IP addresses to MAC addresses) protocol is not supported by the LRU. If ARP is not supported/allowed by the target, the MAC address of the target must be specified for the data loader

6.4 ARINC 615A ARINC 664, Part 7 Supporting ARINC 664, Part 7

This section describes material specific to central point loading using ARINC 615A loading equipment. This may include:

- Parameters in “ARINC Report 615A for Ethernet bus” plus the following additional parameters:
- LRU ARINC 664, Part 7 Communications Port Numbers
- BAG value
- LRU MAC Address
- Ethernet MTU Size
- Lmax value

6.5 ARINC Specification 826 for CAN Bus

This section describes material specific to central point loading using ARINC 826 loading equipment. This may include:

- Units loaded using ARINC 826
- Need some parameters here
- THW_ID + POS?
- CAN ID
- Speed

6.6 Other Protocol Definition If Not Defined by Any ARINC Standard.

When no details are published in a standard about loading protocol, detailed information will be required. Detailed specifications will be required for the following:

- Physical layer communications bus used
- Data loading communication protocol
- Inputs necessary to put LRU in load state
- Distributed software file format
- Complete software load implementation
- CRC calculation method to check the integrity of the software load

7.0 HARDWARE CONSIDERATIONS AND ATTRIBUTES

7.1 Hardware Considerations for Component Software Loading (CSL) Shop Loading

There are several considerations regarding the physical connectivity as part of the Component Software Loading (CSL) process.

7.1.1 Planning for CSL

A supplier of an aircraft component must plan for data loading of the LRU in the shop. Data loading in the shop may be performed at a dedicated loader or the loader may be embedded in an Interface Test Adapter (ITA) or Test Unit Adapter (TUA) and performed during functional test using Automated Test Equipment (ATE). Operational software in a Loadable Software Airplane Part (LSAP) may be loaded on-wing or in the shop.

7.1.2 Accessibility Considerations for CSL

Data loading is used to facilitate various types of data. During maintenance procedures in the shop test software datasets may be temporarily loaded into the LRU, the fault history memory device may be accessed and may be read and erased, or the configuration memory device may be updated with new hardware or software version information. Memory device Integrated Circuits (ICs) may be removed and replaced requiring re-load of the stored data. A memory device that could be removed and replaced includes:

- A microcontroller IC with its embedded memory
- A memory IC associated with a microprocessor
- A Non-Volatile Memory (NVM) memory IC

7.1.3 Physical Bus Specification

The Component Software Loading equipment may be a standard, or may be specifically customized and designed for the component or Unit Under Test (UUT). The physical layer data bus required to transfer data may be according to a published standard, or customized for the UUT requirements. For example, the component’s data load bus may require different voltage-levels. If voltage level adapters needed they may reside in the dedicated loader, in the test system, or in the ITA.

7.1.4 Connectivity for CSL

The computer used to control the load procedure will have to communicate with the bus used requiring a bus adapter. Adapters may be a standalone device between the computer and LRU or may be a card residing in the computer or may be built in to the LRU or ITA. Software on the computer controls the load process. COTS adapters generally come with application software that often needs to be modified to interface with the target LRU.

7.1.5 Shop Replaceable Units (SRUs) such as Printed Wiring Assemblies (PWA)

Generally, there are no provisions to access and load the Shop Replaceable Units (SRUs) such as Printed Wiring Assemblies (PWA) directly on the aircraft. However, in the shop the technician is able to open the LRU and gain access directly to the SRU connectors and any internal LRU connectors.

7.1.6 Interface Specification

The supplier should define the interface specification as it applies to data loading of that specific LRU. The interface specification should include physical definition of the interface connector and its pins, including the pins needing power or ground for data loading. The specification should also provide an electrical description of the signal levels, timing and format.

7.2 The Interface Connector Physical Definition

In the aircraft, the Target LRU interfaces with aircraft systems through the LRU's main connector, which is accessible without any disassembly of the component. The LRU main connector mates with the aircraft connector to interface with the aircraft wiring. The connector provides pins for all signals between the aircraft loader and the Target LRU for loading of LSAP.

Loading in the shop also uses the LRU's main connector with corresponding mating connectors used on the loader side. The actual data load interface may require as few as three dedicated pins embedded within the LRU main connector. A cable connects those pins the LRU main connector either directly to the loader or to an adapter.

7.3 Communications Interface Bus

7.3.1 General

A data bus is needed to transfer the data. Commonly used data busses include: USB, RS232, RS422, RS485, ARINC 429, ARINC 629, ARINC 825 CAN, or ARINC 664, Part 7.

7.3.2 Specifics

Following are some commonly used options for a data loading bus within a LRU with a summary and comparison of each:

7.3.2.1 Twisted-pair Wiring

Twisted-pair wiring is required by many Standards. Shielded twisted-pair cabling is often specified for many applications when Electro-Magnetic Interference (EMI) is a concern. The shield may be grounded at either or both ends.
A Supplier should make known if shielded wire is needed, and make known if the shield is grounded at either or both ends.

7.3.2.2 Interface Integrated Circuits

Interface Integrated Circuits – LRUs or adapters may use a Universal Asynchronous Receiver/Transmitter (UART) integrated circuit that translates the parallel data from the processor into serial data. LRUs or adapters could also use a dedicated Transceiver IC such as an A429 or A629 Transceiver.

When utilizing COTS components, such as an A429 Transceiver or similar, the Supplier should make known the full part number for the transceiver IC to allow users to obtain the manufacturer's specification for the embedded piece part. Additionally, the supplier should clearly make known any departure from the interface standard such as different transmission speed or formatting differences.

7.3.2.3 CAN Bus

CAN bus is a twisted-pair bus. It is non-zero-crossing bus with nominal +7.5V and +2.5 null levels. CAN bus specification requires a 120 Ohm parallel termination resistor for noise immunity. CAN bus requires at least three pins (V_{HI}, V_{LO} and Signal

Ground). CAN 2.0A has 11-bit identifier, CAN 2.0B has a 29 bit identifier. The version used must be made known. CAN bus data rate speed is inversely related to maximum cable length so the CAN bus data rate used must be made known.

The Open Systems Interconnection (OSI) model defines seven (7) levels. The CAN bus Standard only addresses level 1 and 2: the physical and electrical characteristics such as bit timing, amplitude and the Message Identifier (ID) structure. Users determine level 3 through 7: network, transport, control, presentation and application software: a software application must be specifically written for the target LRU. CAN bus standard allows each user to choose their application software, so for example applications using National Instruments products may use LabView as the application software. CAN bus requires a software application specifically written for the target LRU; often by modifying the COTS adapter provided software application to interface with the target LRU. Therefore CAN bus requires a software application specifically written for the target LRU.

7.3.2.4 D-sub Connectors

9-pin D-sub connectors are frequently used for CAN bus. Signals CAN L, CAN GND, CAN H and GND are connected. Optionally there may be a Shield connection and there may be a CAN VCC connection.

7.3.2.5 COTS CAN Bus Adapters and CAN Bus Cards

COTS CAN bus adapters and CAN bus Cards (may reside within PC) converting CAN to USB or to RS-232 are readily available. The hardware may be embedded in the LRU or within a Test Interface Adapter (TIA) or a dedicated CAN bus adapter. These COTS products come with needed cables and generic software applications. The user (Supplier) modifies that application software as needed for the specific LRU application.

COTS CAN Bus test equipment parts should be listed in an equipment table within the CMM that a shop would use to load LSAP or data.

7.4 Interface Adapters

There are instances where a component needs more than a simple cable to establish communication with an independent loader that can perform software loading off wing. In cases where the loader is designed as a universal loader then specific extra hardware needs to be defined. Definition of these extra pieces helps implementers deigning an adapter for specific use of a specific LRU.

7.4.1 Pin Definition to Power the Unit Sufficient for Data Loading

The electronics of the target LRU must be powered up in order for data to be loaded. In the shop, it is preferable to use an existing shop setup that already provides for powering the LRU. It is also an option to have a separate setup solely for loading data.

In the shop, various maintenance activities regarding software may take place, including:

- Data loading of replaced memory components
- Retrieving, dumping and decoding of fault history NVM
- Erasing of fault history NVM

- Reading and writing LRU Hardware/Software Configuration to the NVM
- Pre-test loading of test-only software
- Post-test loading of flight software replacing test-only software
- Writing application(s) to microcontrollers
- Writing program(s) and configuration databases to microprocessors and associated memory devices

In each of these setups in the Shop, it is possible to power the LRU from power source in an ATE or from a specific loader.

Return to service test equipment often has the resource for stimuli including different power supplies that accommodate the need for different functions of a LRU, e.g., 115 Vac, 28 Vdc, 5 Vdc, etc. As a result, when Suppliers embed the interface for shop loading into the test equipment (ATE, Loadboards, TUA, Adapters, etc.) the LRU will be powered by any means cited previously.

Information regarding Pin definition to power an avionics unit must be included in a CMM and in a complementary TSDP.

7.4.2 Pin Definition to Enable the Unit to be Shop Loaded

Avionics may have programmable Shop Mode pins. In the shop the LRU must be configured into the Shop Mode using the Shop Mode pins to start a software download. Usually Shop Mode pins are some grounded pins not used on the aircraft or aircraft air/ground and data load enable discrete input pins.

The pin numbers, type of input (e.g., Open/Ground, +15V/Ground, etc.) and the required voltage levels of the Shop Mode discrete inputs must be defined.

Data loading cannot take place until the LRU or SRU is in proper pins configuration. For example, in the Aircraft data loading is inhibited if an Air/Gnd input(s) is set to Air. This situation is normal mode of the LRUs fitted onto the aircraft. In the shop the shop mode discrete inputs must be correctly set.

Other discrete inputs such as Data Enable pins need to be set to the necessary state to load or read data. These discrete inputs are needed to load data, and a Shop Data Loader must be capable of setting these discrete inputs.

Information regarding Pin definition to enable the unit to be shop loaded must be included in a CMM and in a complementary TSDP.

7.4.3 Pin Definition for the Bus Interface Used to Load

Dedicated pins also need to be defined for specific load protocol such as ARINC 429 or ARINC 825 CAN bus. The circuitry providing load interface between LRU and outside world usually have path to the common connector found in the back plane of the LRU, but there are instances where suppliers opt to provide dedicated pins using a separate connector. Without information regarding these dedicated pins, one will encounter difficulties to creating data loader that is suitable for specific avionics LRUs.

Information regarding Pin definition to enable the unit to be shop loaded must be included in a CMM and in a complementary TSDP.

7.4.4 LRU Power-on Time

The Target LRU must be powered and provided all the required stimuli and discretes to enable data loading. In addition to providing all the required stimuli and discretes, each specific target LRU may have other implementation details to consider such as minimum LRU power-on time before the LRU can accept data. These implementation details must be specified.

7.4.5 Power Specification

Avionics designs aim multiple functions that require different level of power, different frequencies. Specification documentation about powering a LRU for shop data loading functions must provide nominal Voltage and Max Voltage, nominal Current and max Current, and Frequency.

7.5 Data Load CSL Communications Bus

There are many possible connection methods that can be used to transfer data to and from an aircraft component. There are several documented standards that define the transfer of data between a data source and target device at the fundamental level, commonly referred to as the physical layer.

7.5.1 ARINC 429 Bus

ARINC 429 is a twisted pair bus. It is self-clocking, asynchronous 32-bit serial bipolar return-to-zero bus with nominally differential $\underline{+}$10V signal level. Transmitting at either 12.5 Kilobits or 100 kilobits per second. Transmission and reception of ARINC 429 are through separate ports. ARINC 429 requires at least three pins (V_{HI}, V_{LO} and Signal Ground).

Table 6-1 – ARINC 429 Transmitter Voltage Levels

Measurement	State and Voltage (Vdc)		
	HI	NULL	LO
Line A to Line B	+10 ± 1.0	0 ± 0.5	-10 ± 1.0
Line A to Ground	+5 ± 0.5	0 ± 0.25	-5 ± 0.5
Line B to Ground	-5 ± 0.5	0 ± 0.25	+5 ± 0.5

Table 6-2 – ARINC 429 Receiver Voltage Levels with Tolerance

State	Voltage (Vdc) with Tolerance
HI	+6.5 to +13
NULL	+0.5 to -0.5
LO	-7.25 to -11

7.5.2 Ethernet

Text to be Added.

7.5.3 ARINC 664, Part 7

Text to be Added.

7.5.4 ARINC 825 CAN and CAN FD

Text to be Added.

7.5.5 FlexRay

Text to be Added.

7.5.6 USB

USB – Universal Serial Bus. USB is an industry standard communication protocol used to connect a computer to its peripheral devices. The majority of USB implementations use a Type A or Type B connector. The Type A connector is a flat, rectangular interface. The Type B is square in shape with beveled corners. Additionally, there are multiple types of USB.

If USB is utilized, a supplier needs to make known as a minimum the specific USB version, for example USB 1.1, 2.0, or 3.0 used for the data interface.

7.5.7 RS-232

RS-232 is an unbalanced active signal of originally up to +15 Vdc to low to -15 Vdc. Valid signals are > ±3 Vdc, the range between -3 to +3 Vdc is not a valid RS-232 level. The RS232 specification does not define the connector to use, but does recommend 25 pin DB25 connectors which are commonly used. The specification does define signal names.

Table 6-3 – RS232 Logic and Voltage Levels

Data Circuits	Control Circuits	Voltage
0 (Space)	Asserted	+3 to +15 Vdc
1 (Mark)	De-Asserted	-15 to -3 Vdc

RS232 standard was changed over time and in revision D of EIA-232 the voltage range was extended to ±25 Vdc. Signal levels of ±5 Vdc, ±10 Vdc, ±12 Vdc, and ±15 Vdc are all commonly seen depending on the voltages available to the line driver circuit.

In cases where a common interface is used, the signal names provided in the interface specification should be consistently used by the Supplier providing the LRU. Any use of non-standard nomenclature should be listed.

8.0 NON-VOLATILE MEMORY

Non-Volatile Memory (NVM) generally allows for digital memory contents to be retained when power is removed from the memory devices. However, NVM has been used in a specific context within the avionics industry. NVM is often referred to as the part of memory that stores LRU faults or other transient data in real time.

The following sections refer to the transient data storage area of memory where the content of the data is downloaded, stored and analyzed. This part of LRU memory is sometimes cleared after transient data downloaded and is stored in a ground system.

8.1 General

NVM is a term given to the Read-Only Memory (ROM) or Electrically Erasable Programmable Read Only Memory (EEPROM). NVM can also be applied to Random Access Memory (NVRAM). 'Non-Volatile' in ROM and NVRAM means that, stored data is retained when electrical power is turned off.

NVM is widely used in many avionic types of equipment fitted to modern aircrafts. NVM stores product software such as OPS, OPC, navigation data, other embedded software, and mostly NVM is often included in an avionics design to aid identification of fault and rectification by the provision of a history of recorded fault signals or fault codes. Normally BITE design aims to achieve this goal. Thus, particular avionics equipment should have a NVM structure design to serve different purposes as mentioned.

NVM activities management within the avionics equipment such as loading default value, activating operating software from NVM or BITE running schedule, etc., are not within the scope of this document.

For overall aircraft health management and maintenance activity the component shop maintenance function should be able to access, retrieve and decode the fault data recorded in NVM of the aircraft avionics systems using shop environment test equipment.

COMMENTARY

Airlines routinely perform maintenance and utilize the NVM contents for troubleshooting, life cycle support, and fleet health purposes.

Maintenance requirements at shop environment also need the NVM access to retrieve, decode, erase and write data from and to a component as required. Instructions for performing this function need to be included in the CMM. Additionally, specific technical information should be included that is required to develop equipment necessary to perform this function.

If the description of the unit contains NVM it implies that a microprocessor or microcontroller exists in the unit. Detailed schematic of the unit will help loader developer to establish the path for loading procedure. Normally ICs for NVM are located near the microprocessor and will have address and data bus indication.

An NVM activity and sequence flow chart should be provided within a Component Maintenance Manual (CMM) or a Technical Support and Data Package (TSDP) to help understanding the complex usage of NVM and properly handling it in the shop environment. One important matter is not to accidentally erase the NVM faults before retrieving it.

Details of how to develop CMM sections regarding disassembly are defined in ATA iSpec2200.

8.2 NVM Attributes

The NVM attributes discussed hereafter will focus on FLASH or EEPROM that store data for maintenance and flight history purpose. These data are the target that maintenance in a shop environment should deal with.

The following are typical NVM reference attributes for stored data:

- Flight History
- Stored Fault Codes
- Power Cycle
- Elapsed Time Indicator (ETI)
- Serial Number
- Part Number
- Repair History
- Environmental Conditions (Temperature, Humidity, etc.)

8.3 NVM Physical Access

The processes in the following sections should be clearly defined in a CMM.

8.3.1 NVM Access from Assembled Component

Some components provide access to the NVM communications bus via an external connector.

8.3.2 NVM Access from Partially Disassembled Component

Obtaining access to the internal area of the component is sometimes necessary to interface to the NVM communications interface.

8.3.3 NVM access from sub-component

Specific pins number on a common LRU back plane connector should be defined when a circuit card type sub-component must be removed from the component.

8.4 NVM Protocol Access

The following sections describe the processes for accessing the NVM contents.

8.4.1 Protocol Provided by Component Design

Components should provide communications protocol functionality for downloading and managing NVM fault data. The communications protocol used to download NVM data should be described in the CMM.

8.4.2 Protocol Not Provided by Component Design

If protocol functionality is not provided by the CMM, then the TSDP should provide specifications for the items below.

8.4.3 NVM Initialization

To put the NVM circuitry into operational status a unit with NVM should be initialized properly. Initiating voltage and current as well as after power up waiting time need to be clearly defined in a CMM or TSDP.

NVM design structures compose of physical layers indicated by addresses, known as an NVM address. Start and stop access should be provided within a CMM or TSDP for software loading, data retrieving purpose.

Usually NVM contains two regions for fixed addresses and other regions for programmable addresses, known as a NVM memory map. Fixed addresses are reserved for defined field while designated words can be used for programmable areas. As an example, Operating Systems are assigning physical memory, how much file data is cached in NVRAM, or how much NVRAM is used by the kernel and device drivers. The following are examples of information may be recorded in a unit memory map.

- Use Counts: usage summary by type and paging list (Component Power Up/Dn)
- Processes: process working set sizes
- Priority Summary: prioritized standby list sizes
- Physical Pages: per-page use for all physical memory
- Physical Ranges: physical memory addresses
- File Summary: file data in RAM by file
- File Details: individual physical pages by file [2]

8.4.4 NVM Pin Definition

The pin definitions of the NVM must be known, including:

- Power specification
- Other inputs

8.4.5 NVM Storing

Besides product software, part of a LRU maintenance requirements in shop environment is rewriting to NVM the part number and serial number of the unit. To fulfill this requirement, the memory addresses need to be provided. It is understandable that memory map/addresses are required for designing a load program set and a loader needs to know where to direct the software or data. This would include:

- Memory address
- Bus protocol
- Other inputs

8.4.6 NVM Reading

A main function provided by NVM is the stored data/fault that operators need to retrieve and use in troubleshooting a unit or a system. NVM records are key factor for incidents investigations. In order to fulfill this requirement memory, addresses need to be provided within a CMM or a TSDP.

- Memory address
- Bus protocol and special physical bus connectivity and specification
- Communication protocol(s)
- Other inputs

8.4.7 NVM Decoding

Whenever the data is available a decoder or parser is needed to decipher the code into plain English for practical usage. Most of data or faults recorded in the LRU NVM are in form of BIN, HEX, or OCTA formats that need to be translated to human readable format. A decoding utility is essential for unit maintenance or for a system fault analysis. A Technical Support and Data Package should contain the specification or functional requirements for this decoder, including:

- Parser definition to decode the data to human readable information
- Other inputs

8.4.8 NVM Erasing

Sometimes the NVM of a unit is temporarily using test software for functional testing. This test dataset should be erased after completing the performance of a return to service functional test and replaced by flight software. In the operational mode of a unit designed with NVM for fault history recording the NVM older information may be erased when newer data become available since the size of memory pages may be limited per design.

- Memory address
- Other inputs

COMMENTARY

NVM can be used in alerting equipment. In a few specific cases where it is used in alerting equipment, it has been envisaged from the outset that such a history might also be of operational use by additionally recording alert activations – Terrain Collision Avoidance System (TCAS) and Terrain Avoidance and Warning System (TAWS) are the main examples of this and the most likely to be routinely used as data sources for aircraft operators rather than just investigation agencies.

Many familiar items of avionic equipment are nowadays of solid state design and incorporate NVM. Apart from Quick Access Recorders (QARs) and their associated Digital Flight Data Acquisition Units (DFDMUs) fitted for OFDM purposes, which may not necessarily be designed to record much more data than their crash-protected counterparts, examples include, but are not limited to, the following:

- Digital Flight Data Acquisition Units (DFDMUs) in some Digital Flight Data Recorders (DFDRs)
- Full Authority Digital Engine Control (FADEC) units
- Generator Control Units (GCUs)
- Fuel control, management and monitoring units
- Brake Control Units (BCUs)
- GPS-based navigational equipment
- Engine EECs
- Flight Management Computers (FMCs)
- Windshear detection equipment
- Cabin Pressure Controllers
- Central Maintenance Computers (CMCs)
- Spoiler Control Units (SCUs)

- Display Control Panels (DCPs)
- Air Data Computers (ADCs)
- Maintenance Diagnostic Computers (MDCs)
- PFDs and MFDs which may store data from AHRS equipment
- (Air Data) Inertial Reference Units (ADIRUs or just IRUs)

COMMENTARY

Analyzing the NVM recorded data is outside the scope of this document. Only the capability to access NVM contents is beneficial for the processes and requirements of this standard.

9.0 CALIBRATION

Loading of calibration data is sometimes required. Calibration is the application of specifically known and accurately measured input to ensure that an item will produce specifically known output called standard which is accurately measured or indicated. Calibration includes adjustment or recording of corrections as appropriate.

The formal definition of calibration by the International Bureau of Weights and Measures is the following: “Operation that, under specified conditions, in a first step, establishes a relation between the quantity values with measurement uncertainties provided by measurement standards and corresponding indications with associated measurement uncertainties (of the calibrated instrument or secondary standard) and, in a second step, uses this information to establish a relation for obtaining a measurement result from an indication.”

The calibration process may begin with the design of the measuring instrument that needs to be calibrated. The design needs to be able to hold steady measurements through its calibration interval. In other words, the design must be capable of measurements that are within engineering tolerance when used within the stated environmental conditions over some reasonable period. Having a design with these characteristics increases the likelihood of the actual measuring instruments performing as expected. Basically, the purpose of calibration is for maintaining the quality of measurement as well as to ensure the proper working of particular instruments.

Calibration Data can be defined as a set of established values that are proven to be good enough for anticipated use of an application. Calibration Data can be loaded into components for use during a functional test of the unit. For example, LED lights need to provide response accurate voltage to requirement so the emitting light will meet the spectrum that a combination with other lights can create a light show.

Calibration Data needs to be provided in a CMM or in a TSDP for test purposes. The load specification should be provided to enable develop ground loading equipment. The Calibration data loading procedure must also be provided.

10.0 BUILT-IN TEST EQUIPMENT

Built-In Test Equipment (BITE) is characterized primarily as a passive fault management and diagnosis built into airborne systems to support the maintenance process. BITE design is normally oriented towards Line Maintenance function.

The functionality of BITE includes the detection of the fault, the accommodation of the fault such as how the system actively response to the fault, and the annunciation or logging of the fault to warn of possible effects and/or aid in troubleshooting a faulty equipment.

One application of BITE is the performance of the onboard removal procedure specified in an exemplary troubleshooting manual BITE data are processed through the OMS (On Board Maintenance System) to provide plain-language maintenance messages for the identification of failed LRUs. Only the messages obtained during the performance of this procedure are used to remove faulty LRUs.

Downloading of BITE data is sometimes required as part of a shop test. In addition, access to all the data stored in the BITE memories is required as the LRU remains in the repair shop. In this case, although the information is not coded and provided in English, supplementary information may be necessary such as log book information, line maintenance actions, and aircraft configuration.

A specification for decoding binary encoded data to human readable language should be available.

Built-In Tests (BIT) or Built-In Self-Tests (BIST) is often used for this same function or, more specifically, about the individual tests.

PBIT (Power up BIT) is a built-in test that automatically starts when the component is power up. This can happen on the aircraft or in the shop.

CBIT (Continuous BIT) is a built-in test designed such as a test is always performed to check for the health of a component on aircraft during a flight.

10.1 Functionality of BIT or BITS

The results returned from the BIT, or BITS, testing can assist with:

- Analysis of failure monitoring results
- Reporting and memorization of failures
- Management of tests

10.2 Application of BIT or BITS

Almost all modern avionics component incorporates BIT or BIST capability. In avionics, the purpose is to isolate failing line-replaceable units, which are then removed and repaired elsewhere, usually in depots or at the manufacturer. Commercial aircraft only make money when they fly, so they use BIST to minimize the time on the ground needed for repair and to increase the level of safety of the system which contains BIT or BIST. When BIT or BIST is used in flight, a fault causes the system to switch to an alternative mode or equipment that still operates. Critical flight equipment is normally duplicated, or redundant. Less critical flight equipment, such as entertainment systems, might have a “limp mode” that provides limited functionality.

11.0 FIBER OPTIC USE AND TESTING

Fiber Optics is a technology using glass or plastic threads, considered as fibers, to transmit data. A fiber optic cable consists of a bundle of glass threads, each of which is capable of transmitting messages modulated onto light waves. Advanced fiber optic technology enables avionics to provide faster data transmitting and receiving. Manufacturers could take advantage of fiber optics cable over metal cables such as greater bandwidth, less susceptible to interference, thinner and lighter and data can be transmitted digitally rather than analogically, to design and build component with more reliability with less weight.

If fiber optic technology is used in an avionics component the test method needs to be described and a test procedure including fiber optic test equipment/tool needs to be provided within the CMM or in a complementary TSDP.

If data loading is performed using fiber optic technology, a full communication and loading protocol specification should be available.

APPENDIX A ACRONYMS

ACLSP	Aircraft Controlled Loadable Software Parts
ACS	Aircraft Controlled Software
ADC	Air Data Computer
ADIRU	Air Data Inertial Reference Unit
ADN	Aircraft Data Network
AEEC	Airlines Electronics Engineering Committee
ARP	Address Resolution Protocol
BAG	Bandwidth Allocation Gap
BCU	Brake Control Unit
BFP	Batch File Part
BIST	Built-In Self-Tests
BIT	Built-In Tests
BITE	Built-In Test Equipment
CBIT	Continuous BIT
CC	Check Characters
CCA	Circuit Card Assembly
CD	Compact Disk
CMM	Component Maintenance Manual
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSLL	Component Software Load Level
DCP	Display Control Panel
DFDMU	Digital Flight Data Acquisition Unit
DFDR	Digital Flight Data Recorder
DHCP	Dynamic Host Configuration Protocol
DVD	Digital Versatile Disc
EEPROM	Electrically Erasable Programmable Read Only Memory
E/S	End System
ETI	Elapsed Time Indicator
FAA	Federal Aviation Administration
FADEC	Full Authority Digital Engine Control
FBS	Factory BITE Software
FLS	Field Loadable Software
FMC	Flight Management Computer
FPGA	Field Programmable Gate Array
FTK	Functional Test Kernel
GCU	Generator Control Unit
HCLSP	Hardware Controlled Loadable Software Part
HCS	Hardware Controlled Software

APPENDIX A
ACRONYMS

HDR	Header
ICMP	Internet Control Message Protocol
IMA	Integrated Modular Avionics
IP	Internet Protocol
IP	Internet Protocol (IP, Layer 3)
IPC	Illustrated Parts Catalog
IRU	Inertial Reference Unit
JTAG	Joint Test Action Group
Lmax	Largest Ethernet Frame
LRM	Line Replaceable Module
LRU	Line Replaceable Unit
LSAP	Loadable Software Aircraft Part
LSP	Loadable Software Part
MAC	Media Access Control
MDC	Maintenance Diagnostic Computer
MRO	Maintenance Repair Organization
MSD	Mass Storage Device
MSP	Media Set Part
MTU	Maximum Transmission Unit (in bytes?)
NVM	Non-Volatile Memory
NVRAM	Random Access Memory
OSI	Open Systems Interconnection
PAL	Programmable Array Logic
PBIT	Power up BIT
P/N	Part Number
PDL	Portable Data Loader
PMAT	Portable Maintenance Access Terminal
PWA	Printed Wiring Assemblies
QAR	Quick Access Recorder
RAM	Random Access Memory
RBP	Resident Boot Program
ROM	Read-Only Memory
SB	Service Bulletin
SCL	Software Control Library
SCU	Spoiler Control Unit
SNMP	Simple Network Management Protocol
SRU	Shop Replaceable Units
SW	Software
TAWS	Terrain Avoidance and Warning System
TCAS	Terrain Collision Avoidance System
TCP	Transmission Control Protocol
TFTP	Trivial Transfer Protocol

APPENDIX A
ACRONYMS

TIA	Test Interface Adapter
TSDP	Technical Support and Data Package
UART	Universal Asynchronous Receiver/Transmitter
UDP	User Datagram Protocol Layer (UDP, Layer 4)
USB	Universal Serial Bus
UUT	Unit Under Test
VL	Virtual Link

APPENDIX B GLOSSARY

Aircraft Loading

Aircraft loading is performed while the component is installed on an aircraft.

Bandwidth Allocation Gap

Definition is required.

Bandwidth Contract

The End System has a Bandwidth Contract for each Virtual Link and must comply with this contract.

Batch File Part

In the context of aircraft environment data loading, there is a desire by the airlines to be able to define a “batch” type file that enables the maintenance person to select a file that defines for the Data Loader a series of LSPs that should be loaded into one or more Target HW Position. For more information see ARINC 665.

Boeing Legacy Part

Prior to the formation of ARINC Report 665, Boeing had established standards for software parts. LSP standards were found in Boeing Document D6-55562-5. Media Set part standards were in Document D6-55562-6. Parts designed in compliance with these early Boeing standards, are referred to as Boeing Legacy parts.

Although the intent of pre-665 Boeing Standards compliments that of the ARINC Report 665, the structures are not directly compatible. Accordingly, Boeing Legacy parts may be represented in ARINC 665 LSP and Media Set Part formats using select provisions, found in attachments to the ARINC Report 665.

Bus

A bus is a set of physical connections (cables, printed circuits, etc.) which can be shared by multiple hardware components in order to communicate with one another. Aircraft busses such as ARINC 429 and ARINC 664 P7 are external of the components.

CML-1,2,3

For more information, reference **ARINC Report 663:** *Data Requirements for Avionics Component Maintenance*.

Component

A component or an appliance is any instrument, mechanism, equipment, part, apparatus, or accessory installed in or on an aircraft, and is not a part of the airframe, engine, or propeller, that is used or intended to be used in operating or controlling the aircraft in flight.

In the context of this document, component refers to ‘avionics component’ which is a self-contained avionic hardware assembly installed on an aircraft in a manner that facilitates the removal from and replacement to the aircraft.

A component is generally synonymous to a Line Replaceable Unit (LRU). It is commonly used interchangeably with LRU which is a component that can be removed from and replaced to the aircraft in a relatively short period of time.

APPENDIX B
GLOSSARY

Cyclic Redundancy Check (CRC)

A value calculated from a block of data and used to detect changes to the data due to, for example, corruption of memory. CRC algorithms are chosen so that changes in the block of data are very likely to change the calculated value.

Data File

A specific file that contains, in addition to other information, the actual data that is the object of the load process. One or more data files plus a header file make up an object of a load function.

Deterministic Behavior

In the context of ARINC 664: Avionics Full-Duplex Switched Ethernet Network, the determinism is defined as the control of maximum transmission delay through the network where the enabler of such control is precisely the bandwidth contract. The custom Ethernet Switch provides better capability for determinism than usual Ethernet Hub because there is no collision and no transmission random retry.

Device

In the context of this document, a device generally refers to a component part that is programmable, such as an integrated circuit (IC), FPGA, EEPROM, CPLD, PAL.

The term Device is overloaded and is often used as reference to a personal device such as a cell phone or PDA. It is sometimes used as a general reference to a piece of electronic equipment.

Disk

A 3.5-inch Flexible Disk Cartridge as specified in ISO/IEC 9529-1 "International Standard - Dimensions, Physical and Magnetic Characteristics" Section 7.1.

End System (E/S)

ARINC 664, Part 7 network device whose applications access the network components to send or receive data from the network. Examples include: LRU, RDC, modular avionics, etc.

Header File

A specific file that contains information about the load that is needed to support the load process and software handling processes.

Integrated Modular Avionics

The Integrated Modular Avionics concept attempts to reduce the variety of LRU's by designing the avionics with common components. E.g. one type of power supply, one type of memory unit, one type of processing board and only a few variations of input/output modules. This reduces the number of replacement parts needed to be kept on stock. The LRUs of IMA have more generic functions than in a traditional avionics system.

Field Programmable Gate Array

Field Programmable Gate Arrays (FPGAs) are semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. FPGAs can be reprogrammed to desired application

APPENDIX B
GLOSSARY

or functionality requirements after manufacturing. This feature distinguishes FPGAs from Application Specific Integrated Circuits (ASICs), which are custom manufactured for specific design tasks. Although one-time programmable (OTP) FPGAs are available, the dominant types are SRAM based which can be reprogrammed as the design evolves.

Joint Test Action Group (JTAG)

The Joint Test Action Group (JTAG) is an electronics industry association formed in 1985 for developing a method of verifying designs and testing printed circuit boards after manufacture.

One of the functions of the JTAG standard is to allow device programmer hardware to transfer data into internal non-volatile device memory. Data may be transferred directly into a device such as an FPGA, or indirectly via a JTAG device connected to another target device such as a flash memory device.

Line Replaceable Item (LRI)

Synonymous for a component that can be removed from and replaced on the aircraft in a relatively short period of time.

Line Replaceable Module (LRM)

Synonymous for a component that can be removed from and replaced on the aircraft in a relatively short period of time.

Line Replaceable Unit (LRU)

A Line Replaceable Unit.is a piece of hardware that can be exchanged for a replacement part in a relatively short time by only opening and closing fasteners and connectors. Synonymous for a component that can be removed from and replaced on the aircraft in a relatively short period of time.

Loadable Software Airplane (Aircraft) Part (LSAP)

“Software” that is: (1) intended for transfer into its “target hardware” without physically altering the hardware or otherwise triggering the need for return-to-service conformity testing of the “target hardware”, and (2) needs to be formally referenced independently from any other part (hardware or software) by airline or aircraft manufacturer’s processes, and (3) is not configuration controlled as a component part of the target hardware, and (4) is configuration controlled as a component part of the aircraft.

COMMENTARY

Inherent in the definition of a LSAP is the concept that the LSAP is an independent, autonomous aircraft part from the target hardware. Installing a LSAP on the aircraft must not impact the conformity of the target or any other aircraft hardware. However, it may impact the aircraft conformity.

Loadable Software Airplane Parts (LSAP) is a subset of the LSP class of parts. All provisions for LSPs in this document also apply to LSAPs.

Loadable Software Part (LSP)

APPENDIX B
GLOSSARY

"Software" that is intended for transfer into its "target hardware" without physically altering the hardware, and needs to be formally referenced independently from any other part (hardware or software) by airline or aircraft manufacturer's processes.

Logical Media Set Part

Definition is required.

Mass Storage Device

A large capacity nonvolatile storage medium for software or data entities. Example: A hard disk drive or CD-ROM, which contains multiple files, loads, data bases, etc.

Media

Devices or materials which act as a means of transferal or storage of software; for example, programmable read-only memory, magnetic tapes or disks, etc.

Media Set Part

Definition is required.

APPENDIX B
GLOSSARY

Network Switch

ARINC 664, Part 7 network device which performs traffic policing and filtering, and forwards packets towards their destination End-Systems.

Port

A port on a component is a connection to an external bus.

Primary Load Port

A component's primary load port is the primary data loading bus port intended for use in the aircraft environment

Redundancy Management

ARINC 664, Part 7 Ports, Links and Switches are duplicated for redundancy. Redundant frames are managed for integrity.

Software

Data or code (executable or not) that defines, controls or is used by its "target hardware" to perform its function.

Shop Loading

Shop loading is performed in a maintenance environment while the component is not installed on an aircraft.

Shop Replaceable Unit

A unit or assembly that is intended to be replaced only in a repair shop environment, not on the aircraft. It is usually a subcomponent of a component or LRU.

Technical Support and Data Package

For more information, reference **ARINC Report 625:** *Industry Guide for Component Test Development and Management*

Unit Under Test (UUT)

One common industry term for the object of component testing is Unit Under Test (UUT). In this context, unit generally refers to the component; however, may sometimes refer to the sub-assembly of a component, or a device or part within the component or sub-assembly.

Virtual Link

A Virtual Link defines a unidirectional (logical) connection from one source End-System to one or more destination End-Systems. They perform Unicast or Multicast communications.

APPENDIX C TEMPORARY RESEARCH DATA

This appendix is a temporary section dedicated for recording drafting information and reference sources. When the document is mature, this section will be removed.

- ARINC 847 dMat
- RAMMap v1.32 By Mark Russinovich Published: November 1, 2013
- Skybrary, The single point reference for aviation safety knowledge
- Guidance for Supplying Off-Aircraft Component Software Loading Specification Requirements
- Shop Data Load Support Specification Term AI:Ted or whoever TITLE
- Component Data Loading Specification (CDLS)
- Reference Cabinet, shared Modules, LRM
- The Airbus IMA concept is based on “shared Modules”. A module focused approached has been preferred compared with the previous concept of “Cabinet”. Its key features are:
    - ARINC 600 IMA Module packaging connected to A664, P7 network
    - Robust partitioning in computing resource and communications
    - Determinism of application execution and data exchanges
    - Standardized Application Programming Interface (API) examples to avoid
    - Obsolescence impacts on applications
    - Conventional equipment’s mixable