Innovative Control Systems, Inc.
10801 N. 24th Ave. Suite 103
Phoenix, AZ 85029 U.S.A.
www.icsaero.com +01-602-861-6984 Voice +01-602-588-9440 Fax

Standard Hand Held Portable Enclosure

Optional Flight Test Enclosure with Optional Compact Flash Recording

DataTap-10
ASCB versions A/B/C/D & Enhanced-ASCB Telemetry and Acquisition Interface System and Bus Recorder

User's Manual
REV T
December 13, 2016

For further information or questions contact:

Innovative Control Systems, Inc.
10801 N. 24th Ave Suite 103
Phoenix, Arizona 85029
U.S.A.

+01-602-861-6984 Voice
+01-602-588-9440 FAX
www.icsaero.com

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. SCOPE

This document defines the Setup, Installation, and Use of the DataTap-10 ASCB-D & Enhanced ASCB NIC Replacement and Telemetry/Acquisition/Simulator Interface (TIS/SIU) for both the Honeywell ASCB-D and Enhanced ASCB buses. This includes electrical, mechanical, and software interface requirements of the DataTap-10 Platform. This system was developed by Innovative Control Systems, Inc. to provide an ASCB-D Bus to User Telemetry-Acquisition-Simulator System Interface.

1.1. References

- IBM Technical Reference Personal Computer AT (6280070, S229-9611-00)

  IBM, P.O. Box 10659,
  Riverton, New Jersey 08076-0659
  U.S. 1-800-IBM-JOURnals (1-800-426-5687) or U.S. 1-609-786-1714,

- Peripheral Component Interconnect Bus Specification, rev 2.1 edition.

  PCI Special Interest Group
  PO Box 14070
  Portland, Oregon 97214.

- Personal Computer Bus Standard P996, Draft 2.00, Jan. 18, 1990 IEEE

1.2. Abbreviations

BIOS	Basic Input/Output Operating System
COTS	Commercial Off-The-Shelf
CSA	Canadian Standards Association
DMA	Direct Memory Address
EIA	Electronic Industry Association
EMI	Electromagnetic Interference
HSD	High-Speed Serial Data Interface
Hz	Hertz
IEC	International Electrotechnical Commission
ISA	Industry Standard Architecture
PCI	Peripheral Component Interconnect
TCP/IP	Transmission Control Protocol/Internet Protocol
TIU	Test Interface Unit (Honyewell)
UDP	User Datagram Protocol

1.3. Copyrights and Trademarks

AMD	Advanced Micro Devices Corporation.
ARINC	Aeronautical Radio, Inc.
Honeywell	Honeywell Corporation, Business & Commuter Aviation Systems
IBM	International Business Machines
Intel	Intel Corporation
Microsoft	Microsoft Corporation
MSDOS	Microsoft Corporation

2. DataTap-10 Top Level Functionality Overview

The DataTap-10 is a versatile and small replacement for a number of ASCB-D related systems designed and built by ICS over a 12-year period. This one adapter can replace the functionality of

- Two PCI-NICs (Honeywell TIU support required)
- A Dual Channel ASCB Data Bus Reader (with FlightLine Software)
- A Dual Channel Simulator Interface Unit (SIU) (with ICS SIU platform)
- A Telemetry / Acquisition Interface System (TIS) (no additional software required)

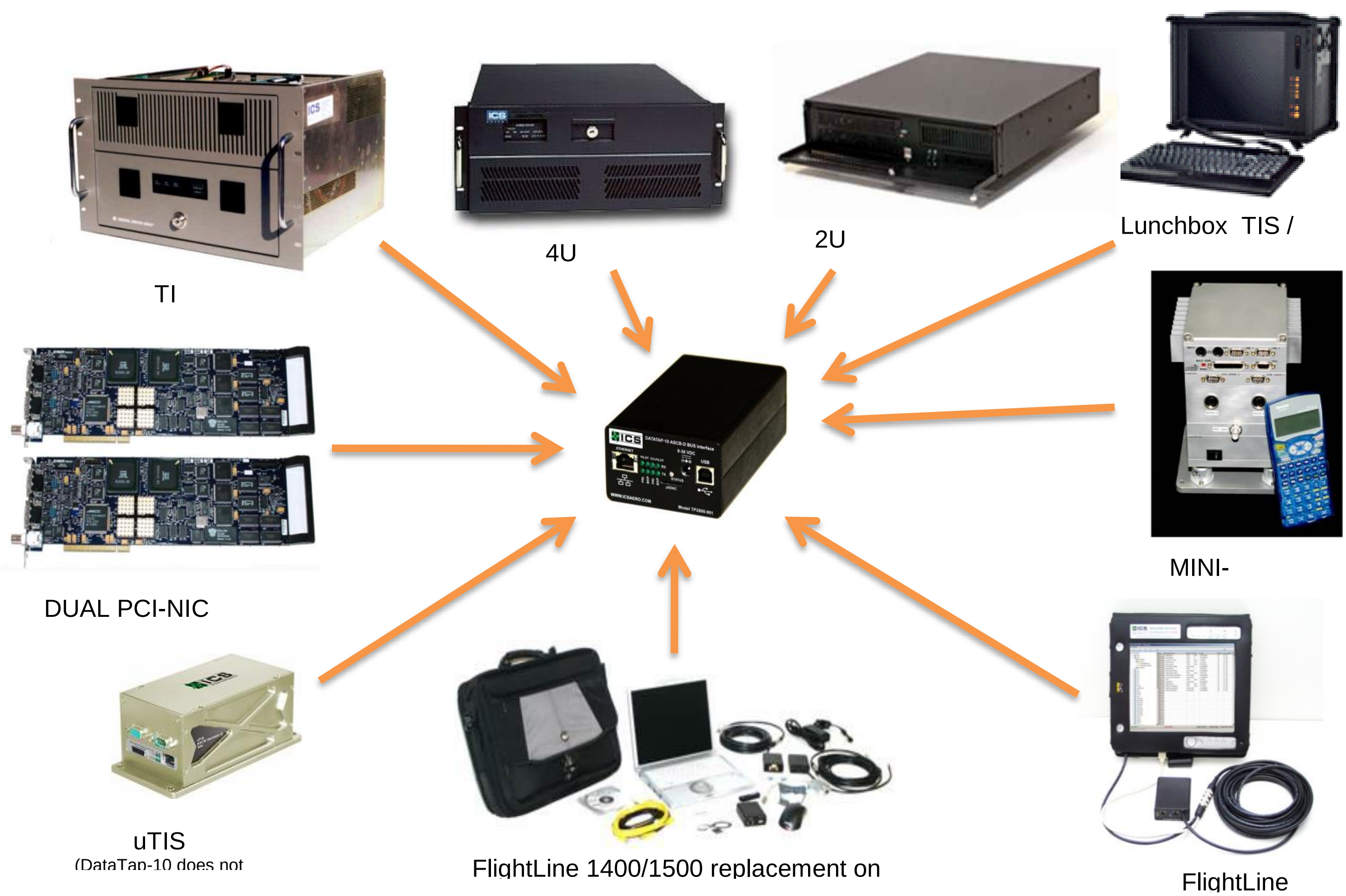


Figure 1 DataTap-10 Functionality Overview

3. Electrical Interface Connectors (All DataTap-10 Platforms)

The DataTap-10 module has two front panel power connectors. The connector to the Left is dedicated to the Ethernet interface and uses a MagJack L829-1J1T-43 Integrated Connector Module. The pinout of this connector is according to the established RJ-45 standard and shown in **Table 3 1000BaseT RJ45 Connectons** below. An Ethernet connection and either a USB or Optional Aircraft power connection is required to use the DataTap-10 with a laptop or other computer.The USB connection is used ONLY for power. You can optionally power the DataTap-10 through the 9-36VDC connection shown.

Figure 2 I/O Connectors

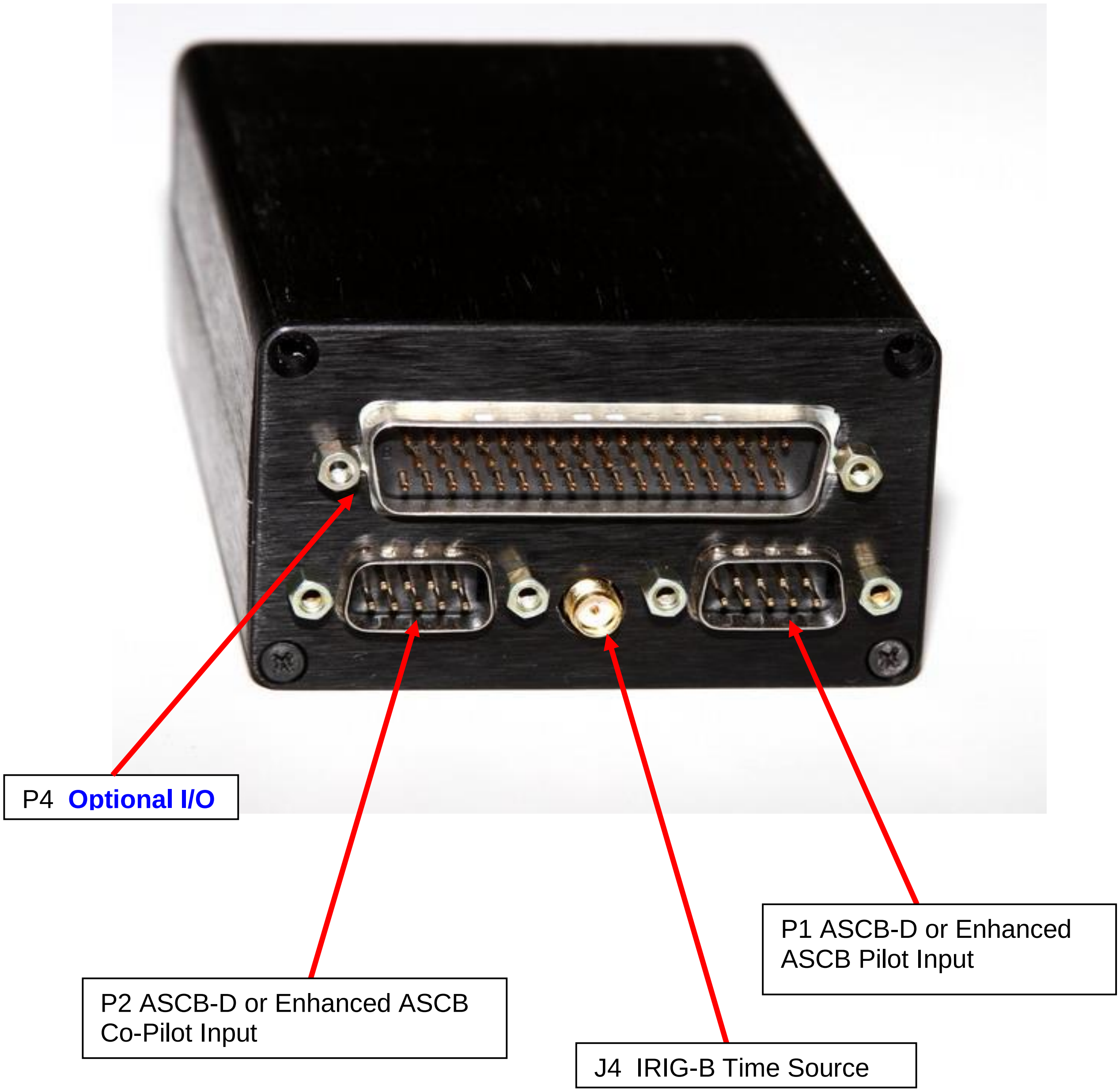


Figure 3 ASCB and IRIG Connections

3.1. 9-36VDC Power Input Locking Barrel Connector

Power may be supplied to the DataTap-10 through either the USB port (5VDC <1 Amp) or through the External Aircraft / Automotive power input (9-36VDC <1 Amp) connectors on the front panel as well as from the optional DB-50 Connector (P4) on the rear panel (when equipped). Power supply through these connectors is not mutually exclusive. Both sources may be connected. The 9-36 input is regulated to 5.0VDC and is combined with the USB 5V power which is usually somewhat less than 5.0V. With both sources connected, the power is drawn from the source higher in voltage which is usually the 9-36VDC input. Connecting both sources is a convenient method of backing up the operation of the DataTap-10. The transition from one power source to the other is completely seamless.

Aircraft or Automotive Power is supplied through a locking barrel connector on the DataTap-10 main board. The DataTap-10 part is a CUI, Incorporated PJ-059BH.

The required mating cable assembly is available from Digi-Key and other suppliers. Tensility International Corp CA-2223 is one such mating Cable. See Digi-Key Part Number CP-2223-ND The center Pin is the positive power supply and the outer barrel is the negative supply. The input voltage is 9-36 VDC and the current is less than 1 amp even at a 9V input voltage.

Figure 4 DataTap-10 Power Connector

Figure 5 Mating Cable Assembly

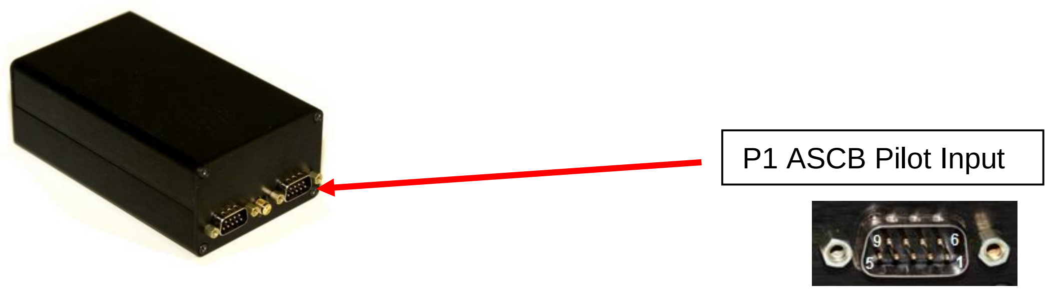
Digikey Number CP-2223-ND

3.2. USB Type B Connector

The DataTap-10 is equipped with a USB interface with a type B Female connector. This interface is only used by the DataTap-10 as a convenient source of power. If the USB connector is plugged into a PC or Laptop USB port, the DataTap-10 will be able to run from the power available at that connector and will not need an external source. Depending on the configuration of the DataTap-10, it may be necessary to use two type A connectors on the Laptop or PC being used as a source of power. ICS supplies a Dual Type A male to Type B Female USB cable with the unit for this purpose. A typical USB port will supply 500ma at approximately 5 VDC. The DataTap-10 will draw 450ma to 900ma depending on the configuration of the DataTap-10. DataTap-10 units that transmit on the ASCB-D bus will definitely draw more than 500 ma during transmit and will require external power or a dual type A USB connection. The USB interface of the DataTap-10 is not presently used except for power. All communication to and from the DataTap-10 is via Ethernet.

3.3. DataTap-10 Pilot Bus Connector

The right most connector at the rear of the DataTap-10 contains the Pilot Primary and Pilot Backup Bus ASCB connections. This connector is a standard 9-pin Male D-sub connector and has the following pin out/ electrical definition. The mating connector required at the user's equipment is a DB-9 Female with metal back shell.



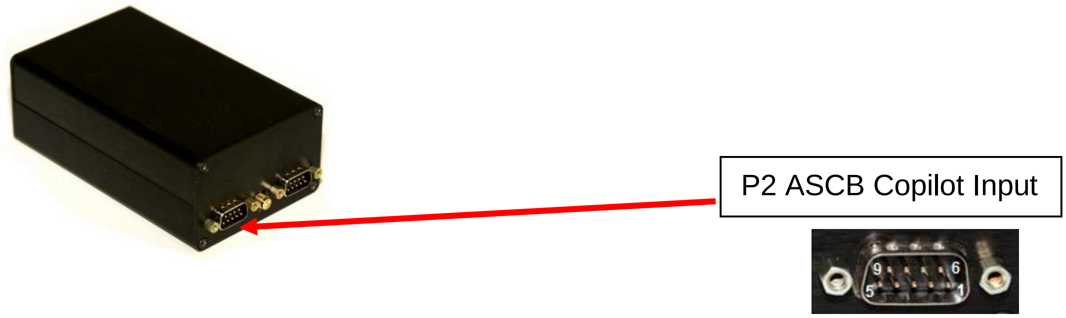
The signals that are indicated in blue are those connected in the ASCB Interface application. Others should be left unconnected.

Pin #	Signal Name	Description
1	**ASCB_PILOT+**	**Pilot Primary ASCB bus, non-inverting wire**
2	ARINC-825_CAN1-	ARINC-825 / CAN CH1 – inverting wire do not connect
3	N.C	No Connection
4	**ASCB_PILOTBKP+**	**Pilot Backup ASCB bus, non-inverting wire**
5	N.C.	No Connection
6	**ASCB_PILOT–**	**Pilot Primary ASCB bus, inverting wire**
7	ARINC-825_CAN1+	ARINC-825 / CAN CH1 – non-inverting wire do not connect
8	**ASCB_PILOTBKP–**	**Pilot Backup ASCB bus, inverting wire**
9	ARINC-825/CAN1_GND	Signal Ground for Isolated CAN / ARINC-825 do not connect

Table 1 ASCB Pilot Bus Connections

3.4. DataTap-10 Secondary Bus Connector

The right most connector at the rear of the DataTap-10 contains the Copilot Primary and Copilot Backup Bus ASCB connections. This connector is a standard 9-pin Male D-sub connector and has the following pin out/ electrical definition. The mating connector required at the user's equipment is a DB-9 Female with metal back shell.



The signals that are indicated in blue are those connected in the ASCB-D Interface application. Others should be left unconnected.

Pin #	Signal Name	Description
1	**ASCB_COPILOT+**	**Copilot Primary ASCB bus, non-inverting wire**
2	ARINC-825_CAN2-	ARINC-825 / CAN CH1 – inverting wire do not connect
3	N.C	No Connection
4	**ASCB_COPILOTBKP+**	**Copilot Backup ASCB bus, non-inverting wire**
5	N.C.	No Connection
6	**ASCB_COPILOT–**	**Copilot Primary ASCB bus, inverting wire**
7	ARINC-825_CAN2+	ARINC-825 / CAN CH1 – non-inverting wire do not connect
8	**ASCB_COPILOTBKP–**	**Copilot Backup ASCB bus, inverting wire**
9	ARINC-825/CAN2_GND	Signal Ground for Isolated CAN / ARINC-825 do not connect

Table 2 ASCB Copilot Bus Connections

3.5. P1 & P2 Primary & Secondary Bus Mating Connector Information

The DataTap-10 is connected to the ASCB bus using two 9-pin D sub connectors. The connectors on the DataTap-10 are male. The following part numbers reflect only a couple of suitable connector types. These are by no means the only acceptable connectors and are provided for information only.

Option-1

Qty	Tyco Part No.	Digi-Key Part No.	Description
2	5-747905-2	A32510-ND	9-Pin Female Connector
2	5748676-1	A34126-ND	Die Cast Shield

Option 2 - Crimp on, snap in pins (for crimping wires to pins) Quantities shown are for 1 DataTap-10 Interface (2 Connectors)

Qty	Tyco Part No.	Digi-Key Part No.	Description
2	1757820-1	A32575-ND	9-Pin Female Connector Housing
18	205090-2	205090-2-ND	Male Crimp on Pins
2	5748676-1	A34126-ND	Die Cast Shield

Optionally you may need to order one or more of the following if your technicians do not have the tooling for these. (Most will have so you may want to ask).

Qty	Tyco Part No.	Digi-Key Part No.	Description
1	91067-2	A1010-ND	Snap In Pin Insertion/Extraction Tool

Qty	Norcomp Part No.	Digi-Key Part No.	Description
1	170-702-170-000	T1000M-ND	TOOL HAND CRIMP MACHINED CONTACT

3.6. 1000/100/20 Mb/s RJ-45 Ethernet Connector

The DataTap-10 processing module supports 1000BaseT, 100BaseT and 10BaseT signaling. 1000BASE-T (also known as IEEE 802.3ab) is a standard for gigabit Ethernet over copper wiring. It requires, at least Category 5 cable (the same as 100BASE-TX), but Category 5e (Category 5 enhanced) or Category 6 cable may also be used and CAT 6 is recommended. 1000BASE-T requires all four pairs to be present and is far less tolerant of poorly installed or inferior wiring than 100BASE-TX.

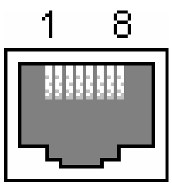


Table 3
1000BaseT RJ45
Connectons

Pin	Name	Description	TIA/EIA 568A	TIA/EIA 568B
1	BI_DA+	Bi-directional pair A+ (tranceive)	white/green	white/orange
2	BI_DA-	Bi-directional pair A- (tranceive)	green	orange
3	BI_DB+	Bi-directional pair B+ (receive)	white/orange	white/green
4	BI_DC+	Bi-directional pair C+	blue	blue
5	BI_DC-	Bi-directional pair C-	white/blue	white/blue
6	BI_DB-	Bi-directional pair B- (receive)	orange	green
7	BI_DD+	Bi-directional pair D+	white/brown	white/brown
8	BI_DD-	Bi-directional pair D-	brown	brown

Pin	Name	Description	EIA/TIA 568A cable colors	EIA/TIA 568B or AT&T 258A cable colors
1	TX+	Transmit Data+	White with green stripe	White with orange stripe
2	TX-	Transmit Data-	Green with white stripe or solid green	Orange with white stripe or solid orange
3	RX+	Receive Data+	White with orange stripe	White with green stripe
4	n/c	Not connected	Blue with white stripe or solid blue	Blue with white stripe or solid blue
5	n/c	Not connected	White with blue stripe	White with blue stripe
6	RX-	Receive Data-	Orange with white stripe or solid orange	Green with white stripe or solid
7	n/c	Not connected	White with brown stripe	White with brown stripe
8	n/c	Not connected	Brown with white stripe or solid brown	Brown with white stripe or solid brown

Table 4 100BaseT and 10BaseT wiring

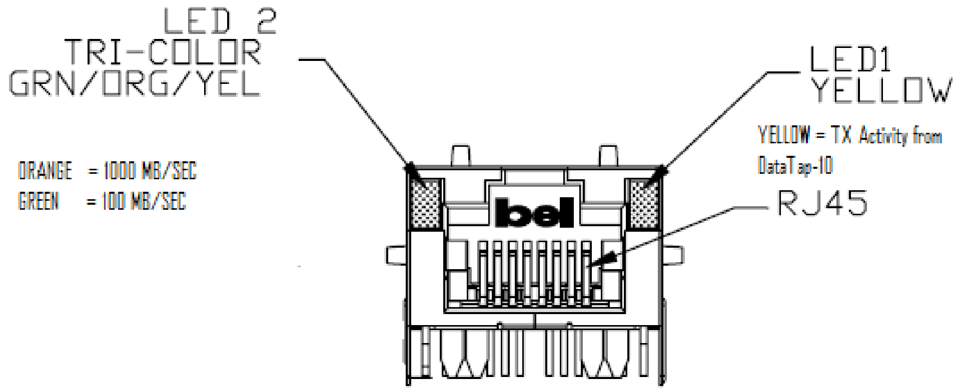


Figure 6 ***RJ-45 Connector Indicator Light Definitions***

ICS recommends the use of the Omron Tough Latch Style of RJ-45 connectors with the DataTap-10 when used in airborne acquisition systems. See image below. These connectors are better suited for harsh environments, but are not up to IP67 standards. They have the advantage of working with existing RJ-45 jacks.

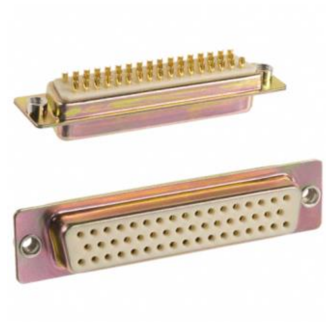
Figure 7 Omron Tough Latch RJ-45 Connectors

3.7. OPTIONAL I/O Connector P4 (Standard and Dassault Specific TTL I/O)

The DataTap-10 processing module can be ordered with Optional ARINC-429, Discrete IN, Discrete OUT, and Analog INPUT I/O. This option has it's I/O routed on a DB-50 Male connector (on the DataTap-10) when the option is installed. The mating FEMALE connector is available from several manufacturers in solder cup and crimp varieties.

Please note the pin defintions below in BOTH Table 5 and Table 7 DataTap-10 Optional I/O Connections (P4) Dassault Specific TTL Output" There are two sets of definitions for this connector depending on the end use configuration. Check with the factory if you are unsure of your end configuration. We can tell you by serial number of the unit the end configuration.

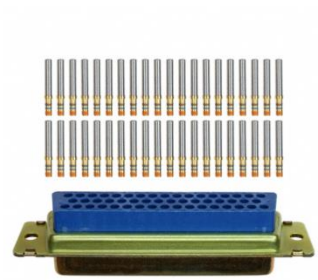
Option-1 Solder Cup



Qty	Norcomp Part No.	Digi-Key Part No.	Description
1	172-E50-201R011	5150FEA-ND	50-Pin Female Connector

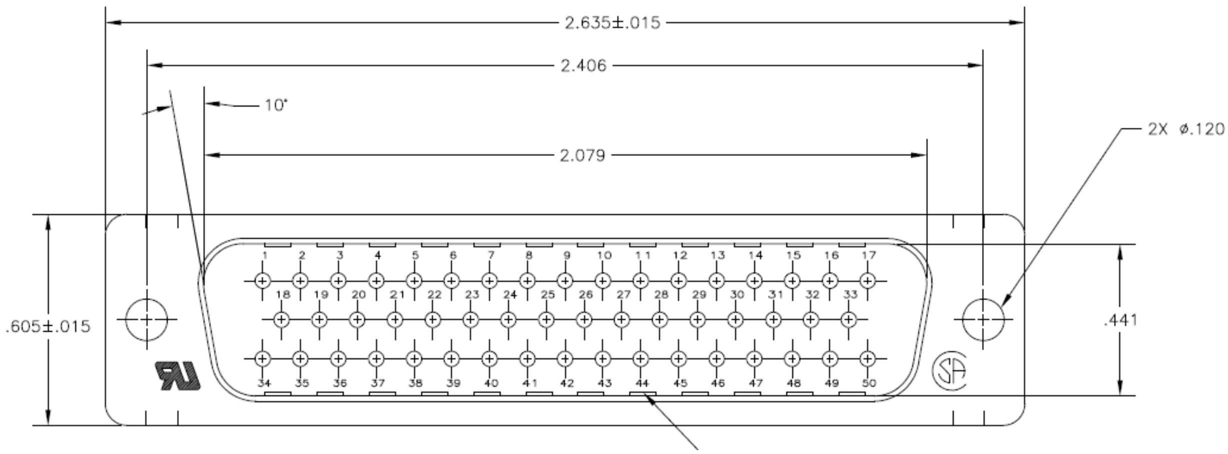
Qty	TE Part No.	Digi-Key Part No.	Description
1	5745175-3	A34169-ND	Die Cast Shield

Option 2 - Crimp on, snap in pins (for crimping wires to pins)



Qty	TE Part No.	Digi-Key Part No.	Description
1	M24308/2-5	A34462-ND	50-Pin Female with pins
1	5745175-3	A34169-ND	Die Cast Shield

Table 5 P4 Maiting Connector for Optional I/O

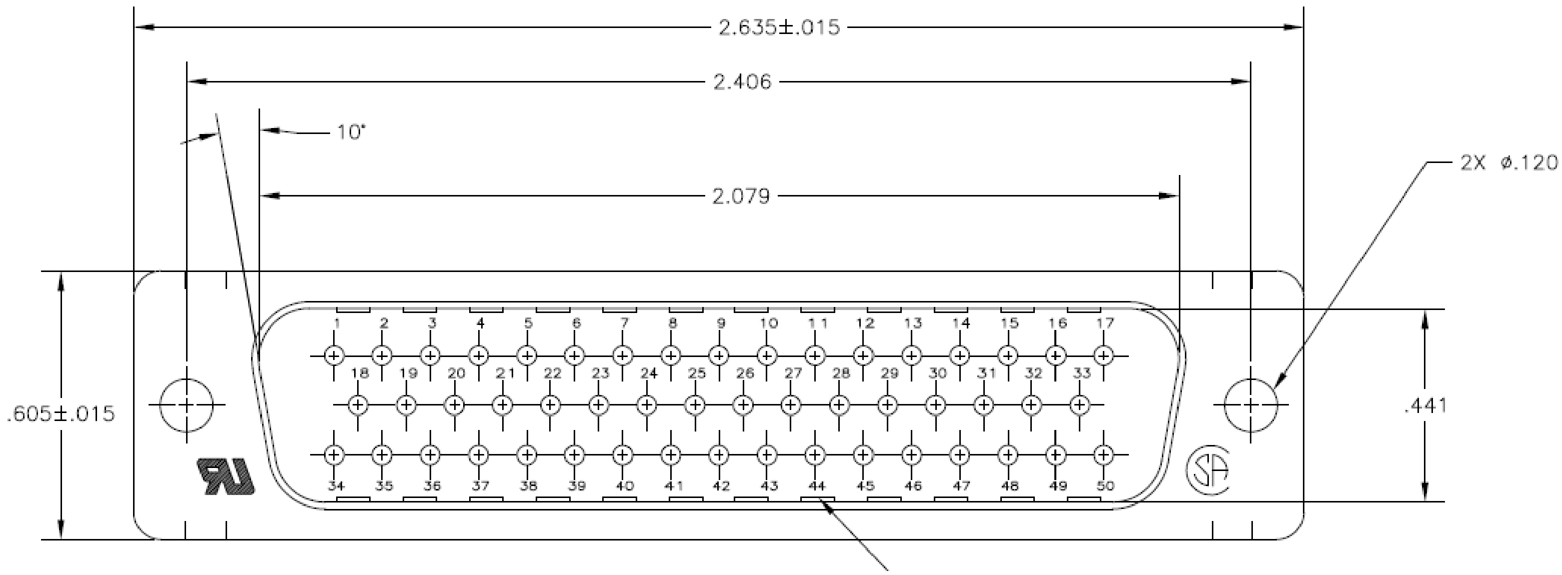


3.8. OPTIONAL I/O Connector P4 Pin Mapping (Standard DataTap-10 Product)

Pin #	Signal Name	Description
34	9-36VDC - PWR	Optional power feed -9-36 VDC Input Return
35	9-36VDC - PWR	Optional power feed -9-36 VDC Input Return
36	9-36VDC - PWR	Optional power feed -9-36 VDC Input Return
48	ANALOG IN 0 (IRIG-B)	Optional Tie point for IRIG-B Input (same as J4 SMA)
49	9-36VDC +PWR	Optional power feed +9-36 VDC Input Aircraft Power
50	9-36VDC + PWR	Optional power feed +9-36 VDC Input Aircraft Power
1	A429 CH1 RX A	ARINC 429 Input 1 A side **(When Option Installed)**
18	A429 CH1 RX B	ARINC 429 Input 1 B side (**(When Option Installed)**
2	A429 CH2 RX A	ARINC 429 Input 2 A side **(When Option Installed)**
19	A429 CH2 RX B	ARINC 429 Input 2 B side **(When Option Installed)**
3	A429 CH3 RX A	ARINC 429 Input 3 A side **(When Option Installed)**
20	A429 CH3 RX B	ARINC 429 Input 3 B side **(When Option Installed)**
4	A429 CH4 RX A	ARINC 429 Input 4 A side **(When Option Installed)**
21	A429 CH4 RX B	ARINC 429 Input 4 B side **(When Option Installed)**
6	A429 CH1 TX A	ARINC 429 Output 1 A side **(When Option Installed)**
23	A429 CH1 TX B	ARINC 429 Output 1 B side **(When Option Installed)**
5	A429 CH2 TX A	ARINC 429 Output 2 A side **(When Option Installed)**
22	A429 CH2 TX B	ARINC 429 Output 2 B side **(When Option Installed)**
37	ANALOG IN 1	Analog Input 1 +/-10 VDC 16-bits
38	ANALOG IN 2	Analog Input 2 +/-10 VDC 16-bits
39	ANALOG IN 3	Analog Input 3 +/-10 VDC 16-bits
40	ANALOG Common	Analog Input Common
46	DISC OUT + SUPPLY	Discrete Outputs Positive Power Rail 9-60V

Pin #	Signal Name	Description
47	DISC OUT + SUPPLY	Discrete Outputs Positive Power Rail 9-60V
44	DISC OUT - SUPPLY	**Discrete Outputs Return Power Common**
45	DISC OUT - SUPPLY	**Discrete Outputs Return Power Common**
11	DISCOUT 1 P SRC	Discrete Output 1 + Open SRC **(PPS_OUT) Pull to Positive power on Pin 46,47 (Active High) when enabled in system.cfg file**
28	DISCOUT 1 N DRN	Discrete Output 1 – Open Drain **(PPS_OUT) Pull to GND on Pin 44,45 (Active Low) when enabled in system.cfg file**
12	DISCOUT 2 P SRC	Discrete Output 2 + Open SRC **(Health_Status_OUT) Pull to Positive power on Pin 46,47 (Active High = Healthy) when enabled in system.cfg file**
29	DISCOUT 2 N DRN	Discrete Output 2 – Open Drain **(Health_Status_OUT)) Pull to GND on Pin 44,45 (Active Low = Healthy) when enabled in system.cfg file**
13	DISCOUT 3 P SRC	Discrete Output 3 + Open SRC **(Memory_Card_Status_OUT) Pull to Positive power on Pin 46,47 (Active High = < configured time remaining) when enabled in system.cfg file**
30	DISCOUT 3 N DRN	Discrete Output 3 – Open Drain **(Memory_Card_Status_OUT)) Pull to GND on Pin 44,45 (Active Low = < configured time remaining) when enabled in system.cfg file**
14	DISCOUT 4 P SRC	Discrete Output 4 + Open SRC (Pull to PWR 46,47) **(Recording_Status_OUT) Pull to Positive power on Pin 46,47 (Active High = recording active) when enabled in system.cfg file**
31	DISCOUT 4 N DRN	Discrete Output 4 – Open Drain (Pull to ground 44,45) **(Recording_Status_OUT) Pull to GND on Pin 44,45 (Active Low = recording active) when enabled in system.cfg file**
7	DISCIN 1 P	Discrete Input 1 + 3.3-60V
24	DISCIN 1 N	Discrete Input 1 - 3.3-60V
8	DISCIN 2 P	Discrete Input 2 + 3.3-60V
25	DISCIN 2 N	Discrete Input 2 - 3.3-60V
9	DISCIN 3 P	Discrete Input 3 + 3.3-60V
26	DISCIN 3 N	Discrete Input 3 - 3.3-60V
10	DISCIN 4 P	Discrete Input 4 + 3.3-60V
27	DISCIN 4 N	Discrete Input 4 - 3.3-60V
15	RS232-1 TX OUT	RS-232 Port 1 TX Out (from DataTap-10)
32	RS232-1 RX INP	RS-232 Port 1 RX Inp (into DataTap-10)
16	RS232-2 TX OUT	RS-232 Port 2 TX Out (from DataTap-10)
33	RS232-2 RX INP	RS-232 Port 2 RX Inp (into DataTap-10)
17	Reserved N.C.	Do Not Use. Reserved for expansion I/O
41	Reserved N.C.	Do Not Use. Reserved for expansion I/O
42	Reserved N.C.	Do Not Use. Reserved for expansion I/O
43	Reserved N.C.	Do Not Use. Reserved for expansion I/O

Table 6 DataTap-10 Optional I/O Connections (P4) Standard DataTap-10 Product



3.9. OPTIONAL I/O Connector P4 Pin Mapping (Dassault Specific TTL Output)

Pin #	Signal Name	Description
34	9-36VDC - PWR	Optional power feed -9-36 VDC Input Return
35	9-36VDC - PWR	Optional power feed -9-36 VDC Input Return
36	9-36VDC - PWR	Optional power feed -9-36 VDC Input Return
48	ANALOG IN 0 (IRIG-B)	Optional Tie point for IRIG-B Input (same as J4 SMA)
49	9-36VDC +PWR	Optional power feed +9-36 VDC Input Aircraft Power
50	9-36VDC + PWR	Optional power feed +9-36 VDC Input Aircraft Power
1	A429 CH1 RX A	ARINC 429 Input 1 A side **(When Option Installed)**
18	A429 CH1 RX B	ARINC 429 Input 1 B side (**(When Option Installed)**
2	A429 CH2 RX A	ARINC 429 Input 2 A side **(When Option Installed)**
19	A429 CH2 RX B	ARINC 429 Input 2 B side **(When Option Installed)**
3	A429 CH3 RX A	ARINC 429 Input 3 A side **(When Option Installed)**
20	A429 CH3 RX B	ARINC 429 Input 3 B side **(When Option Installed)**
4	A429 CH4 RX A	ARINC 429 Input 4 A side **(When Option Installed)**
21	A429 CH4 RX B	ARINC 429 Input 4 B side **(When Option Installed)**
6	A429 CH1 TX A	ARINC 429 Output 1 A side **(When Option Installed)**
23	A429 CH1 TX B	ARINC 429 Output 1 B side **(When Option Installed)**
5	A429 CH2 TX A	ARINC 429 Output 2 A side **(When Option Installed)**
22	A429 CH2 TX B	ARINC 429 Output 2 B side **(When Option Installed)**

37	ANALOG IN 1	Analog Input 1 +/-10 VDC 16-bits
38	ANALOG IN 2	Analog Input 2 +/-10 VDC 16-bits
39	ANALOG IN 3	Analog Input 3 +/-10 VDC 16-bits
40	ANALOG Common	Analog Input Common
46	**Do Not Connect**	**Do Not connect to this pin. (+5 VDC DataTap-10 Internal Supply)**
47	**Do Not Connect**	**Do Not connect to this pin. (+5 VDC DataTap-10 Internal Supply)**
44	DISC OUT Common	**Discrete Outputs Ground Return (Common)**
45	DISC OUT Common	**Discrete Outputs Ground Return (Common)**
11	DISCOUT 1 P SRC	Discrete Output 1 + Open SRC **(PPS_OUT) Pull to Positive power on Pin 46,47 (Active High) when enabled in system.cfg file**
28	DISCOUT 1 N DRN	Discrete Output 1 – Open Drain **(PPS_OUT) Pull to GND on Pin 44,45 (Active Low) when enabled in system.cfg file**
12	DISCOUT 2 P SRC	Discrete Output 2 + Open SRC **(Health_Status_OUT) Pull to Positive power on Pin 46,47 (Active High = Healthy) when enabled in system.cfg file**
29	DISCOUT 2 N DRN	Discrete Output 2 – Open Drain **(Health_Status_OUT)) Pull to GND on Pin 44,45 (Active Low = Healthy) when enabled in system.cfg file**
13	DISCOUT 3 P SRC	Discrete Output 3 + Open SRC **(Memory_Card_Status_OUT) Pull to Positive power on Pin 46,47 (Active High = < configured time remaining) when enabled in system.cfg file**
30	DISCOUT 3 N DRN	Discrete Output 3 – Open Drain **(Memory_Card_Status_OUT)) Pull to GND on Pin 44,45 (Active Low = < configured time remaining) when enabled in system.cfg file**
14	DISCOUT 4 P SRC	Discrete Output 4 + Open SRC (Pull to PWR 46,47) **(Recording_Status_OUT) Pull to Positive power on Pin 46,47 (Active High = recording active) when enabled in system.cfg file**
31	DISCOUT 4 N DRN	Discrete Output 4 – Open Drain (Pull to ground 44,45) **(Recording_Status_OUT) Pull to GND on Pin 44,45 (Active Low = recording active) when enabled in system.cfg file**
7	DISCIN 1 P	Discrete Input 1 + 3.3-60V
24	DISCIN 1 N	Discrete Input 1 - 3.3-60V
8	DISCIN 2 P	Discrete Input 2 + 3.3-60V
25	DISCIN 2 N	Discrete Input 2 - 3.3-60V
9	DISCIN 3 P	Discrete Input 3 + 3.3-60V
26	DISCIN 3 N	Discrete Input 3 - 3.3-60V
10	DISCIN 4 P	Discrete Input 4 + 3.3-60V
27	DISCIN 4 N	Discrete Input 4 - 3.3-60V
15	RS232-1 TX OUT	RS-232 Port 1 TX Out (from DataTap-10)
32	RS232-1 RX INP	RS-232 Port 1 RX Inp (into DataTap-10)
16	RS232-2 TX OUT	RS-232 Port 2 TX Out (from DataTap-10)

33	RS232-2 RX INP	RS-232 Port 2 RX Inp (into DataTap-10)
17	Reserved N.C.	Do Not Use. Reserved for expansion I/O
41	Reserved N.C.	Do Not Use. Reserved for expansion I/O
42	Reserved N.C.	Do Not Use. Reserved for expansion I/O
43	Reserved N.C.	Do Not Use. Reserved for expansion I/O

Table 7 DataTap-10 Optional I/O Connections (P4) Dassault Specific TTL Output

3.10. Hardware Discrete Health and Status Outputs

The DataTap-10 can be programmed to output four heath and status signals via the discrete outputs defined in **Table 6** and **Table** 7. The I/O option must be installed to access the discrete outputs. If the Health and Status Output is enabled via the system.cfg file, the discrete outputs have the indicated meaning:

Discrete 1: PPS Output, 1 Hz, PPS output available for both IRIG-B and PTP synchronization modes Signal goes high at the one-second rollover point and remains high for 100 milliseconds.

Discrete 2: Health Status Output, Healthy is a logical AND of all of the following conditions being met:

1. *The NIC Registry is present and valid.*
2. *The ESCAPE Registry is present and valid.*
3. *Micro-SD Flash Memory Card is present and initialized correctly.*
4. *TIS.CFG file is present on the Micro-SD card.*
5. *SYSTEM.CFG file is present on the Micro-SD card with no unrecognized parameters or parameter values.*
6. *The Analog-to-Digital converter initialized correctly.*
7. *ASCB packets are being received on Left and Right ASCB-D bus inputs.*
8. *The ASCB connections are correct based on the received NIC IDs.*
9. *The ASCB Bus is synchronized to the master timing NIC.*
10. *No overflows have occurred during recording.*
11. *There are no recording errors during recording. Note that a Micro-SD card becoming full during recording is considered an error.*

Note: Numbers 7 through 9 can only be true if the apparent version of ASCB correctly matches that set in the SYSTEM.CFG file. This is due to the encoding differences between Enhanced-ASCB and ASCB-D. If the configured version is incorrect, no data or frame sync messages will be received and conditions 7 through 9 will all be false.

Discrete 3: Memory Card Status Output, Less than N Minutes remaining. The number of minutes is programmable and defined in the system.cfg file.

Discrete 4: Recording Status. This discrete output is asserted when recording in progress and is deasserted when recording is not in progress.

Note: The discrete outputs of the DataTap-10 are available as both pull to Ground and pull to a supplied voltage (or in optional TTL levels) Due to this, Both Active High and Active Low variations of the discrete outputs are available depending on which output pin of connector P4 is used.

3.11. IRIG-B Timing Input Connector

The DataTap-10 processing module has a built-in IRIG-B Timing Input. The connector used is a TE Connectivity (TYCO) 5-1814400-1 SMA Jack (Female Socket) on the DataTap-10. The internal IRIG timestamp clock free runs in the absence of a timecode input, and subsequently syncs to the timecode input if a valid format IRIG-B source becomes available. The IRIG timecode and clock is used to timestamp data captured by the DataTap-10. A suitable mating pigtail is available from Amphenol Connex part "245101-01-06.00 CABLE ASSY SMA STR PLUG RG316 6" This allows a transition to a BNC plug which is more commonly used in IRIG-B installations.

J4 IRIG-B Time Code Input

J4 Mating Adapter Amphenol Connex part "245101-01-06.00

NOTE:

The IRIG-B input requires a transformer coupled IRIG-B signal. Some lower cost IRIG-B sources use a capacitive coupled IRIG-B output. The IRIG-B input of the DataTap-10 is an input to a Delta-Sigma A/D converter that will cause a capacitive output to charge up and float 2.5V above ground. In these situations, we recommend a simple 600 Ohm to 600 Ohm Audio Isolation transformer be placed on the output of the IRIG timing source for all users to benefit from, or alternately, that the transformer be placed on the IRIG-B input of the DataTap-10. IRIG sources that are transformer coupled such as Symmetricom units do not require any additional isolation. An **optional** internal transformer may be installed and is installed in all Dassault airborne flight test enclosure applications. You must request this option if you are not Dassault and are not specifically ordering a Dassault Flight Test Platform configuration.

4. Electrical Interface Connectors (DataTap-10 Optional Flight Enclosure)

The DataTap-10 TIS modules enclosed in the Rugged Flight Test enclosure has an additional front panel power connector. The connector located on the Flight Test Enclosure is a **LEMO EXG-0B-303-HLN.**

The pin out used is as follows:

LEMO Pin #	Signal name
1	INPUT POWER +
2	INPUT POWER -
3	CHASSIS/GND

Figure 8 OPTIONAL LEMO Power Connections (Dassault FlightTest Enclosure)

5. DataTap-10 to ASCB Version D Aircraft Connections

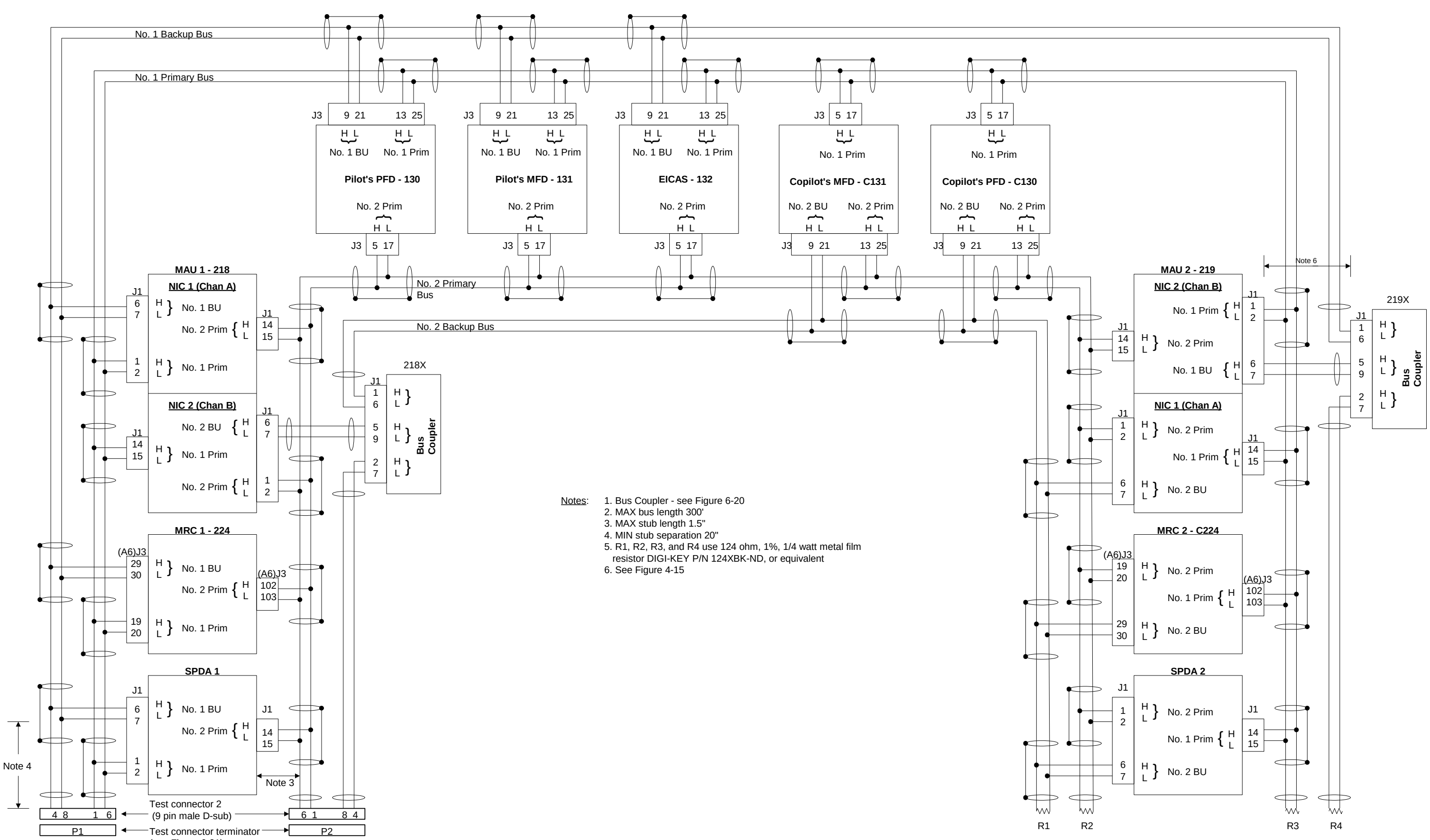
The DataTap-10 system must be connected to Aircraft's Pilots and Co-Pilots ASCB D busses in order for the system to be able to receive data from both sides of the aircraft. There are many ways to accomplish this and the final wiring of the DataTap-10 to the aircraft (during flight test) and will ultimately be documented and communicated to you by Honeywell Inc. Note that it is not necessary to connect both sides of the aircraft to the DataTap-10 platform if only data from one side of the aircraft is needed, or if there is only one ASCB-D bus in the aircraft.

The diagrams and information in this section is provided only to help you understand how this unit might be connected to the aircraft wiring. You should not use this information to wire your aircraft.

5.1. Aircraft Test Connector Placement

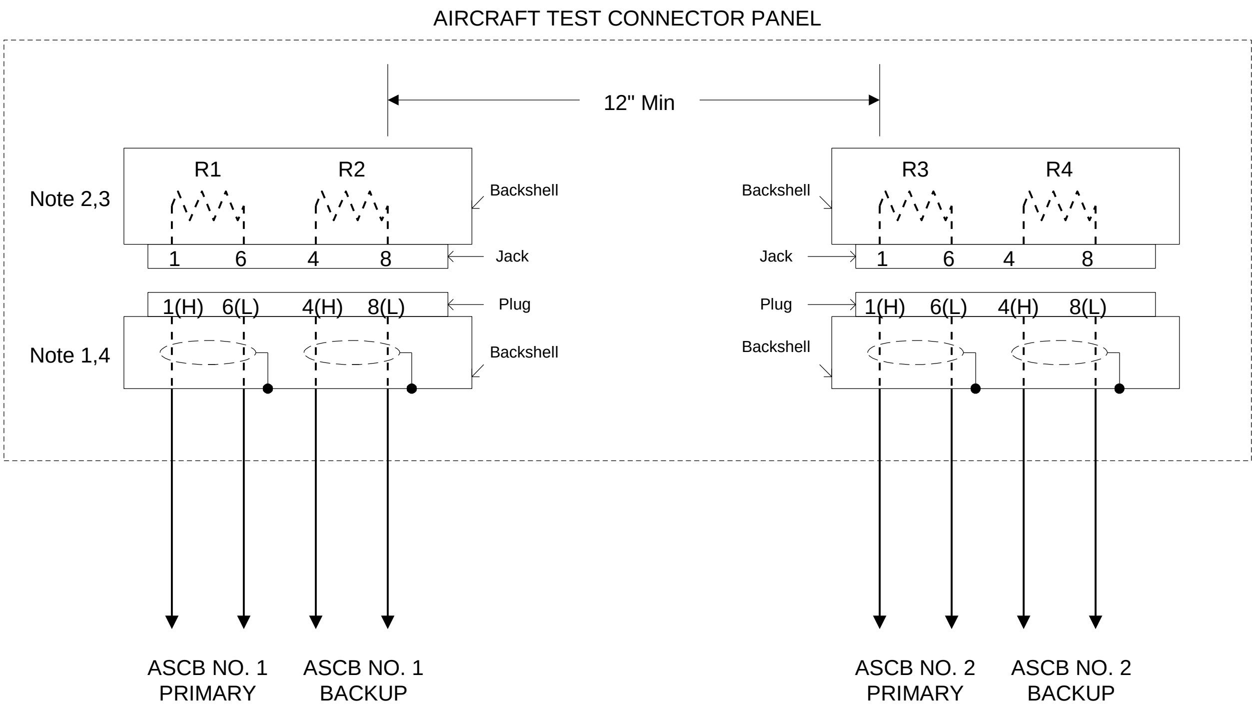
In some flight test arrangements, ASCB D bus test connectors may be installed for connection to a Honeywell TIU, an ICS DataTap-10 platform, or other possible equipment requiring connection to the ASCB D buses.

The diagram below shows one possible configuration of ASCB D buses in an aircraft, and the placement of the ASCB D bus test connectors.



5.2. Aircraft Test Connector Wiring (Typical)

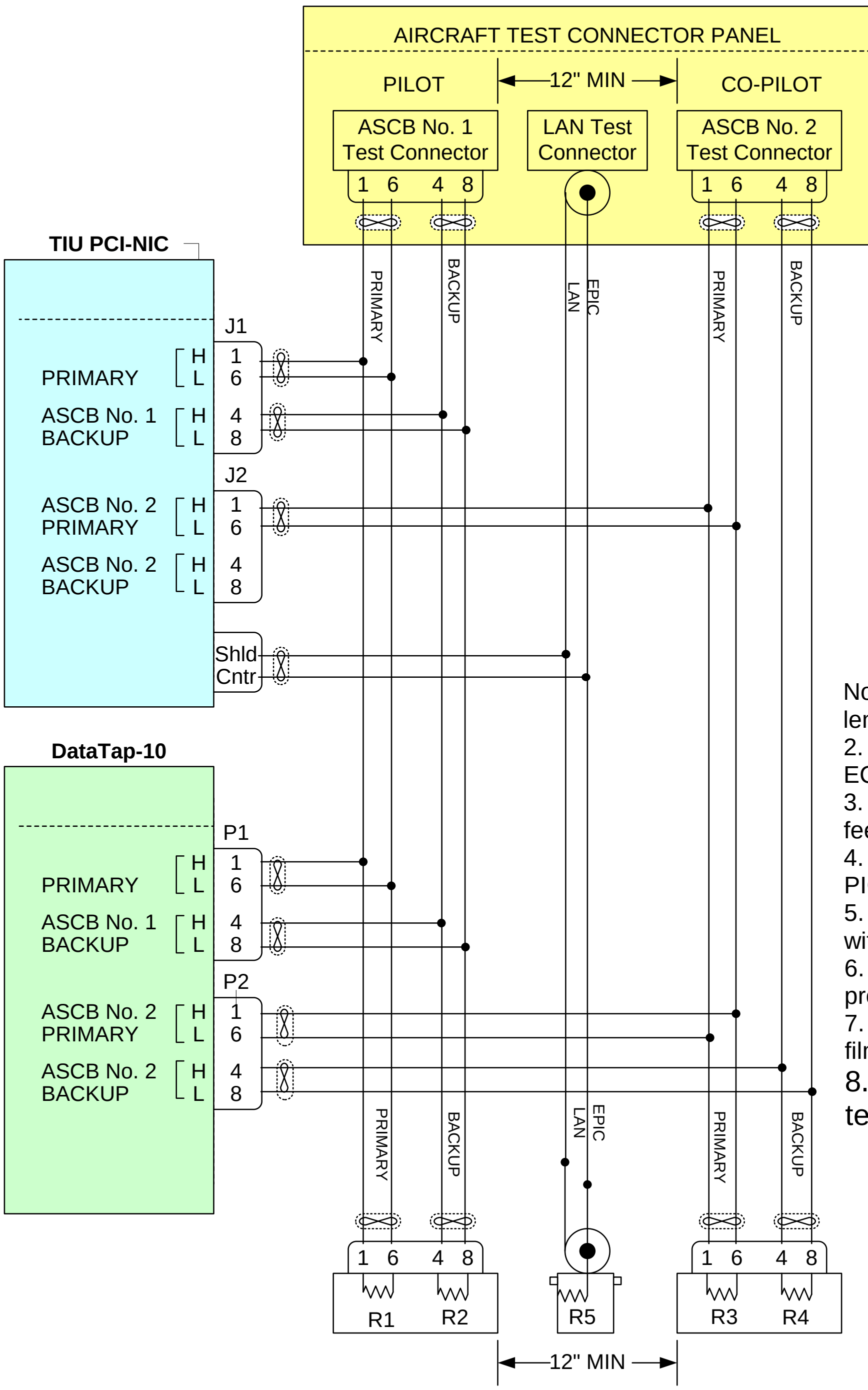
(NOT the same on all aircraft installations. May be aircraft manufacturer defined.)



Notes:
1. Use 9 pin male plug D-sub connector with 0.120 inch mounting through holes for aircraft test panel mounted ASCB test connector (Positronic P/N RD9M00000, requires 9 each contacts, Positronics P/N MC6020D-14). Mount connector on rear of test panel. Use Glenair flange mount backshell part number 557T110M1F03C.
2. Use 9 pin female receptacle with 0.120 inch mounting through holes - Positronics P/N RD9F00000 (requires 9 each contacts Positronics P/N FC6020D-14). Use sealed backshell Glenair P/N 550-200M1R1K. This backshell is designed to mate with a rear mounted plug installed on a 0.31 inch thick panel.
3. R1, R2, R3, and R4 use 124 ohm, 1%, 1/4 watt metal film resistor. DIGI-KEY P/N 124XBK-ND, or equivalent.
4. Fabricate the ASCB wires as per step 1 of the section 4.5.1. Attach the part number MC6020D-14 contacts directly to the ASCB wire (Don't need to use the spliced 22 AWG wire extensions). Ground the ASCB shield pigtail wires (each 2 inch max length) to the backshell strain relief.

5.3. BENCH Test Connector Wiring (TIU, DataTap-10 platform)

The following diagram shows one possible wiring configuration commonly used on Integration Test Facility benches. It is NOT the preferred method of connection in Aircraft Flight Test equipment. See the ICS preferred wiring for aircraft flight test on the next page.



Notes: 1. ASCB and ASCB Test Cable has a maximum length of 300 feet.

2. ASCB wire shall be Emteq cable P/N D12502-200 or ECS cable P/N 322402.

3. LAN and LAN Test Cable has a maximum length of 300 feet.

4. LAN wire shall be Emteq cable P/N TFLX130-100 or PIC cable P/N S86208.

5. Test Connectors shall be 9 pin D-sub connectors (mate with Positronic P/N RD9F00000).

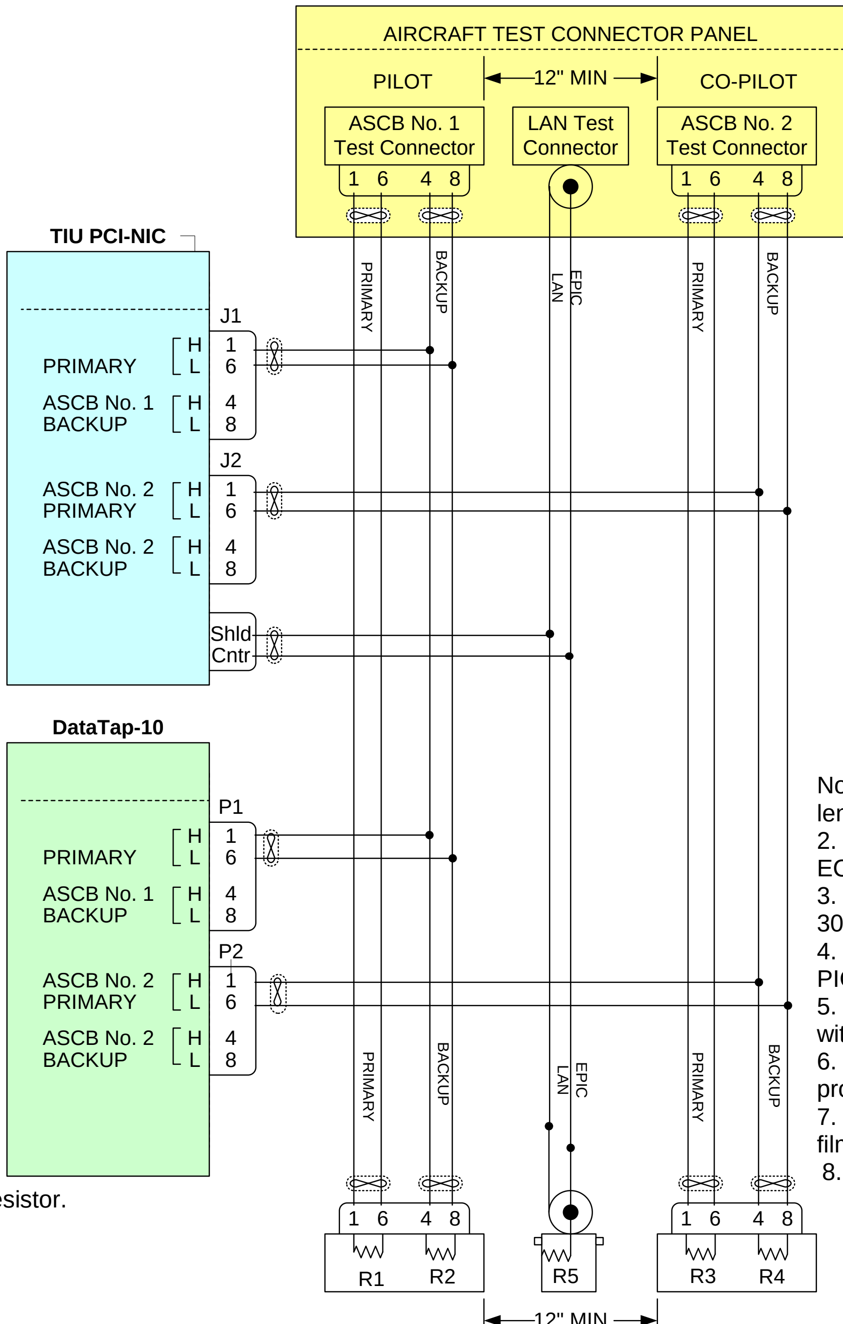
6. Test Connector termination shall be sealed in HIRF proof container.

7. R1, R2, R3 and R4 use 124 ohm, 1%, ¼ Watt metal film resistors (DIGI-KEY P/N 124XBK-ND).

8. R5 use BNC plug with integral 50 ohm termination resistor.

5.4. Preferred AIRCRAFT Test Connector Wiring (TIU, TIS)

The following diagram shows the ICS preferred wiring for aircraft installations of TIU and DataTap-10 units that cannot transmit on the ASCB-D busses. ICS does not recommend using a Honeywell TIU and PCI-NIC to transmit on an ASCB-D bus, however, this is done in some applications for Auto Pilot tuning by Honeywell. The following aircraft wiring will **NOT** work properly for auto pilot tuning applications. For applications involving auto pilot tuning, contact Honeywell for TIU wiring guidelines. You will notice that there are no connections to the PRIMARY aircraft busses, only the backup busses. It is important when connecting to only the backup busses that connector J1 of the PCI-NIC gets data on the PRIMARY bus pins as well as the BACKUP bus pins. If this is not done, and only the backup bus pins are connected, the PCI-NIC will periodically test the primary bus pins for data and when this occurs, data can be lost or corrupted if the pins are not connected.



Notes: 1. ASCB and ASCB Test Cable has a maximum length of 300 feet.
2. ASCB wire shall be Emteq cable P/N D12502-200 or ECS cable P/N 322402.
3. LAN and LAN Test Cable has a maximum length of 300 feet.
4. LAN wire shall be Emteq cable P/N TFLX130-100 or PIC cable P/N S86208.
5. Test Connectors shall be 9 pin D-sub connectors (mate with Positronic P/N RD9F00000).
6. Test Connector termination shall be sealed in HIRF proof container.
7. R1, R2, R3 and R4 use 124 ohm, 1%, ¼ Watt metal film resistors (DIGI-KEY P/N 124XBK-ND).
8. R5 use BNC plug with integral 50 ohm termination

resistor.

6. DataTap-10 Functionality

The DataTap-10 is small, versatile, FPGA-based system designed for the following:

- Receiving and physical-layer decoding of ASCB-A, ASCB-B, ASCB-C, ASCB-D, and eASCB data from the pilot and co-pilot bus.
- Periodic Device Driver implemented in hardware for ASCB-D and eASCB.
- Transmission onto the ASCB-D and eASCB bus in a NIC-only mode or PDD and NIC mode.
- Ability to decode and send out 8192 parameter values over UDP for ASCB-D and eASCB.
- Ability to change up to 8192 transmitted parameter values via UDP when in the hardware-based PDD and NIC transmission mode for ASCB-D and eASCB.
- IRIG-B decode
- PTPv2 (IEEE1588v2) decode with hardware time-stamping using an ICS-developed 10/100/1000 Ethernet MAC.
- Ability to record raw ASCB, decoded ASCB, ARINC-429, and/or ARINC-717 to microSD or (optionally) Compact Flash.
- 4 channels of ARINC-429 receive
- 2 channels of ARINC-429 transmit with the use of a hardware auto-transmit module (ATM).
- 1 channel of ARINC-717 decode supporting 64 wps to 8192 wps
- 4 analog inputs
- 4 discrete outputs
- 4 discrete inputs

6.1. FPGA Block Diagram

A simplified block diagram of the DataTap-10's Xilinx Spartan-3 FPGA is shown in Figure 9. Note that all modules shown are hardware modules implemented in the FPGA fabric. The software running on processors 0 and 1 handle the Ethernet processing, recording, playback, and ASCB frame tick management, while all ASCB encoding and decoding is handled in hardware.

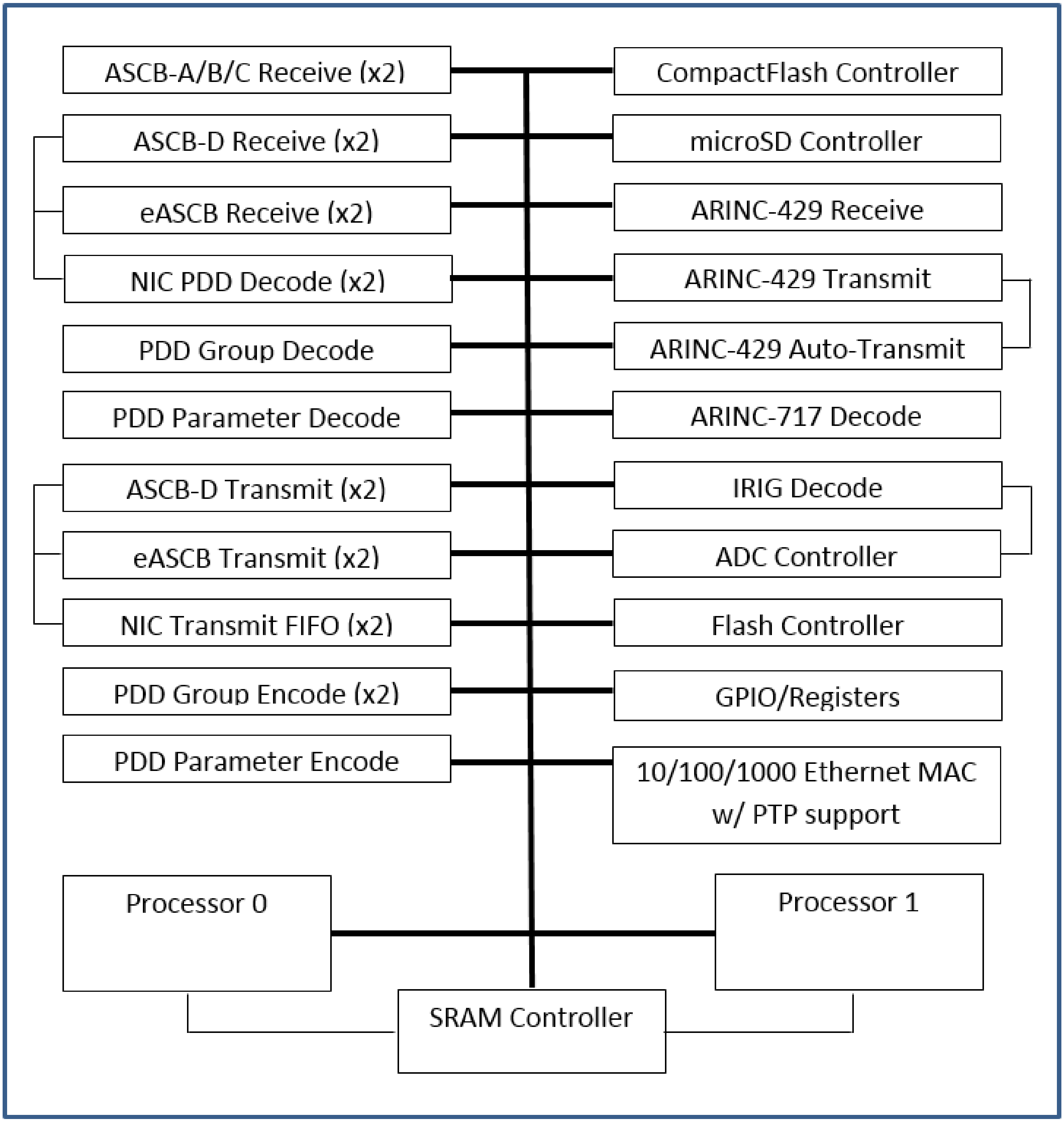


Figure 9 DataTap-10 FPGA Block Diagram

6.2. Operating Modes

The DataTap-10 can operate in 6 different modes, as described in the sections below. With the exception of Host Tx Playback Mode, which must be initiated with FlightLine, the initial operating mode on power-up is configured in the system configuration file. Once communication with the DataTap-10 has been established, the current operating mode can be changed using FlightLine (with the exception of switching between "NIC-Only Transmission Mode" and "PDD and NIC Transmission Mode", which can only be done in the system configuration file). Note that Receive-Only, NIC-Only Transmission, and PDD and NIC Transmission Modes are all considered LIVE modes since they interface to a live ASCB bus with other NICs potentially transmitting on the bus. Rx Raw Playback, Tx Raw Playback and Host Tx Playback Modes are all considered PLAYBACK modes since they are driven from previously recorded data.

6.2.1. Receive-Only Mode

In Receive Only Mode, the DataTap-10 will only receive data off of the ASCB bus. It will not transmit onto the ASCB bus at any time. On power-up, the DataTap-10 will listen to the ASCB bus for frame synchronization messages. If these messages are found, the DataTap-10 will then begin processing ASCB data. If they are not found or if the synchronization messages ever go away, then the DataTap-10 will remain in or return to the listening state.

This mode is entered whenever the DataTap-10's assigned NIC IDs are both zero and when playback mode is not active. Note that the assigned NIC IDs can be changed through FlightLine.

6.2.2. NIC-Only Transmission Mode

NIC-Only Transmission Mode (section 6.4.1) is where the DataTap-10 acts as one or two MAU NICs and receives full frame buffers of data from a host (i.e. FlightLine). This data is not modified at all by the DataTap-10 and is only transmitted out in the correct time slot as directed by the NIC Registry. In this mode, the DataTap-10 will listen to the ASCB bus for frame synchronization messages on power-up. If these messages are found, the DataTap-10 will then begin processing ASCB data and transmitting out its own synchronization messages if the assigned NIC IDs indicate that of a timing NIC. If no frame synchronization messages are found but the assigned NIC IDs indicate that of a timing NIC, then the DataTap-10 will arbitrate to become the master timing NIC on the bus. If no frame synchronization messages are found and the assigned NIC IDs do NOT indicate that of a timing NIC, then the DataTap-10 will remain in the listening state.

This mode is entered whenever at least one of the DataTap-10's assigned NIC IDs is not zero, the HW_TX_PDD_ENABLE is set to FALSE in the system configuration file, and playback mode is not active.

6.2.3. PDD and NIC Transmission Mode

PDD and NIC Transmission Mode (section 6.4.2) is where the DataTap-10 uses its hardware-based PDD to generate the frame buffers of data for each side of the bus rather than having to receive them from a host in real time. By eliminating any real-time behavior required by a host to generate the frame buffers, this mode is much more robust than the NIC-Only Transmission Mode. Parameter UDP Input Datagrams (section 9.5) can be synchronously or asynchronously sent to the DataTap-10 to change any desired transmitted parameter values. This mode follows the same bus synchronization procedure as the NIC-Only Transmission Mode.

This mode is entered whenever at least one of the DataTap-10's assigned NIC IDs is not zero, the HW_TX_PDD_ENABLE is set to TRUE in the system configuration file, and playback mode is not active.

6.2.4. Rx Raw Playback Mode

In Rx Raw Playback Mode (section 6.5), the DataTap-10 reads a recorded file set from its microSD or optional CompactFlash card and synchronously feeds it into the Receive Engine. With the exception of the Decoded Parameter UDP Output Datagrams (which can intentionally be changed using the Consume TIS Configuration File), this mode results in identical UDP output as when the recording was taken. As such, it will appear to FlightLine as if the data is coming from a live ASCB bus. Note that this mode does not transmit anything onto the ASCB bus and any ASCB input data is ignored.

Rx Raw Playback Mode is entered by setting PLAYBACK_MODE to RXRAW in the system configuration file or it can be entered dynamically from any mode using FlightLine.

6.2.5. Tx Raw Playback Mode

In Tx Raw Playback Mode (section 6.5), the DataTap-10 reads a recorded file set from its microSD or optional CompactFlash card and synchronously feeds it into the Transmit Engine. This reproduces the ASCB bus with the exact timing and data as when the recording was taken. Consequently, no other transmitting NICs should be on the ASCB bus when in this mode. Because the DataTap-10 receives its own transmissions (as all NICs on the bus do), the data will also be fed into the Receive Engine as it is received off of the bus. As such, Tx Raw Playback Mode encompasses the behavior of Rx Raw Playback Mode.

Tx Raw Playback Mode is entered by setting PLAYBACK_MODE to TXRAW in the system configuration file or it can be entered dynamically from any mode using FlightLine. Note that the transmission hardware option must be installed on the DataTap-10 in order for this mode to work.

6.2.6. Host Tx Playback Mode

In Host Tx Playback Mode (section 6.5), the DataTap-10 receives the playback data from FlightLine rather than having to read it from the microSD or optional CompactFlash card. Other than that, this mode is identical to the Tx Raw Playback Mode. Host Tx Playback Mode was developed to prevent a user from having to copy entire recording sets to the DataTap-10's media card, which can be time-consuming. Note that a large playback buffer is used such that FlightLine does not have to operate in real-time. FlightLine aims to keep the buffer half-full to prevent any underruns or overruns.

Host Tx Playback can only be entered using FlightLine. No special configuration is required in the system configuration file and this mode can be entered from any other mode. Note that the transmission hardware option must be installed on the DataTap-10 in order for this mode to work.

6.3. ASCB Reception

6.3.1. ASCB-D and Enhanced-ASCB

The NIC PDD, PDD, and PDD API are all implemented in hardware on the DataTap-10 for reception and decoding of ASCB-D and eASCB data. Software is only involved in decoding of the ASCB frame sync messages and setting up the next frame tick, all of which allows syncing to the ASCB bus. At each frame tick, the DataTap-10 can be configured to send out over UDP multiple frame sync messages (typically used in transmit applications), the raw NIC frame buffers gathered by the NIC PDD, and up to 8192 decoded ASCB parameters. See section 0 for the various UDP output formats sent out by the DataTap-10.

The NIC PDD module uses the NIC Registry programmed into the Flash memory of the DataTap-10 to direct it where the incoming ASCB data should be placed in the NIC frame buffers. It also makes any received frame sync messages available to software for processing.

The PDD module uses the ESCAPE Registry programmed into the Flash memory to direct it how to convert the NIC frame buffers into parameter groups. This process includes checking the integrity of the data as well as the freshness. Both results are placed into the associated parameter group structures.

The PDD API module uses the tis.cfg file located on the DataTap-10's microSD card to direct it how to convert the data from the parameter group structures into decoded ASCB data. This file is typically generated from one of several ICS-developed tools which allow picking of parameters from an ASCB database and placing them into one or more of the available 8192 channels. The decoded parameters and their status may be sent out over UDP by the DataTap-10 and are also available for viewing over Telnet (see section 7.4).

Note: Older DataTap-10 units contain only a 2MB local RAM. In this case, only 2048 parameters can be decoded. A 4MB or 8MB RAM is required to support decoding of 8192 parameters. Please contact ICS for information regarding an upgrade to a 4MB or 8MB local RAM. The detected RAM size can be found on page 3 of the Telnet screens.

6.3.2. ASCB versions A, B, and C

The DataTap-10 allows reception of ASCB versions A, B, and C on both the pilot bus and the copilot bus. The data on all of these busses is contained in 8 application-defined frames that are transmitted every 25ms or at a 40Hz rate. The frames start at frame number 0 and increment to 7 and then repeat. As each frame is received, the DataTap-10 places the data from each bus into separate UDP datagrams with user-defined destination port numbers which are specified in the system configuration file. Once a full frame is received, it then ships the datagrams to one or more Ethernet-based data acquisition platforms for recording and/or decoding.

On the physical bus, data is in a modified HDLC format with Manchester encoding and zero-bit insertion after every 5 consecutive one bits, allowing a flag to be defined with 6 consecutive bits. The flag is 0x7E and starts and ends each message type. For more information, please see Honeywell document EB7020475. The bit rate for ASCB-A/B/C is 666.666 Kb/s. The main purpose of the DataTap-10 is to decode the data at the physical layer and put the received data into a packed UDP datagram to be decoded by the end-user application.

The ASCB-A/B/C UDP datagrams sent out by the DataTap-10 always have a 2048-byte payload which is zero-padded. Each datagram starts with the 0x7E flag of the Frame Start message and ends with the 0x7E flag of the Last User Data message within the same frame. There is also a 16-byte header containing the IRIG time which prepends the data, details of which can be found in section 9.11.

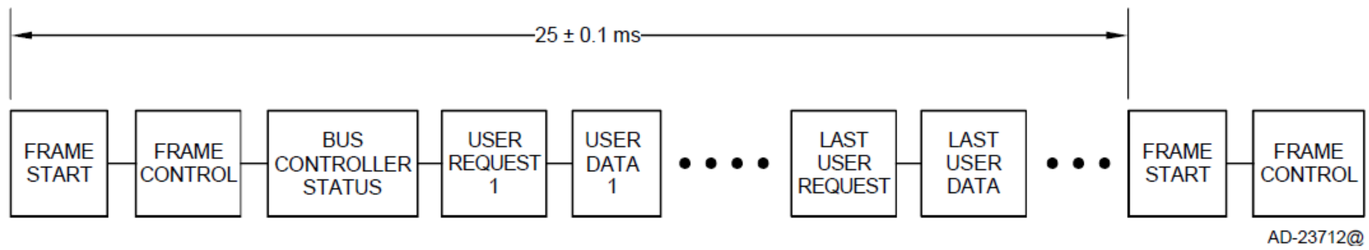


Figure 10 Overall ASCB-A/B/C Frame Message

Each message type has a SYNC at its beginning and a MARK at its end. The SYNC and MARK are not placed in the UDP datagram, but all other message bytes including the 0x7E FLAG bytes are. The detection of back to back 0x7E (0x7E7E) is the end of one message and the start of the next message within the ASCB-A/B/C frame. There are no empty gaps in the UDP message even though there are gaps in time on the ASCB-A/B/C bus between messages.

See the message formats in Figure 11. All messages on the bus are placed in the UDP datagram starting with their 0x7E FLAG and ending with their 0x7E flag.

Note that for ASCB-A, a clock is present on the Backup bus rather than redundant data. The DataTap-10 expects the ASCB data to be on the Primary bus and the ASCB clock to be on the Backup bus when configured for ASCB-A mode.

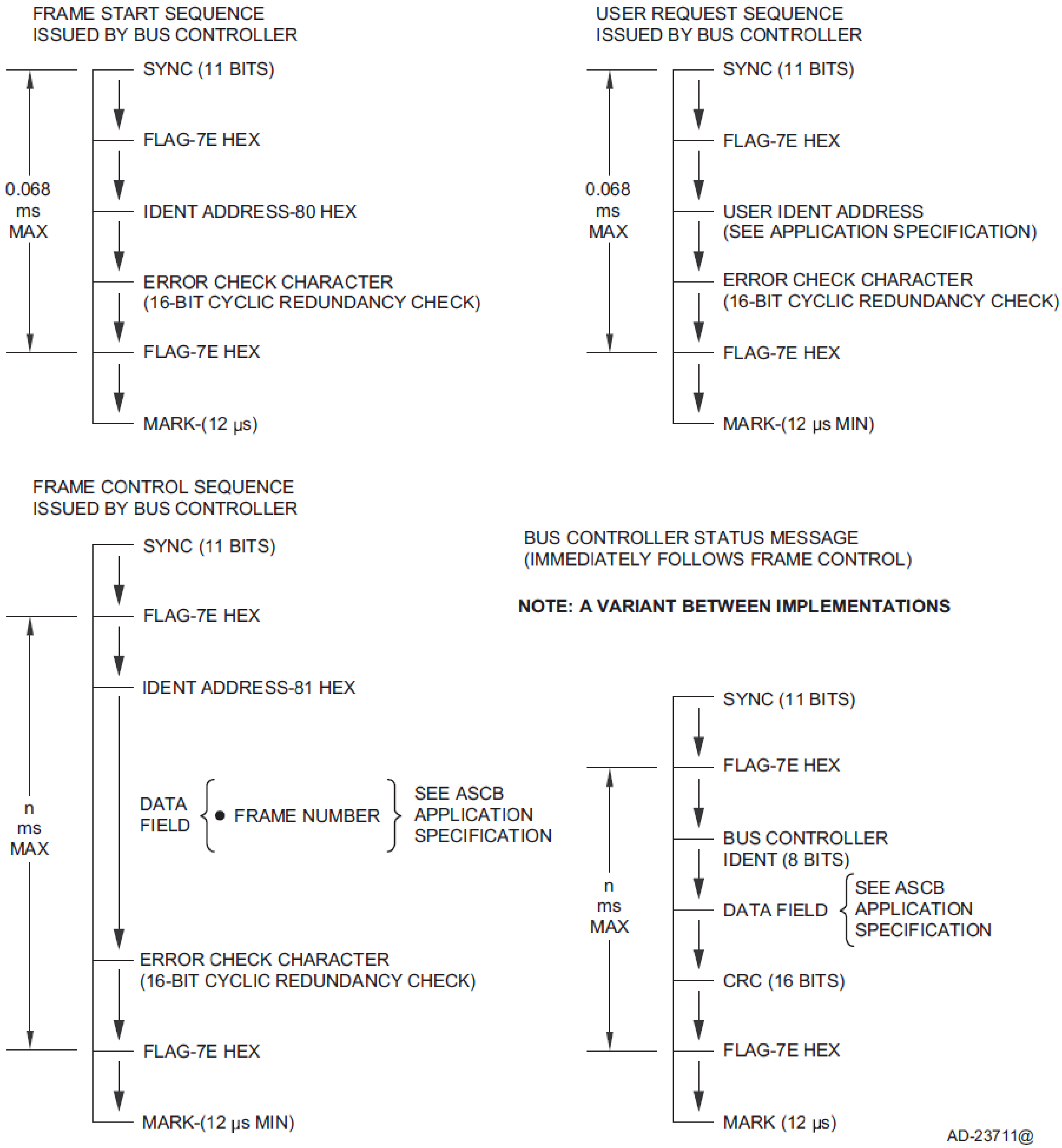


Figure 11 ASCB-A/B/C message formats

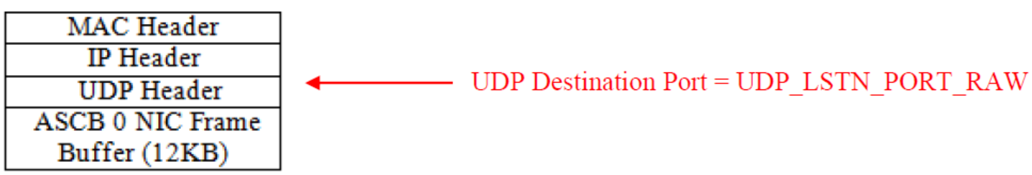
6.4. ASCB Transmission (OPTIONAL)

The DataTap-10 has the (optional) ability to transmit data onto the primary and backup busses of each ASCB bus when configured for ASCB-D or eASCB. It can also act as a timing NIC (or multiple timing NICs) and automatically produce the necessary frame sync messages onto each ASCB bus if the NIC IDs are configured as such. The DataTap-10 has two modes of transmission: 1) NIC mode and 2) PDD and NIC mode. The Transmission option must be purchased in order for transmission to be enabled.

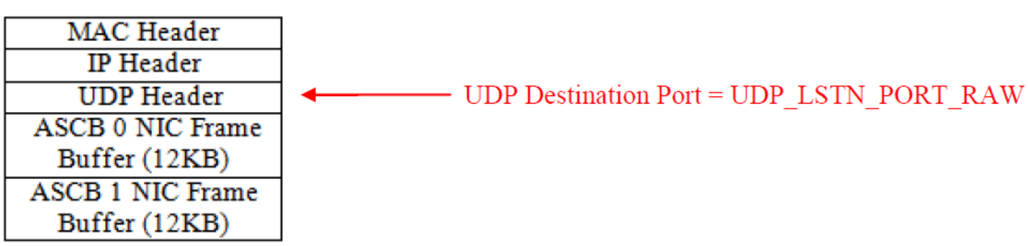
6.4.1. NIC-Only Transmission Mode

In this mode, the DataTap-10 acts as two or more MAU NICs. A host PC must send the required NIC frame buffers (one for each ASCB bus) over UDP to the DataTap-10 each ASCB frame in order to transmit valid ASCB data. The DataTap-10 can act as any desired number of NICs, however, the NIC frame buffers must be created accordingly. The only configuration steps required to enable transmission on the DataTap-10 is to set the NIC IDs. This can be done in the system configuration file or via a UDP command packet (section 9.8).

The data to be transmitted onto each ASCB bus is sent to the DataTap-10 through one 24KB or two 12KB UDP datagrams. The datagrams contain the entire 12KB NIC transmit frame buffer for each bus. The DataTap-10 takes these frame buffers and transmits them onto the appropriate ASCB bus as directed by the NIC Registry. The DataTap-10 listens for the datagrams on the port specified by the UDP_LSTN_PORT_RAW parameter in the system.cfg file. The frame buffer data for ASCB Bus 0 is expected on the specified port and the frame buffer data for ASCB Bus 1 is expected on the port above it (specified port + 1). Alternatively, one 24KB datagram may be sent containing the frame buffer data for both busses (ASCB 0 data followed by ASCB 1 data) on the port specified by the UDP_LSTN_PORT_RAW parameter. This is depicted in the diagrams below.



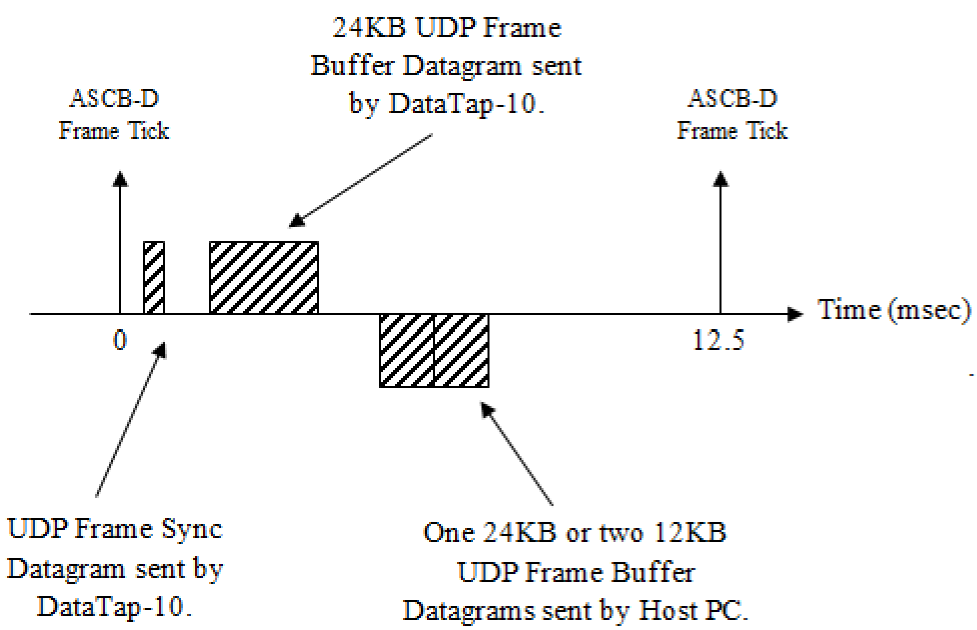
The Frame Buffer data can be sent to the DataTap-10 in the form of two 12KB datagrams sent to successive UDP ports, as shown above.



Alternatively, the Frame Buffer data can be sent to the DataTap-10 in the form of one 24KB datagram, as shown above.

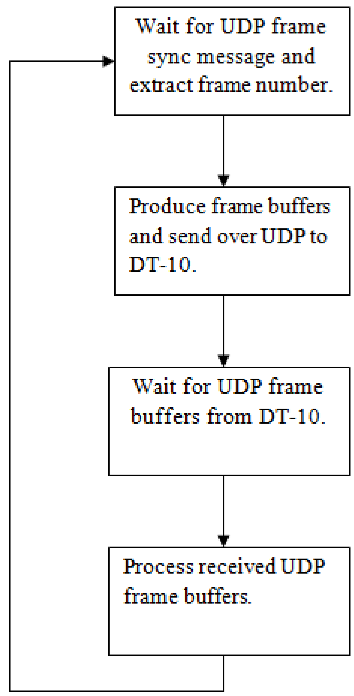
At the start of each ASCB frame, the DataTap-10 sends out a UDP frame sync datagram (see section 9.5) over the Ethernet link. The host PC must first listen for this datagram and extract the frame number. It then produces the frame buffers for ASCB-D transmission based on the extracted frame number. Once the frame buffers have been produced, the host PC sends them over UDP to the DataTap-10. The DataTap-10 must receive the buffers and be able to DMA them into memory before the start of the next ASCB frame. It is expected that this datagram (or datagrams) will be fragmented in order to meet the Maximum Transfer Unit constraint of the Ethernet link. Typically, the last fragment must be received greater than or equal to 200 microseconds before the end of the current frame, assuming all other fragments have already been received and DMA'd into memory.

The sending and receiving of UDP datagrams over the Ethernet link during a typical ASCB-D frame is shown below.



Once the host PC has sent out the datagram(s) over UDP, it can then wait for the UDP frame buffer datagram sent out by the DataTap-10 representing the data received from each ASCB bus, or it can alternatively use a separate thread to wait and process the received datagram. Once the full 24KB of data has been received, the host PC can consume the received frame buffers, if desired. It then waits for the next UDP frame sync datagram to indicate the start of the next ASCB frame.

A flow chart depicting the typical processing steps on the host PC for an ASCB frame is shown below.



6.4.2. PDD and NIC Transmission Mode

In this mode, the DataTap-10 can not only act as multiple MAU NICs, but also up to two MAU modules (one for each ASCB bus). A valid ESCAPE registry must be programmed into the DataTap-10 via the microSD card for each MAU module. Each ESCAPE registry specifies which parameters are produced by the module. The DataTap-10 uses a hardware-based PDD module to produce the parameters. All parameters listed in each ESCAPE registry are produced including valid integrity check and freshness fields.

Any parameter being produced can have its value changed by sending in UDP datagrams to the DataTap-10 (section 9.5). A host has the ability to change the value each ASCB frame by first listening to the Frame Sync Datagram(s) sent out by the DataTap-10 at the start of each frame. The host can then send the required UDP packets. It can alternatively send the packets asynchronously at any desired rate. Note that previous values persist if a new value is not received by the DataTap-10.

The UDP datagrams sent to the DataTap-10 also give the ability to corrupt the integrity check and/or freshness fields associated with a given parameter using the datagram's STATUS fields (section 9.5). Because the integrity check and freshness fields are sent with each group of data on the ASCB bus, corrupting a field for a single parameter will cause corruption for the entire group.

A TIS configuration file must exist on the microSD card for each ASCB bus in order for the DataTap-10 to be able to dynamically change the produced values. The configuration file assigns a channel to each desired parameter. The channel indicates where in the UDP packets the DataTap-10 should fetch each parameter value. The DataTap-10 fetches the parameter values, encodes them, and then transmits them onto the associated ASCB bus. Up to 8192 parameters can be encoded by the DataTap-10.

To set the DataTap-10 into this mode, the “HW_TX_PDD_ENABLE” parameter must be set to “TRUE” in the system.cfg file (see section 7.1.8).

Note: Older DataTap-10 units contain only a 2MB local RAM. A 4MB or 8MB RAM is required in order to support PDD and NIC Mode Transmission. NIC-Only Mode Transmission is still supported with a 2MB RAM, however. Please contact ICS for information regarding an upgrade to a 4MB or 8MB local RAM.

6.5. ASCB Playback

The DataTap-10 can play back ASCB raw recorded data in 3 different modes (section 6.2), making it appear to a host as if the DataTap-10 is receiving live data from the ASCB bus. In all playback modes, the Raw ASCB Frame Buffer UDP Output Datagrams sent out by the DataTap-10 are identical as to when the recording was taken, including the IRIG time. The Decoded Parameter UDP Datagrams, on the other hand, can be configured to contain any desired parameters by simply changing the Consume TIS Configuration file (tis.cfg). Because the raw recording files contain data from the entire bus, any desired parameters on the bus can be selected in the TIS Configuration file when in playback mode. All parameter values contained in the datagrams will be driven from the recorded data.

Note that any of the playback modes can be entered using FlightLine and even though previously recorded data is being used, FlightLine should still be configured for “Live Mode” in the ASCB Receive Configuration Window. This is because from the host’s perspective, the DataTap-10 is still receiving data from a live bus.

In Rx Raw and Tx Raw Playback Modes, the recorded raw data is read from the DataTap-10’s microSD or optional CompactFlash cards. Either mode can be configured in the system configuration file or started through FlightLine. In either case, a mode is chosen as well as the name of the first file in the recording set to play back and the desired IRIG start time. When playback is started, the DataTap-10 will quickly search the file set for the specified IRIG time, if necessary, and begin playback from that point. Playback can be configured to start on power-up in the system configuration file. Once configured, it can also be controlled through Telnet. Note that when starting playback through FlightLine, it provides automatic retrieval of all filenames currently on the selected media card to simplify selection of the desired playback file.

In Host Tx Playback Mode, the data is received from a host (i.e. FlightLine) rather than read from the DataTap-10’s media card. This prevents having to copy entire recording sets to the media card, which can be time-consuming. This mode is otherwise identical to Tx Raw Playback Mode. See section 6.2.6 for more details regarding Host Tx Playback Mode.

Note that Tx Raw Playback and Host Tx Playback are only supported with ASCB-D and eASCB.

6.6. IRIG-B Time Decode

The DataTap-10 IRIG decode logic uses channel 0 of the on-board 16-bit Delta Sigma Analog-to-Digital Convertor (ADC) to sample the incoming IRIG Signal. The input voltage specification of the ADC is +/- 10 VDC and is divided into approximately 300 µVolts / count. The IRIG logic pulls a sample from the ADC at a rate of 12,000 samples per second. This provides 12 samples per millisecond, which is one sine wave cycle of the IRIG-B signal. The logic uses the positive zero-crossing of the signal to extract the highest sample of each sine wave. This sample is compared to a threshold value to eventually determine if a 10-wave cycle indicates an

IRIG zero, an IRIG one, or a reference marker. To determine the threshold value, the logic saves the highest sample taken during each 10-wave cycle. The sample is divided in half to calculate the threshold value to be used during the following cycle. This allows the incoming IRIG signal level to vary from high to very low voltages as long as the 3:1 mark-to-space ratio requirement is met as detailed in the IRIG specification.

Note that IRIG formats B120 to B127 are supported, however, only the time-of-year is decoded and used by the DataTap-10. In the absence of an IRIG signal, the DataTap-10's internal IRIG clock is free-running.

6.7. IEEE 1588-2008 / Precision Time Protocol Version 2

The DataTap-10 implements a slave-only implementation of the IEEE 1588-2008 specification, which defines the Precision Time Protocol (PTP) Version 2. It supports the following modes:

- End-to-End
- Multicast
- One-Step or Two-Step Clocks
- PTP over IPv4/UDP
- PTP over IEEE 802.3

Timestamping is done in hardware at the MAC level and the protocol stack is run in software. The hardware timestamping allows for very accurate alignment with the master clock. An adjustable clock is implemented in hardware and can be aligned to the master using both frequency and phase adjustment.

Frequency alignment to the master is accomplished by measuring the difference between timestamps of successive sync messages from the chosen master. These differences are averaged and fed into a software-driven proportional-integral (PI) controller to control the frequency.

Accurate phase alignment to the master is accomplished by monitoring for sync messages and periodically sending delay request messages as directed by the IEEE specification. This process allows calculation of the mean path delay, which along with the timestamps and any corrections added by end-to-end transparent clocks in the path, is used to calculate the time offset from the master.

The DataTap-10 PTP software stack uses the Best Master Clock algorithm as defined in the IEEE specification whenever an announce message is received from a master. Once a master is selected, the DataTap-10 will listen and process sync messages only from that master. It will periodically send a delay request message and wait for the delay response in order to calculate the mean path delay. The periodicity of the delay request sending is defined by the master clock.

When the timing source is set to PTPv2 rather than IRIG, the PTP time is converted into an IRIG format for all outgoing UDP datagrams set in the TISNATIVE format (see section 0). Note that because PTP time is expressed in Coordinated Universal Time (UTC), the DataTap-10 allows a time zone offset to be specified in the system configuration file such that the time can be adjusted for the current time zone.

6.8. Removable Media Recording

6.8.1. Overview

The DataTap-10 allows raw ASCB, decoded ASCB, ARINC-429, and ARINC-717 data to be recorded in real-time. It supports recording to microSD and **OPTIONALLY** CompactFlash removable media cards. Recording to microSD is standard. Recording to CompactFlash requires an optional daughter card and the Flight Test enclosure which must be purchased from ICS. The DataTap-10 supports hot-insertion of both media types; however, some CompactFlash cards do not support hot-insertion.

Recording can be started via a Telnet session, UDP command packets, or automatically at power-up if specified in the system configuration file. Recording status including the current filename, file size, space remaining, and any errors encountered can be monitored over Telnet (section 7.4) as well as in the Status Output Datagrams. This allows monitoring of any performance degradation of the media cards. Note that all of this information and the ability to start and stop recording is available through FlightLine.

ASCB (raw or decoded), ARINC-429, and ARINC-717 can all be recorded simultaneously, however, only the recommended microSD or CompactFlash cards listed below should be used. In addition, the DataTap-10 MUST be used to format the card when using microSD. This allows the DataTap-10 to place the FAT32 file structures in the recommended locations as listed in the SD Specification, helping to increase the write performance. It also allows the DataTap-10 to issue erase commands to the entire card which further increases performance as well as extends the life of the card, which is described in more detail in section 6.8.4.

Although rare, it is possible for a microSD or CompactFlash card to get corrupted if power is removed while data is being written to the card. This is only an issue if the FAT tables or directory structure are being written to when power is removed. In this case, it could result in total card corruption and is only fixable by reformatting the card. To prevent this, recording can be stopped prior to power being removed. This can be accomplished via a Telnet session, sending a specific UDP command packet (see section 9.8.14), simply connecting to the DataTap-10 via File Transfer Protocol, or using a discrete input (see section 6.14). Note that when recording multiple streams, full clusters of data (typically 32KB) are first buffered up for each data stream prior to being written to the card. As a result, up to 5 seconds' worth of data plus 1 cluster may be lost for each data stream if power is removed in the middle of recording. Note that 5 seconds is the flush period during recording.

Once a recording is complete, the media card may be removed from the unit and placed into a PC for processing (section 7.3.1). The files may alternatively be transferred from the media card using File Transfer Protocol (section 7.3.2). The media card MUST be reformatted whenever files are deleted from the card to prevent the DataTap-10 from having to search hard for free clusters during ASCB recording. Data loss can otherwise occur. Note that the DataTap-10 FTP server can be used to reformat the card (section 7.3.2) and MUST be used when recording multiple data streams. FlightLine automates formatting the card using the DataTap-10 and can optionally retrieve and replace all configuration (*.cfg) files around the formatting process.

The DataTap-10 supports both FAT16 and FAT32 file systems on the media card. Because the file size limit is 4GB-1 for both of these file systems, recording files are limited to 2GB. Once this limit is reached, the current file segment is closed and the next opened with the same base filename but with an incrementing extension. The first file segment always starts with a .01 extension. The base filename for both raw and decoded ASCB recording can be specified in the system configuration file, while for ARINC-429 it is always "A429" and for ARINC-717 it is always "A717". Note that when recording starts and a file already exists on the card with the same name, a "_2" will be appended to the base filename. If a file already exists with that name as well, the number will be incremented until a unique filename is found. This does not apply to Circular Recording Mode (see section 6.8.6.1) as the filenames are already forced to be unique. Enabling the "TISNATIVE" mode for the recording format in the system configuration file allows a timestamp based on the current system time to be automatically appended to the base filename. This applies to all data stream types.

6.8.2. Supported Cards

While many microSD cards will work fine, raw ASCB recording and decoded ASCB data recording of more than a few thousand parameters (assuming all parameters are recorded each frame) can present some data rate problems with slower cards. The following microSD cards are recommended and have been heavily tested by ICS:

- Samsung 64GB microSDXC UHS-I Card PRO or PRO+

Similarly, some CompactFlash cards can cause issues when performing decoded ASCB data recording of several thousand parameters due to the buffering required when the card becomes busy. Initialization times can vary greatly between brands and models of cards, with some taking up to 31 seconds as allowed by the CompactFlash specification. The following CompactFlash cards are recommended and have been heavily tested by ICS:

- Sandisk 128GB Extreme Pro Compact Flash Card SDCFXP-128G

  - These cards support hot-insertion and have a very fast initialization time. Certain versions of the card, however, may have a longer initialization time and may not support reliable hot-insertion. All versions record reliably, however.

6.8.3. Performance Degradation

The performance of NAND Flash-based devices such as microSD and Compact Flash degrades over time due to increased write and erase times as well as correctable bit errors on reads. As such, after several thousand hours of recording the devices may need to be replaced to prevent data loss due to the limited buffering on the DataTap-10. The number of hours also depends on the RAM size of the DataTap-10. Units with an 8MB RAM rather than a 4MB RAM will be able to handle much more performance degradation of the cards due to the increased buffering.

To aid in the monitoring of the card performance, several fields have been added to the Status Output datagrams. These fields include the current size of the recording buffer, which is allocated dynamically to use up all available RAM, the current occupancy of the buffer during recording, and the maximum occupancy of the buffer during the current recording session. These fields can be used to determine how close to full the buffer is getting during recording and how often. If multiple data streams are being recorded, these fields will alternate between each of the streams.

Note that performing a full format of a microSD card using the DataTap-10 FTP server rather than a PC has a significant positive impact on the performance and wear-leveling of the card. The DataTap-10 issues erase commands to the entire media during a full format, which frees the blocks into an allocation pool for wear-leveling, whereas a PC typically only has the ability to write all zeros or ones to the card. The maximum occupancy of the recording buffer during recording can go down significantly after performing a full format. Note that formatting a microSD card using the DataTap-10 is mandatory when recording multiple data streams.

6.8.4. Formatting the Media

Any new microSD or Compact Flash cards must contain a FAT16 or FAT32 file system in order to be used for recording by the DataTap-10. Larger microSD cards are delivered with the exFAT file system and must be replaced with FAT32. In recent versions of the DataTap-10 software, the FAT32 formatting can be performed by the FTP server, regardless of the existing file system. Note that booting a DataTap-10 with a microSD card containing the exFAT file system will result in a default configuration being used since the system configuration file cannot be read. As such, it will only respond to IP address 192.168.1.1 in this case. Note that microSD cards can be removed and inserted without having to cycle power on the DataTap-10, however, files such as the system configuration file and the TIS configuration files will not be recognized until the next power cycle.

Issuing the command "quote site format" using an FTP client connected to the DataTap-10 will perform a quick FAT32 format of the card. This command does not actually erase data from the card, but creates a FAT32 partition along with the FAT table and root directory, making the card empty.

The command "quote site full format" has recently been added to the FTP server to allow a full format of the card by issuing erase commands to virtually the entire card followed by creation of the FAT32 file system. This process only adds 10 to 30 seconds to a new Samsung PRO 64GB microSD card and is the recommended command to use when reformatting. Note that most PC applications indicating the ability to perform a full format actually write all zeros or ones to the entire card rather than issuing erase commands. This method is not as effective as using the DataTap-10's full format command because it does not indicate to the card that memory blocks are no longer used, which affects the card's wear-leveling algorithm. This command is only available when recording to microSD and not Compact Flash.

A new command "quote site fill" has also been added to the FTP server. This allows filling of the entire media with empty 2GB files for use in Circular Recording Mode (section 6.8.6.1). The files have the name "EMPTY_0" with an incrementing file extension for each file segment starting at .01. Note that this command MUST be run at least once after reformatting when in Circular Recording Mode to prevent recording errors. At power-up, the DataTap-10 will automatically run the command if in Circular Recording Mode and no valid recording files exist on the card. The command takes approximately 11 seconds to complete on a new Samsung PRO 64GB microSD card.

Note that FlightLine v12.10 or later automates the formatting procedure using the DataTap-10 by selecting the "Format Media" button on the DataTap-10 Status and Control Form. If the selected recording media type is set to CompactFlash in the system configuration file, FlightLine will issue "quote site format" command using the FTP server. If the selected type is microSD, it will issue the "quote site full format" command. For microSD, FlightLine also optionally automates retrieval of all configuration files (*.cfg) prior to the formatting process, and then restoring of the files once the formatting process is complete.

6.8.5. Recording Types

6.8.5.1.Raw ASCB-D or eASCB Recording

Raw ASCB-D or eASCB data can be recorded to the selected media type (microSD or optionally Compact Flash) in a binary format and be played back later using the DataTap-10 or FlightLine. When the raw record/send format is configured as TISNATIVE, the first recorded file segment starts with 4KB of zeros followed by 4MB of the ESCAPE registry currently programmed into the DataTap-10. The recording then consists of back-to-back record_block_t (section 9.3.1) structures, each 0x6200 bytes in length. Just before the forced file size limit of 2GB is reached, a new file segment is opened. Subsequent file segments contain only the record_block_t structures and no file header preceding them. Note that a single record_block_t structure will never span two files.

When the raw record/send format is configured as DIANE62, the recording consists only of back-to-back DIANE_RAW_ASCB structures (section 9.4). This gives each recording block a total length of 0x6222 bytes. Similar to the TISNATIVE format, a new file segment is opened just before the 2GB limit is reached. Note that a single DIANE_RAW_ASCB structure will never span two files.

6.8.5.2.Decoded ASCB-D or eASCB Recording

Decoded ASCB-D or eASCB data can be recorded to the selected media type in a binary format. When the decoded record/send format is configured as TISNATIIVE, the decoded ASCB recording consists of back-to-back DATAGRAM structures (section 9.1.1) written to the recording file along with a 16-byte header prepended to each datagram. If “UDP_DEC_LEGACY_MODE” is set to “FALSE” in the system configuration file, the length of each datagram varies according to the number of configured channels associated with that datagram. The datagram will only be as large as necessary to encompass the highest configured channel number for that datagram. If “UDP_DEC_LEGACY_MODE” is set to “TRUE”, then the length of each datagram will be fixed to its maximum size supporting 256 channels. Because the periodicity of each datagram is programmable through the system configuration file, all datagrams may not be written to the recording file each ASCB frame.

The 16-byte recording header prepended to each datagram when in the TISNATIVE format consists of the following:

```
typedef struct
{
      uint32 datagramNum;
      uint32 numChannels;
      uint32 IRIG43;
      uint32 IRIG21;

} dec_rec_hdr_t;
```

When the format is configured as DIANE62, the decoded ASCB recording consists of back-to-back DIANE_CODE_62_DATAGRAM structures (section 9.2.1). Because the ASCB Data portion can end on any byte boundary depending on the number of configured channels, the DIANE trailer (0xDEAD) will be always be forced to a 16-bit boundary.

6.8.5.3. Raw ASCB-A/B/C Recording

Raw ASCB-A/B/C data from both busses can be recorded to the selected media type as back-to-back datagrams (section 9.11) prepended with a 16-byte header, as shown below. Similar to ASCB-D, a frame is added to the recording buffer at the start of a new frame. Unlike ASCB-D, however, no data will be recorded if there is no valid activity present on the bus. The data from the 2 busses is interleaved and may switch at any point between which one is written first. Consequently, the bus number should be processed as well as the frame number which is present in the data.

```
typedef struct
{
   uint32 status;
   uint32 timestamp;
   uint32 count;
   uint32 size;

} ascbc_rec_hdr_t;
```

Where,

status - 32-bit status field, which has the following bit definitions:
- [31:28] - Value of the discrete inputs 4:1, which can be used for event-marking
- [27:1] - Reserved for Future Use
- [0] - Bus on which this frame was received

timestamp - Current value of the 32-bit system counter, which increments every 20 ns.
count - Recorded frame count starting at 1 and incrementing by 1 for each frame.
size - Size of the frame not including header. Always set to 2048.

6.8.5.4.ARINC-429 Recording

ARINC-429 data can be recorded to the selected media type in a binary format which supports recording of all enabled channels. In DataTap-10 software dated prior to Feb 4, 2016. In software dated prior, ARINC-429 data is recorded into a CSV format and only one channel is supported. As with the ASCB recording, ARINC-429 recording can be started and configured via Telnet or UDP command packets. Note that a tool has been provided in FlightLine v12.10 and later to convert a binary recording file into a CSV file.

The binary recording has the following format for each received label:

```
typedef struct
{
   uint32 sec;
   uint32 ch_nsec;
   uint32 data;

} a429_rec_entry_t;
```

Where,

sec Either seconds since 1970 if the selected time source is IEEE1588v2 and is synchronized, or seconds since the beginning of the current year if time source is IRIG and is synchronized. If unsynchronized, it represents the seconds since power-on. When converting to a CSV file, FlightLine uses this field to generate an IRIG time to write into the file. It does not matter whether its seconds since 1970 or since the beginning of the year for this conversion.

ch_nsec The upper 2 bits are the channel the data was received on. The remaining 30 bits are the nanoseconds of the current second.
data The received ARINC-429 data.

6.8.5.5.ARINC-717 Recording

ARINC-717 data is recorded to the selected media type in a binary format. Frames are recorded as they are received, so nothing will be recorded if there is no activity on the bus. The length of each recorded frame is the word rate times 2 plus a 16-byte recording header prepended in the following format:

```
typedef struct
{
   uint32 count;
   uint32 rate;
   uint32 IRIG43;
   uint32 IRIG21;

} a717_rec_hdr_t;
```

Where,

count The frame count starting with 1.
rate The configured rate of the bus in words.
IRIG43 IRIG4 in the upper 16 bits and IRIG3 in the lower 16 bits.
IRIG21 IRIG2 in the upper 16 bits and IRIG1 in the lower 16 bits.

Decoding of the IRIG fields is as follows:

Field	[15:12]	[11:8]	[7:4]	[3:0]
IRIG1	10^0 mS	10^2 uS	10^1 uS	10^0 uS
IRIG2	10^1 Seconds	10^0 Seconds	10^2 mS	10^1 mS
IRIG3	10^1 Hours	10^0 Hours	10^1 Minutes	10^0 Minutes
IRIG4	0	10^2 Days	10^1 Days	10^0 Days

6.8.6. Auxliliary Recording Modes

6.8.6.1.Circular Recording Mode

Circular Recording is a recording mode in which the DataTap-10 overwrites existing recording files rather than creating new ones, preventing the media card from ever filling up. When in Circular Recording Mode, upon the start of recording or opening of a new file segment during recording, the DataTap-10 first verifies at least 2GB of free memory is remaining on the media card. If 2GB or greater is available, the new file is opened using the ***_m.n*** nomenclature described below. If less than 2GB is available, the DataTap-10 searches for the oldest recording file on the card based solely on the existing filenames. Once the oldest file has been found, the file is renamed and then overwritten. Overwriting eliminates the need to delete a file, which can be time-consuming and lead to data loss.

When Circular Recording Mode is enabled, all recording filenames are created with a suffix in the form of **_m.n**, where:

> **m** is the session number [999 >= m >= 1], and
> **n** is the number of the file created during session **m** [2^32 > n >= 1]

The session number is incremented whenever a new recording session is started. The DataTap-10 first parses the entire file directory on the media card for the highest existing session number based on the existing filenames in the directory. It then increments the session number **m**, creates the new filename with a suffix of **_m.n** (where n is 01), and begins recording.

Circular Recording Mode requires that the card be filled with empty 2GB files prior to recording. This should be done immediately after reformatting the card and transferring on the required configuration files. The command "quote site fill" exists in the DataTap-10 FTP server to fill the entire card with empty 2GB files. A fill will be executed automatically on power-up if Circular Recording Mode is enabled and no valid **_m.n**.files exist on the card.

Circular Recording Mode is enabled by setting the option "ASCB_CIRC_REC_MODE" to "TRUE" in the DataTap-10 system configuration file (system.cfg). The DataTap-10 uses the filenames to determine the oldest file on the card. Only recording files in the form of **_m.n** are considered in the search. Once the oldest file is found, it is renamed and its file size set to 0. Instead of searching for free memory once a file system cluster has been filled up, the DataTap-10 uses the existing cluster chain from the file being overwritten. Once the size of the new file reaches near 2GB, the file is closed and the algorithm to find the oldest file is rerun for the next file segment.

The algorithm used to determine the oldest file on the card consists of first searching the entire file directory on the media card for the filename with the lowest session number (**m**). Once that is found, the DataTap-10 then uses the file with the lowest file extension (**n**) within that session as the oldest file. Note that a session number of 0 is valid as the lowest session number, which is generated when the card is filled with empty 2GB files. Any files not containing a valid session number (**m**) and file extension (**n**) are skipped in the search.

If no files exist on the card in the form of **_m.n** yet less than 2GB is available on the card, a recording error is generated. This stops recording and causes the HEALTH_STATUS discrete output to indicate unhealthy due to the error, if the discrete outputs are configured as such. The RECORD_STATUS status discrete output also indicates recording has stopped.

Note that the term "reaches near 2GB" is used because the recording files must end on a recording frame boundary, so they are always be stopped short of crossing the 2GB boundary. As a result, the DataTap-10 must assume that the cluster chain of the file being overwritten is at least 2GB, which it always will be if the card was first filled with empty 2GB files. If the card was not filled with empty 2GB files after reformatting, it may result in the end of the cluster chain being reached prior to 2GB, in which case a recording "End of File" error is generated. This stops recording and causes the HEALTH_STATUS and RECORD_STATUS discrete outputs to deassert, if configured.

The session number suffix is added to the filename regardless of the setting for the existing option "REC_FILENAME_FORMAT", which can be set to "STANDARD" or "TISNATIVE". Note that to maintain compatibility with existing DataTap-10 versions, the recording session suffix (**_m**) is NOT added when Circular Recording mode is disabled.

Example 1:

The following example shows 5 recording sessions when circular recording mode is enabled and "REC_FILENAME_FORMAT" is set to "STANDARD":

ASCB_CIRC_REC_MODE TRUE
REC_FILENAME_FORMAT STANDARD
FULL_REC YES
FULL_REC_FILE Raw

	Nb of files created during the session					
Format	5	EMPTY_0.01	EMPTY_0.02	EMPTY_0.03	EMPTY_0.04	EMPTY_0.05
Session 1	2	Raw_1.01	Raw_1.02	EMPTY_0.03	EMPTY_0.04	EMPTY_0.05
Session 2	2	Raw_1.01	Raw_1.02	Raw_2.01	Raw_2.02	EMPTY_0.05
Session 3	4	Raw_3.02	Raw_3.03	Raw_3.04	Raw_2.02	Raw_3.01
Session 4	7	Raw_4.03	Raw_4.04	Raw_4.05	Raw_4.06	Raw_4.07
Session 5	1	Raw_5.01	Raw_4.04	Raw_4.05	Raw_4.06	Raw_4.07

Example 2:

The following example shows 5 recording sessions when in circular recording mode and "REC_FILENAME_FORMAT" is set to "TISNATIVE":

ASCB_CIRC_REC_MODE TRUE
REC_FILENAME_FORMAT TISNATIVE
FULL_REC YES
FULL_REC_FILE Raw

	Nb of files created during the session					
Format	5	EMPTY_0.01	EMPTY_0.02	EMPTY_0.03	EMPTY_0.04	EMPTY_0.05
Session 1	2	Raw_<time>_1.01	Raw_<time>_1.02	EMPTY_0.03	EMPTY_0.04	EMPTY_0.05
Session 2	2	Raw_<time>_1.01	Raw_<time>_1.02	Raw_<time>_2.01	Raw_<time>_2.02	EMPTY_0.05
Session 3	4	Raw_<time>_3.02	Raw_<time>_3.03	Raw_<time>_3.04	Raw_<time>_2.02	Raw_<time>_3.01
Session 4	7	Raw_<time>_4.03	Raw_<time>_4.04	Raw_<time>_4.05	Raw_<time>_4.06	Raw_<time>_4.07
Session 5	1	Raw_<time>_5.01	Raw_<time>_4.04	Raw_<time>_4.05	Raw_<time>_4.06	Raw_<time>_4.07

6.8.6.2. Record Append Mode

Record Append is a mode in which a file previously being recorded into within the same power cycle is reopened and appended to if recording is restarted. This option can be enabled in the system configuration file by setting "ASCB_REC_APPEND" to "TRUE". When set to "TRUE" and a recording is restarted during the same power cycle, the DataTap-10 first verifies the size of the last recording file. If it is less than 2GB and has room for several ASCB frames worth of data, the DataTap-10 then searches for the end of the file and begins recording from that point. If it is already at or near 2GB, the DataTap-10 instead opens a new file for recording with the same base filename but with an incremented file extension. When "ASCB_REC_APPEND" is set to "FALSE", a new recording file is opened each time recording is started, regardless of whether a power cycle occurred or not.

Note that when recording is started the first time after a power cycle, a new recording file is opened regardless of the setting of "ASCB_REC_APPEND". If the media card is removed or the ASCB recording type is changed via Telnet or UDP Command Input datagrams, the previous file is no longer considered valid for appending.

6.9. ARINC-429 Receive

The DataTap-10 has four ARINC-429 receive channels. The channels can be enabled and configured through Telnet or UDP Command packets. The configuration process is automated through the FlightLine software, which also allows viewing of individual received labels. Received labels can also be monitored over Telnet.

At the rate specified in the UDP Command packet when the channel was enabled, the DataTap-10 software will package all received labels into a single UDP packet (section 9.9) along with each label's calculated receive rate in microseconds. The recommended and default rate is 50 microseconds.

6.10. ARINC-429 Auto-Transmit Module (ATM)

The DataTap-10 ARINC-429 Auto-Transmit Module (ATM) allows ARINC labels to be transmitted periodically on the bus with no software intervention. The only action required by the DataTap-10 software is to set up and update, if necessary, the transmit descriptor ring and to configure the ATM registers. The DataTap-10 ATM and the transmit descriptor ring can be configured and updated via UDP command packets. This process is automated in the FlightLine software.

If the ATM module is enabled on a given channel, hardware will parse the entire transmit descriptor ring for that channel at the specified loop rate. A transmit descriptor ring exists in DataTap-10 system memory for each transmit channel. The ring looks like the following, with each descriptor containing two 32-bit fields:

Index	31 0	31 0
0	Rate Divider	Tx Data
1	Rate Divider	Tx Data
2	Rate Divider	Tx Data
…	Rate Divider	Tx Data
254	Rate Divider	Tx Data

Hardware will begin parsing the transmit descriptor starting at index 0 and ending at the number of configured descriptors. A value of 0 for the number of descriptors will cause no parsing of the ring by hardware.

At the start of each polling loop, hardware will first increment an internal frame counter. As it is parsing the ring, if it encounters a rate divider with a value of 0, the transmit descriptor will be skipped and hardware will move to the next descriptor. A value of 1 indicates the data in that transmit descriptor will be transmitted each and every frame. For values other than 0 or 1, hardware will use modular math to determine if the data in that transmit descriptor should be transmitted or not.

Note that if the loop rate is specified at a time faster than the required time to send out the specified number of labels onto the ARINC bus, the loop rate will automatically be slowed down to account for this. In this case, hardware will immediately begin parsing index 0 of the transmit descriptor ring as soon as it completes parsing of the last index in the ring.

6.11. ARINC-717 Decoder

The DataTap-10 has the ability to decode ARINC-717 data with supported word rates of 64, 128, 256, 512, 1024, 2048, 4096, and 8192 words per second. The decoder can be enabled and configured via the system configuration file (section 7.1.8) or UDP command packets (section 9.8). As of FPGA version 0x12291501 or later, the ARINC-717 data can be received on any discrete input or ARINC-429 channel, and is specified in the system configuration file. For earlier FPGA versions, only discrete input 1 of connector P4 (see Table 6) can be used to receive the data. Note that when using a discrete input, the maximum supported word rate is 512 wps due to the optical isolation used in the DataTap-10 hardware. When using an ARINC-429 channel, all of the word rates are supported.

When first enabled, the ARINC-717 hardware decoder uses 8 separate match units to search the incoming bit stream for one of the four synchronization words. Once one has been found, the configured number of 12-bit words is counted. If the following word matches the next expected synchronization word, the decoder indicates to software it is synchronized and continuously fills a FIFO with the received data for software processing. At the start of each and every frame, the decoder verifies the first word of the frame matches the next expected synchronization word. If there is a mismatch, the decoder indicates to software that it is no longer synchronized and restarts the synchronization process.

The received ARINC-717 data can be viewed through Telnet (see section 7.4). A UDP datagram will also be sent out containing the data from the entire frame (see section 9.9.1) once per second when synchronized. These datagrams are used by FlightLine to decode and display the data, or they can be received by custom software to process the data. Contact ICS if you wish to receive any example code.

6.12. Status LED

The single status LED on the front of the DataTap-10 is used to signal status information, including initialization status, connection status, and/or recording status.

The conditions shown in Table 8 apply to initialization. Any condition involving red indicates an initialization error and will persist until the next power cycle. Upon a successful initialization, control of the LED is determined by the setting of "STS_LED_USE" in the system configuration file.

Status LED Indication	Meaning
Blinking green at a 1Hz rate	The DataTap-10 is initializing. This can take up to 30 seconds depending on the number of parameters in the TIS configuration files and whether the CompactFlash interface is being used for recording.
Blinking red once every 2 seconds	The microSD card is missing. Default values are used for all parameters normally found in the system configuration file.
Blinking red twice every 2 seconds	The system configuration file (system.cfg) is missing from the microSD card. Default values are used for all parameters normally found in the file.
Blinking red 3 times every 2 seconds	An invalid parameter or parameter value was found in the system configuration file. Default values are used for any invalid parameter or parameter values. The DataTap-10 will also write an error log to the microSD card giving information about the error. See section 7.1.8 for further details regarding the error log.

Table 8 Status LED Indications for Initialization

The conditions shown in Table 9 are valid only when "REC_STS_LED" is set to "FL_CONN" in the system configuration file, meaning the status LED is used for indicating FlightLine connection status.

Status LED Indication	Meaning
Static green	The DataTap-10 is properly connected to FlightLine or it is receiving UDP command packets from another source.
Static red	The DataTap-10 is not connected to FlightLine nor receiving UDP command packets from another source.

Table 9 Status LED Indications for FlightLine Connection

The conditions shown in Table 10 are valid only when "STS_LED_USE" is set to "REC_STS" in the system configuration file, meaning the status LED is used for indicating recording status.

Status LED Indication	Meaning
Alternating green and red at a 1Hz rate	Recording is in progress and the amount of recording time remaining on the selected media card is less than the threshold value specified with "SD_MEMORY_RECORD_WARNING" in the system configuration file.
Static green	Recording is in progress.
Blinking red at a 4Hz rate	Recording has been stopped due to an error. It will also be in this state whenever the media card is removed from the unit.
Static red	Recording is stopped.
Static orange	An automatic recording is pending. Depending on the configuration, this means it is waiting on the time source to lock, the power-up recording delay to expire, or ASCB traffic to be present.

Table 10 Status LED Indications for ASCB Recording

6.13. Activity LEDs

The 8 activity LEDs on the front of the DataTap-10 are used to indicate reception and transmission activity as well as errors detected by the bootloader. The possible states and meaning of the activity LEDs are shown in Table 11. Note that the states are mutually exclusive such that only one state will ever be active at a time.

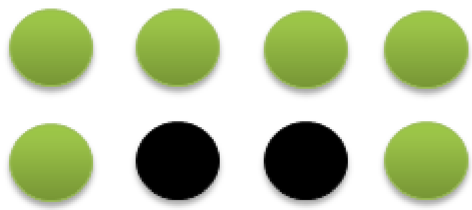
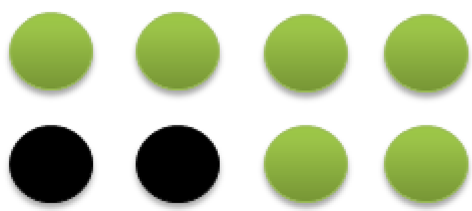
Activity LED Indication	Meaning
Static pattern indicating the letter C rotated 90 degrees clockwise. 	One or more of the two software executables is corrupted. This can occur if the software update files were corrupted prior to or during the transfer to the microSD card, or if power is removed during a software update and the card is then removed prior to the next power-up. To recover, simply remove the microSD card, copy the software files onto the card with a PC, and then put the card back into the DataTap-10 and cycle power.
Static pattern indicating the letter P rotated 90 degrees clockwise. 	An error occurred during the programming of software or registry files from the microSD card to internal Flash. This can indicate a hardware error, but most likely indicates the registry being programmed into Flash is either larger than 4MB or does not contain a valid header. To determine the error, remove the microSD card and cycle power. If the DataTap-10 boots up as normal, connect via Telnet and look at page 4 to view any detected registry errors. If the DataTap-10 does not boot up, retry the software update. If this still fails, contact ICS as there is likely a hardware error.
Turning on one-by-one once per second in a rectangular pattern.	The bootloader is busy programming software or registry files from the microSD card to internal Flash memory.
Turning on one-by-one using only the top row sweeping left to right during one second and then back to left in the following second.	An ASCB playback over UDP (Rx RAW) is in progress.
Turning on two-by-two using the top and bottom rows sweeping left to right during one second and then back to left in the following second.	An ASCB playback over ASCB (Tx RAW) playback is in progress
Individually flashing on and off at a 5Hz rate.	Indication of ASCB reception and/or transmission activity.

Table 11 Activity LED Indications

6.14. Discrete Inputs

The 4 discrete inputs of the DataTap-10 can be used as general purpose inputs whose values can be read over Telnet or in the Status Output datagrams sent out by the DataTap-10. Any one of the discrete inputs can alternatively be configured to control recording to the microSD or optional Compact Flash card. The discrete inputs can accept voltages up to 60V and can be triggered high by voltages as low as 3.3V. Only positive and negative wires need to be hooked up for a given discrete input, which are fed into an opto-coupler. There are no other grounds or power pins dedicated to the discrete inputs that need to be hooked up. The DataTap-10 performs de-bouncing of the discrete inputs by only recognizing a state change if the value has been stable for 100 milliseconds.

The following options are available in the system configuration file for discrete input configuration (see section 13):

DIN1_CONFIG	NONE | REC_CTL_LVL | REC_CTL_EDGE
DIN2_CONFIG	NONE | REC_CTL_LVL | REC_CTL_EDGE
DIN3_CONFIG	NONE | REC_CTL_LVL | REC_CTL_EDGE
DIN4_CONFIG	NONE | REC_CTL_LVL | REC_CTL_EDGE

Where "NONE" indicates the discrete input has no specific function, "REC_CTL_LVL" indicates the discrete input is to be used for level-sensitive recording control, and "REC_CTL_EDGE" indicates the discrete input is to be used for edge-triggered recording control. Note that using a discrete input for recording control is referred to as MANUAL RECORDING control. If no discrete inputs are configured for recording control, they will not have any effect on recording regardless of their state or transitions.

When a discrete input is configured for level-sensitive control, a high level on that input triggers recording to be started if a recording is not currently in progress. A low level indicates to stop recording. If AUTOMATIC RECORDING is enabled by setting AUTO_REC_ENABLE to TRUE and any of RAW_REC_EN, DEC_REC_EN, A429_REC_EN, or A717_REC_EN to TRUE, any discrete input configured for level-sensitive recording control will be ignored until automatic recording has started. If this discrete input is low when automatic recording attempts to start, however, recording will not start until a high level is detected on the discrete input. Note that recording control over Telnet, UDP, or FlightLine is not possible when using level-sensitive discrete input recording control since the recording will immediately be stopped or started based on the state of the discrete input.

When a discrete input is configured for edge-triggered control, a low-to-high transition on that input starts recording and a high-to-low transition stops recording. Edge-triggered control over-rides any automatic recording control, meaning that if an automatic recording is pending, recording can still be stopped or started using the discrete input. Using edge-triggered control allows control using Telnet, UDP, or FlightLine to be used as well.

Note that only one discrete input can be configured for recording control. If multiple are assigned, similar to any error in the system configuration file, the status LED will flash red and the generated ERRLOG.TXT file will list the option(s) causing the error (see section 7.1.8). If the error is ignored, only the first discrete input configured for recording control encountered in the system configuration will be used. The configuration of the discrete inputs can be viewed over Telnet.

Upon connecting to the DataTap-10 FTP server, any discrete inputs configured for level-sensitive recording control will no longer be able to control recording until the next power cycle. The existing FTP command "quote site reset" can always be used to cause a virtual power cycle.

Which data streams to record are initially set using RAW_REC_EN, DEC_REC_EN, A429_REC_EN, and A717_REC_EN in the system configuration file on power-up. These record enables can be changed over Telnet when recording is not active. When the discrete input indicates to record, whichever streams are enabled for recording will be recorded. Note that if AUTO_REC_ENABLE is set to TRUE, which is the default to maintain backwards compatibility, automatic recording will begin at power-up based on the record enables.

The truth table below describes the result of AUTOMATIC and level-sensitive MANUAL recording control:

AUTO_REC_ENABLE set to TRUE (AUTO)	RAW_REC_EN or DEC_REC_EN or A429_REC_EN or A717_REC_EN set to TRUE	One DINx_CONFIG set to REC_CTL_LVL (Level-Sensitive MANUAL)	Result
FALSE	X	FALSE	Recording OFF. Can only be enabled via Telnet/UDP/FlightLine.
X	FALSE	TRUE	Recording OFF due to no recording streams selected. Can only be enabled via Telnet/UDP/FlightLine, but then MANUAL Recording Control will take over once a stream is selected.
FALSE	TRUE	TRUE	MANUAL Recording Control. Recording of the selected streams will start based on the discrete input state.
TRUE	FALSE	FALSE	Recording OFF. Can only be enabled via Telnet/UDP/FlightLine.
TRUE	TRUE	FALSE	AUTOMATIC Recording Control. Recording will start automatically at boot-up after all conditions, if enabled, are met (time source locked, power-on delay expired, and/or ASCB traffic present).
TRUE	TRUE	TRUE	Recording will start automatically at boot-up after all conditions, if enabled, are met. Control is then handed over to level-sensitive MANUAL Control. If DINx configured for level-sensitive recording control is LOW once the conditions are met, then recording will not begin until DINx is HIGH.

7. DataTap-10 Adapter Configuration

7.1. Configuration Files

The DataTap-10 uses up to eight configuration files depending on the mode of operation. If the DataTap-10 is an RX only device and is not decoding parameters on-board the DataTap (i.e. you are only using FlightLine to view parameters), then only two files are actually required: ”nic.reg” and “system.cfg”.

Of the eight configuration files, four of them exist on the unit’s microSD card and four are programmed into the unit’s internal Flash memory. Depending on the DataTap-10’s mode of operation, conversion of ASCB-D or Enhanced ASCB data may not take place if any of the files are missing.

The four files which exist on the unit’s microSD card are named “system.cfg”, “tis.cfg”, “tis_tx0.cfg”, and “tis_tx1.cfg”. They must reside in the base directory of the microSD card. These files can be transferred to the microSD card either by removing the card and copying the files onto it directly with a PC, or through the use of File Transfer Protocol (FTP) over the DataTap-10 Ethernet Interface. Power will need to be cycled after the files have been transferred to the microSD card in order for the new files to be recognized.

The four files which are programmed into the unit’s internal Flash memory are named ”nic.reg”, “escape.reg”, “esc_tx0.reg”, and “esc_tx1.reg”. These files are programmed via the microSD card - either by copying the files onto the card directly with a PC or through the use of File Transfer Protocol (FTP). The DataTap-10 will check if these files exist in the base directory of the microSD card on boot-up. If any exist, the file(s) will automatically be programmed into the internal Flash memory and then erased from the microSD card. The LEDs on the front panel will flash one-by-one in a rectangular pattern while the programming operation is in progress. The unit will begin responding to ping requests once the programming is complete. No cycling of power is required after the programming is complete; however, power will need to be cycled after the files have been transferred to the microSD card in order to initiate the programming sequence. This programming process only needs to occur once unless different files wish to be used.

Note that any of the above files can be transferred to the DataTap-10's microSD card using FlightLine software version 12.1 or later. FlightLine will also automate resetting of the unit to initiate the programming procedure. Starting with DataTap-10 FPGA version 09031401 or later, all registries can be in Little Endian or Big Endian format. Registries must be in Little Endian format for FPGA versions prior to this, meaning only the files with the "le_reg" file extension can be used from the ESCAPE tool.

Only the "nic.reg" file is required when using the DataTap-10 along with the FlightLine software to decode parameters and/or transmit parameters in NIC-Only Mode. Both the "nic.reg" and "escape.reg" are required when using the DataTap-10 in a standalone mode or viewing decoded parameters over Telnet. Neither file is required if using the DataTap-10 in a pass-thru mode of operation, which is indicated in the system.cfg file and allows the ASCB-D data to be passed through the DataTap-10 unaltered to the Ethernet port. The "esc_tx0.reg" and "esc_tx1.reg" file are only required when performing transmission onto the ASCB bus in PDD and NIC Mode (section 6.4.2).

Up to 8192 total parameters and/or channels can exist in the "tis.cfg", "tis_tx0.cfg", and "tis_tx1.cfg" files. The "tis_tx0.cfg" and "tis_tx1.cfg" files are only used when the DataTap-10 is configured for PDD and NIC Mode ASCB Transmission (i.e. HW_TX_PDD_ENABLE set to TRUE). At initialization time, the DataTap-10 first reads the produce TIS configuration file for bus 0 ("tis_tx0.cfg"), then the produce TIS configuration file for bus 1 ("tis_tx1.cfg"), and finally the consume TIS configuration file ("tis.cfg"). Once the 8192 parameter/channel limit is reached, no more parameters will be added for produce or consume. It is recommended that the channel numbering in each TIS configuration file start with channel 0 because the highest channel number found in each file plus one indicates the number of channels used by that file. For example, if the bus 0 produce file has only one entry and is assigned channel 8191, the DataTap-10 will interpret this as using 8192 channels and no entries from the bus 1 produce file or the consume file will be used. The number of entries found in each TIS configuration file as well as the channels allocated can be found on the Telnet screens.

It is possible that less than 8192 total ASCB parameters (encode + decode) when recording is active. This is because the parameter encode and decode engines must complete within one ASCB frame (12.5 milliseconds), or data corruption on transmission and/or inadvertent group status errors can occur. As the memory bandwidth requirement is increased due to recording, completing within 12.5 milliseconds may not be possible when encoding or decoding a large number of parameters. As such, multiple fields have been added to the Telnet screens and Status UDP Output Datagrams (see section 9.7) to monitor this condition. These fields indicate the load of the engines and the number of times each engine has exceeded the allotted 12.5 millisecond frame time. They are all available for display in FlightLine v12.10 or later. The number of parameters in ANY of the TIS configurations should be reduced until the counts are no longer incrementing and the peak load percentages are below 100%. For example, if the frame exceeded count is incrementing for the decoding engine, the number of parameters in ANY of the TIS configuration files can be reduced to help the decoding engine complete within its allotted time. Note that if recording ARINC-429 or ARINC-717 data and UDP output is not required, the UDP output can be turned off for these data streams in the system configuration file to save memory bandwidth and potentially increase the number of parameters supported, if less than 8192.

The configuration files are described in the following sections.

7.1.1. TIS Consume Configuration File (tis.cfg)

This binary file is created by the FlightLine software and contains a list of parameter entries indicating which ASCB-D parameters are to be decoded by the DataTap-10. The specified parameters and their decoded values are available for display over Telnet and will also be sent out in UDP datagrams (section 9.1). The file must be named "tis.cfg" and must exist on the microSD card at power-up in order for the UDP datagrams to be sent out. Note that this file does not affect which parameters are consumed by the FlightLine software. The FlightLine software has the ability to decode any parameter on the bus as long as a valid NIC registry is programmed into the DataTap-10.

7.1.2. TIS Bus 0 Produce Configuration File (tis_tx0.cfg)

This binary file is created by the FlightLine software and contains a list of parameter entries to be produced onto ASCB bus 0 by the DataTap-10. Each parameter listed has the ability to have its value changed by sending in UDP datagrams to the DataTap-10. The specified parameters and their values are available for display over Telnet. The file must be named "tis_tx0.cfg".
This file is only required if the transmission option was purchased and transmission is desired.

7.1.3. TIS Bus 1 Produce Configuration File (tis_tx1.cfg)

This binary file is created by the FlightLine software and contains a list of parameter entries to be produced onto ASCB bus 1 by the DataTap-10. Each parameter listed has the ability to have its value changed by sending in UDP datagrams to the DataTap-10. The specified parameters and their values are available for display over Telnet. The file must be named "tis_tx1.cfg".
This file is only required if the transmission option was purchased and transmission is desired.

7.1.4. Consume ESCAPE Registry File (escape.reg)

This file is used by the DataTap-10 to decode parameters from the ASCB bus. It specifies which parameters are consumed from either ASCB bus. Often a "Consume All" registry will be used here such that all parameters from either bus can be consumed. ESCAPE registries produced by the Honeywell ESCAPE have the name "escape.le_reg" or "escape.be_reg". This file must be renamed to "escape.reg" and programmed into the DataTap-10 via the microSD card at least once. Without this file being programmed, the DataTap-10 will not consume any parameters; however, it still has the ability to send entire NIC frame buffers to the FlightLine software where parameters can be consumed. FlightLine has the ability to generate "Consume All" registries. See the FlightLine User's Manual for instructions on how to generate this registry.

7.1.5. Bus 0 Produce ESCAPE Registry File (esc_tx0.reg)

This file is used by the DataTap-10 to encode parameters onto the ASCB bus. It specifies all parameters produced on a single ASCB bus by a specific MAU module. ESCAPE registries produced by the Honeywell ESCAPE have the name “escape.le_reg” or “escape.be_reg”. This file must be renamed to “esc_tx0.reg” and programmed into the DataTap-10 via the microSD card at least once. Without this file being programmed, the DataTap-10 will not produce any parameters; however, it still has the ability to work in the “NIC transmission” mode and receive entire NIC frame buffers from the FlightLine software and transmit them onto a given ASCB bus. This file is only required if the transmission option was purchased and transmission is desired.

7.1.6. Bus 1 Produce ESCAPE Registry File (esc_tx1.reg)

This file is used by the DataTap-10 to encode parameters onto the ASCB bus. It specifies all parameters produced on a single ASCB bus by a specific MAU module. ESCAPE registries produced by the Honeywell ESCAPE have the name “escape.le_reg” or “escape.be_reg. This file must be renamed to “esc_tx1.reg” and programmed into the DataTap-10 via the microSD card at least once. Without this file being programmed, the DataTap-10 will not produce any parameters; however, it still has the ability to work in the “NIC transmission” mode and receive entire NIC frame buffers from the FlightLine software and transmit them onto a given ASCB bus. This file is only required if the transmission option was purchased and transmission is desired.

7.1.7. NIC Registry File (nic.reg)

Like all NIC cards in an ASCB system, the DataTap-10 ASCB-D requires a transmission registry file. The transmission registry files in the PRIMUS EPIC system are the same for every NIC in the system. That is, only one file is produced by the ESCAPE tool named “NIC.le_reg” or “NIC.be_reg”. The file must be renamed to “nic.reg” prior to being transferred to the microSD card for internal Flash memory programming.

7.1.8. System Configuration File (system.cfg)

The system configuration file is a text file which allows system parameters of the DataTap-10 ASCB-D to be assigned away from their default values. A missing “system.cfg” file or missing parameters within the file will result in default values being used. Comments can be added to the system.cfg file by preceeding the comment with the ‘#’ character.

If an unrecognized parameter or parameter value is found while parsing the file, the DataTap-10 will create a file called “ERRLOG.TXT” on the microSD card listing which parameters and/or parameter values were found to be invalid. The Status LED will also report this error by flashing red (see section 6.12) three times every two seconds indefinitely.

An example system configuration file with the latest parameters and a thorough description of each parameter is shown in Section 13. This is the file typically shipped with latest DataTap-10 units, though some parameter values may be changed by ICS prior to shipment based on known customer requirements. Listed below are the available parameters along with their default values, the system configuration file version in which they were added, and a short description. Refer to Section 13 for a detailed description of each parameter.

To simplify the configuration file, any parameters which will never be changed away from their default value can be removed from the file, if desired.

Parameter	Default	Added in Version	Description
BOARD_MAC	00:0a:35:00:01:01	1.1	Sets the MAC address of the DataTap-10.
BOARD_IP	192.168.1.1	1.1	Sets the IP address of the DataTap-10.
NET_MASK	255.255.255.0	1.1	Sets the network mask for the DataTap-10.
HOST_IP_STS	192.168.1.200	1.1	Sets the destination IP address for the Status Output Datagrams sent by the DataTap-10.
HOST_IP_DEC	192.168.1.200	1.1	Sets the destination IP address for the Decoded Parameter Output Datagrams sent by the DataTap-10.
HOST_IP_RAW	192.168.1.200	1.1	Sets the destination IP address for the Raw Frame Buffer Output Dtagrams sent by the DataTap-10.
HOST_IP_A429	192.168.1.200	1.1	Sets the destination IP address for the ARINC-429 Rx Output Datagrams sent by the DataTap-10.
HOST_IP_A717	192.168.1.200	1.1	Sets the destination IP address for the ARINC-717 Rx Output Datagrams sent by the DataTap-10.
UDP_RAW_EN_DEF	TRUE	1.1	Specifies whether raw output is enabled by default.
UDP_DEC_EN_DEF	TRUE	1.1	Specifies whether decoded output is enabled by default.
UDP_STS_EN_DEF	TRUE	1.1	Specifies whether status message output is enabled by default.
UDP_HOST_PORT_STS	51000	1.1	Sets the destination UDP port for the Status Output Datagrams sent by the DataTap-10.
UDP_HOST_PORT_DEC	51020	1.1	Sets the destination UDP port for the Decoded Parameter Output Datagrams sent by the DataTap-10.
UDP_HOST_PORT_RAW	51008	1.1	Sets the destination UDP port for the Raw Frame Buffer Output Datagrams sent by the DataTap-10.
UDP_HOST_PORT_A429	51012	1.1	Sets the destination UDP port for the ARINC-429 Rx Output Datagrams sent by the DataTap-10.
UDP_HOST_PORT_A717	51016	1.1	Sets the destination UDP port for ARINC-717 Rx Output Datagrams sent by the DataTap-10.

Parameter	Default	Added in Version	Description
UDP_LSTN_PORT_CMD	51001	1.1	Sets the UDP port on which to listen for Command Input Datagrams sent to the DataTap-10.
UDP_LSTN_PORT_RAW	51004	1.1	Sets the UDP port on which to listen for Raw Frame Buffer Input Datagrams sent to the DataTap-10.
ESC_REG_CRC_EN	FALSE	1.1	Specifies whether to perform a CRC check on the ESCAPE registries.
ASCB0_RXSW_DEF	PRIMARY	1.1	Specifies the default ASCB bus 0 receive switch setting.
ASCB1_RXSW_DEF	PRIMARY	1.1	Specifies the default ASCB bus 1 receive switch setting.
ASCB0_PASSTHRU_EN	FALSE	1.1	Specifies whether to enable pass-thru mode for ASCB bus 0.
ASCB1_PASSTHRU_EN	FALSE	1.1	Specifies whether to enable pass-thru mode for ASCB bus 1.
ASCB0_NIC_IDS	0x00000000	1.1	Sets the NIC IDs of the DataTap-10 for ASCB bus 0. Used for transmission only.
ASCB1_NIC_IDS	0x00000000	1.1	Sets the NIC IDs of the DataTap-10 for ASCB bus 1. Used for transmission only.
A717_WORD_RATE_DEF	512	1.1	Sets the default word rate of the ARINC-717 bus.
A717_EN_DEF	FALSE	1.1	Specifies whether the ARINC-717 decode module is enabled by default.
IRIG_HAS_DC_OFFSET	AUTO	1.1	Indicates whether the IRIG source has a DC offset.
NUM_UDP_SYNC_MSGS	2	1.1	Specifies the number of frame sync messages to be sent out by the DataTap-10.
PTP_ANNOUNCE_TIMEOUT	3	1.2	Specifies the PTP announce timeout.
PTP_TIMEZONE_OFFSET	0	1.2	Specifies any desired time zone adjustment for the PTP time since it is always expressed in UTC.
DEVICE_NUM	0	1.2	Specifies the DataTap-10 device number. This value is copied into the DIANE headers.
HOST_IP_SYNC	192.168.1.200	1.2	Sets the destination IP address for sync message datagrams sent by the DataTap-10.
UDP_SYNC_EN_DEF	TRUE	1.2	Specifies whether sync message output is enabled by default.

Parameter	Default	Added in Version	Description
UDP_HOST_PORT_SYNC	51009	1.2	Sets the destination UDP port for frame sync datagrams sent by the DataTap-10.
USE_TIMING_SOURCE	IRIGB	1.2	Specifies the timing source for the DataTap-10: IRIG-B or PTPv2 (IEEE1588v2).
OUTPUT_HEALTH_INFO	FALSE	1.2	Specifies whether the discrete outputs are used to indicate health and status of the DataTap-10.
SD_MEMORY_RECORD_WARNING	10	1.2	Specifies the number of minutes left on the media before issuing a warning using the discrete outputs, if enabled.
USE_COMPACT_FLASH	FALSE	1.3	Indicates whether to use the Compact Flash interface rather than microSD for recording.
UDP_LSTN_PORT_SIU0	51100	1.3	Sets the UDP port on which to listen for ASCB bus 0 Parameter Input Datagrams sent to the DataTap-10.
UDP_LSTN_PORT_SIU1	51200	1.3	Sets the UDP port on which to listen for ASCB bus 1 Parameter Input Datagrams sent to the DataTap-10.
USE_STS_LED_FOR_REC_STS	FALSE	1.4	Indicates whether to use the status LED for recording status rather than connection to FlightLine.
HW_TX_PDD_ENABLE	FALSE	1.5	Indicates whether the hardware transmit PDD is enabled.
UDP_DEC_LEGACY_MODE	FALSE	1.5	Indicates whether the Decoded Output datagrams should be in legacy format.
PLAYBACK_MODE	OFF	1.5	Indicates the default playback mode at power-up.
PLAYBACK_FILE	ASCB_RAW.01	1.5	Indicates the file to playback at power-up, if enabled.
PLAYBACK_IRIG	001:00:00:00	1.5	Indicates an IRIG time found in the specified recording file at which to begin playback.
REC_FILENAME_FORMAT	STANDARD	1.6	Indicates the recording filename format.
IRIG_YEAR	2015	1.6	Sets the year for the IRIG time if not sent by the IRIG source.
ASCB_VERSION	ASCB-D	1.7	Indicates the ASCB version for transmit and receive.
RAW_REC_EN	FALSE	1.7	Specifies whether to record ASCB Raw data.
DEC_REC_EN	FALSE	1.7	Specifies whether to record ASCB Decoded data.

Parameter	Default	Added in Version	Description
RAW_REC_FILE	ASCB_RAW	1.7	Specifies the base filename used for raw recording.
DEC_REC_FILE	ASCB_DEC	1.7	Specifies the base filename used for decoded recording.
ASCB_FLOAT_ENCODE_MODE	ROUND	1.7	Indicates whether the encoding for float parameters onto the ASCB bus should use rounding or truncation.
RAW_SEND_REC_FORMAT	TISNATIVE	1.7	Indicates the file format used for sending and recording raw data.
DEC_SEND_REC_FORMAT	TISNATIVE	1.7	Indicates the file format used for sending and recording decoded data.
DEC_SEND_[1:32]_RATE	80	1.7	Specifies the rate at which each Decoded Parameter Output Datagram should be sent out.
TMG_MSG_STRING	ICS DataTap-10	1.9	Specifies the string to be used in frame sync messages transmitted onto the ASCB bus by the DataTap-10.
ASCB_REC_APPEND	FALSE	1.10	Indicates whether record append mode is enabled.
ASCB_CIRC_REC_MODE	FALSE	1.10	Indicates whether circular recording mode is enabled.
ASCB_REC_WAIT_DATA	FALSE	1.10	Indicates whether ASCB data must be present on the bus prior to starting automatic recording, if enabled.
AUTO_REC_DELAY_SEC	0	1.10	Specifies the number of seconds to wait after power-up before starting automatic recording, if enabled.
DIN1_CONFIG	NONE	1.10	Indicates the configuration of Discrete Input 1.
DIN2_CONFIG	NONE	1.10	Indicates the configuration of Discrete Input 2.
DIN3_CONFIG	NONE	1.10	Indicates the configuration of Discrete Input 3.
DIN4_CONFIG	NONE	1.10	Indicates the configuration of Discrete Input 4.
A429_REC_EN	FALSE	1.11	Specifies whether to record ARINC-429 data.
A717_REC_EN	FALSE	1.11	Specifies whether to record ARINC-717 data.
UDP_A429_EN_DEF	TRUE	1.11	Specifies whether ARINC-429 UDP Output is enabled by default.
UDP_A717_EN_DEF	TRUE	1.11	Specifies whether ARINC-717 UDP Output is enabled by default.

Parameter	Default	Added in Version	Description
AUTO_REC_ENABLE	TRUE	1.11	Indicates whether to start recording at power-up, which is referred to as automatic recording. Note that one of RAW_REC_EN, DEC_REC_EN, A429_REC_EN, or A717_REC_EN must also be set to TRUE in order for recording to begin at power-up.
A429_RX1_CONFIG	OFF	1.11	Specifies the configuration of ARINC-429 Channel 1.
A429_RX2_CONFIG	OFF	1.11	Specifies the configuration of ARINC-429 Channel 2.
A429_RX3_CONFIG	OFF	1.11	Specifies the configuration of ARINC-429 Channel 3.
A429_RX4_CONFIG	OFF	1.11	Specifies the configuration of ARINC-429 Channel 4.
A717_INPUT	DIN1	1.11	Specifies the discrete input or ARINC-429 receive channel to use for ARINC-717.
STS_LED_USE	FL_CONN	1.12	Indicates how the status LED is to be used after a successful initialization.
REC_APPEND_MODE	FALSE	1.13	Indicates whether record append mode is enabled.
CIRC_REC_MODE	FALSE	1.13	Indicates whether circular recording mode is enabled.
AUTO_REC_WAIT_ASCB	FALSE	1.13	Indicates whether ASCB data must be present on the bus prior to starting automatic recording, if enabled.
REC_STOP_NO_ASCB	FALSE	1.13	Indicates to automatically stop any active recordings when no ASCB data is present.

The parameters listed below are also valid but have been replaced by other parameters for the sake of clarification or continuity. They will remain valid for all DataTap-10 versions going forward.

Parameter	Default	Added in Version	Description
ENHANCED_ASCB_MODE	FALSE	1.1	Superseded by ASCB_VERSION.
PARTIAL_REC	no	1.2	Synonymous with DEC_REC_EN.
PARTIAL_SEND	yes	1.2	Synonymous with UDP_DEC_EN_DEF.
FULL_REC	no	1.2	Synonymous with RAW_REC_EN.
FULL_SEND	yes	1.2	Synonymous with UDP_RAW_EN_DEF.
PARTIAL_REC_FILE	ASCB_DEC	1.2	Synonymous with DEC_REC_FILE.

FULL_REC_FILE	ASCB_RAW	1.2	Synonymous with RAW_REC_FILE.
PARTIAL_SEND_REC_FORMAT	TISNATIVE	1.2	Synonymous with DEC_SEND_REC_FORMAT.
FULL_SEND_REC_FORMAT	TISNATIVE	1.2	Synonymous with RAW_SEND_REC_FORMAT.
PARTIAL_SEND_[1:32]_RATE	80	1.2	Synonymous with DEC_SEND_[1:32]_RATE.
HW_PDD_ENABLE	FALSE	1.3	Superseded by HW_TX_PDD_ENABLE.
USE_STS_LED_FOR_REC_STS	FALSE	1.4	Superseded by STS_LED_USE.
ASCB_REC_APPEND	FALSE	1.10	Superseded by REC_APPEND_MODE.
ASCB_CIRC_REC_MODE	FALSE	1.10	Superseded by CIRC_REC_MODE.
ASCB_REC_WAIT_DATA	FALSE	1.10	Superseded by AUTO_REC_WAIT_ASCB.

The table below lists the applicable software version for each revision of the system configuration file:

Config File Version	Software Version
1.1	-
1.2	10-Sep-13
1.3	24-Apr-14
1.4	28-May-14
1.5	6-Aug-14
1.6	26-Aug-14
1.7	15-Jan-15
1.8	10-Feb-15
1.9	17-Jun-15
1.10	15-Sep-15
1.11	04-Feb-16
1.12	17-Jun-16
1.13	13-Dec-16

7.2. DataTap-10 System Software Updates

The DataTap-10 platform has the ability to perform system software updates. The software files must be loaded onto the base directory of the microSD card by either removing the card from the unit or by using File Transfer Protocol (FTP). Once the files have been loaded onto the microSD card, the unit will automatically reprogram internal flash memory on the next power cycle and delete the files from the microSD card. The names of the software files MUST be “mb0.srd” and “mb1.srd” in order for the software update to take place. Note that if power is removed during the programming process, the DataTap-10 will simply re-attempt the programming on the next power cycle because the file(s) will still exist on the microSD card.

A file named “install.log” is created on the microSD card by the DataTap-10 which provides information regarding the update. It will look like the following on a successful update:

```
Programming File 'mb0.srd' found:
   MB0 Software Programming Successful.
   MB0 Software Verification Successful.
   MB0 Software File Deleted.
Programming File 'mb1.srd' found:
   MB1 Software Programming Successful.
   MB1 Software Verification Successful.
   MB1 Software File Deleted.
```

The update process takes less than 30 seconds once power is cycled. The activity LEDs will turn on one at a time in a rectangular pattern while programming is in progress. The unit will begin responding to ping requests over the Ethernet link once the new software has been programmed. If desired, the time and date of the new software can be verified via Telnet (see section 7.4). The unit is then ready for use.

7.3. Transferring Files to and from the DataTap-10

7.3.1. MicroSD Card Removal

The first method of transferring files to and from the microSD card is to remove the card from the DataTap-10 and place it into a card reader connected to a PC. The DataTap-10 supports both FAT16 and FAT32 file system formats. Any files written to the microSD card MUST be written to the base directory of the card and not placed into a sub-directory; otherwise they will not be seen by the DataTap-10.

7.3.2. File Transfer Protocol

The second method to transfer files to and from the DataTap-10 is to use File Transfer Protocol (FTP) from a PC. The IP address of the DataTap-10 to use for the FTP session is specified in the “system.cfg” file as BOARD_IP. Note that no user or password checking is performed by the DataTap-10 FTP server. Just hitting <enter> when prompted for both is acceptable.

Files may be transferred to and from the microSD card as well as the CompactFlash card, if present. At the start of an FTP session, access is initially directed to the microSD card. This can be changed by issuing the command “quote site cf”, which will redirect all future commands for the current FTP session to the CompactFlash card.

Note: After initially connecting to the FTP server, the Telnet server will show the selected recording media as microSD, regardless of the setting in the system configuration file. It may also change if the “quote site cf” or “quote site msd” commands are issued. The selected recording media will always switch back to the media type specified in the system configuration file with the USE_COMPACT_FLASH parameter when a new recording or playback is started.

The following FTP commands are supported:

FTP Command	Description
dir	Lists the files currently on the selected media card. Wilcard characters may be used.
bin	Sets the file transfer mode to binary.

ascii	Sets the file transfer mode to ASCII.
get <filename>	Retrieves a file from the selected media card.
mget <filename>	Retrieves multiple files from the selected media card. Wildcard characters may be used.
del <filename>	Deletes a specified file or multiple files from the selected media card. Wildcard characters may be used.
put <filename>	Stores a specified file to the selected media card.
mput <filename>	Stores multiple files to the selected media card. Wildcard characters may be used.
ren <fn1> <fn2>	Renames a file from filename <fn1> to filename <fn2> on the selected media card.
quote site reset	Indicates to reboot the DataTap-10, similar to a power cycle.
quote site format	Performs a FAT32 formatting of the selected media card regardless of its current file system. To avoid inadvertent formatting, it is not possible to reformat a microSD card when the USE_COMPACT_FLASH parameter in the system configuration file is set to TRUE.
quote site full format	Performs a full FAT32 format of the selected media card regardless of its current file system. A full format is performed by first issuing erase commands to the entire card, greatly improving the recording performance of the card and the wear-leveling. It then creates a FAT32 partition. Note that this command is only supported for the microSD card. It is highly recommended to use this command rather than the regular format command when recording to the microSD card. It can add anywhere from 10 to 30 seconds for a new 64GB Samsung PRO microSD card depending upon the current state of the card.
quote site udp off	Turns off all UDP output from the DataTap-10. This can be useful when retrieving large files from the selected media card and may help increase the transfer rate due to CPU time being freed up. The UDP output can only be turned back on via Telnet, UDP command packets, or by cycling power on the DataTap-10.
quote site cf	Switches the selected media type to CompactFlash. All future commands during the current FTP session will be directed to the CompactFlash card.
quote site msd	Switches the selected media type to microSD. All future commands during the current FTP session will be directed to the microSD card.
quote site fill	Fills the selected media card with empty 2GB files with the name EMPTY_0.01, .02, etc. until less than 2GB remains on the card. This command must be performed after a reformat of the card when recording in Circular Recording Mode. It will be performed automatically at power-up if Circular Recording Mode is enabled and no valid recording files exist on the card.

An example FTP session for storing the system configuration file to the microSD card is shown below.

```
MS-DOS>ftp 10.1.1.2
Connected to 10.1.1.2.
220 Welcome to the DataTap-10 FTP Server.
User (10.1.1.2:(none)):
331 Password required.
```

```
Password:
230 Login Successful.
ftp> put system.cfg
200 PORT command successful.
150 Opening BINARY data connection.
226 Transfer Complete.
ftp: 171 bytes sent in 0.00Seconds 171000.00Kbytes/sec.
ftp> dir
200 PORT command successful.
150 Here comes the directory listing.

-rw-r--r-- 1 user group         58579 Sep 14 15:10 SYSTEM.CFG
-rw-r--r-- 1 user group           579 Sep 14 01:42 INSTALL.LOG

226 Transfer Complete.
ftp: 54 bytes received in 0.00Seconds 54000.00Kbytes/sec.
ftp> bye
221 Goodbye.
```

An example FTP session for updating the DataTap-10 system software is shown below. Note that the "quote site reset" command is used following the "put" commands such that a power-cycle is not required.

```
MS-DOS>ftp 10.1.1.2
Connected to 10.1.1.2.
220 Welcome to the DataTap-10 FTP Server.
User (10.1.1.2:(none)):
331 Password required.
Password:
230 Login Successful.
ftp> bin
200 Switching to BINARY mode.
ftp> put mb0.srd
200 PORT command successful.
150 Opening data connection.
226 Transfer Complete.
ftp: 126870 bytes sent in 0.11Seconds 1163.94Kbytes/sec.
ftp> put mb1.srd
200 PORT command successful.
150 Opening data connection.
226 Transfer Complete.
ftp: 359458 bytes sent in 0.59Seconds 606.17Kbytes/sec.
ftp> dir
200 PORT command successful.
150 Here comes the directory listing.

-rw-r--r-- 1 user group         58579 Sep 14 15:10 SYSTEM.CFG
-rw-r--r-- 1 user group        379108 Sep 12 11:15 TIS.CFG
-rw-r--r-- 1 user group       1086258 Sep 14 15:09 MB0.SRD
-rw-r--r-- 1 user group         79438 Sep 14 15:09 MB1.SRD
```

```
226 Transfer Complete.
ftp: 109 bytes received in 0.00Seconds 109000.00Kbytes/sec.
ftp> quote site reset
Connection closed by remote host.
ftp> bye
```

7.4. Telnet Remote Status Monitoring

The DataTap-10 contains a Telnet server in order to display configuration and status information. Once the DataTap-10 begins responding to ping requests following a power cycle, a user may connect with a Telnet client. The recommended client is the standard Windows Telnet client. From an MS-DOS prompt, simply type “telnet 192.168.1.1” and the Telnet client will connect with the DataTap-10. Note that the IP address in the command must match what is specified in the system configuration file (system.cfg) for the BOARD_IP parameter.

The basic Telnet commands are shown below, which are just key presses during the Telnet session. Note that letter keys can be upper-case or lower-case. A list of all available commands for a specific page can be accessed by pressing ‘h’ while on that page. Note that some pages can only be accessed by using the left and right arrow keys rather than a number or letter.

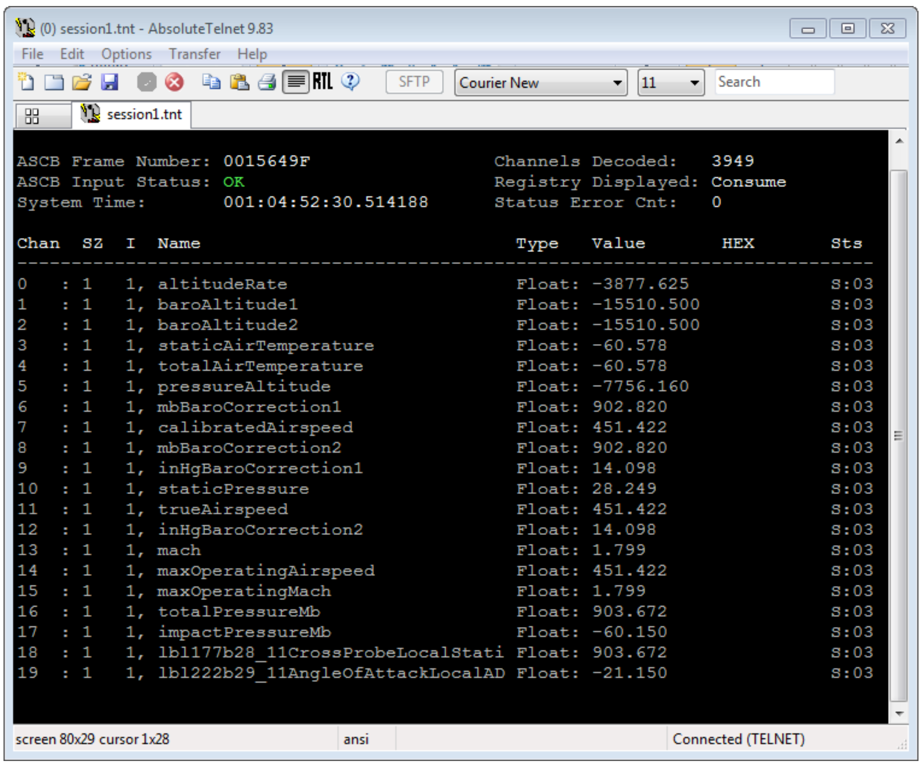
Key(s)	Description
→	Switches to the next Telnet page.
←	Switches to the previous Telnet page.
1	Switches to the ASCB-D Parameter Decode page.
2	Switches to the ASCB-D Status page.
3	Switches to the System Status page.
4	Switches to the Initialization Status Page.
5	Switches to the Recording Page.
6	Switches to the ARINC Channel 0 and 1 Receive page.
7	Switches to the ARINC Channel 2 and 3 Receive page.
8	Switches to the ARINC Transmit page.
9	Switches to the ARINC-717 Receive Page.
n	Switches to the Networking Configuration Page.
g	Switches to the ASCB-D Group Structure page.
m	Switches to the Ethernet MAC Statistics page.
↓	Scrolls the display to the next group of data for the current page.
↑	Scrolls the display to the previous group of data for the current page.
c	Resets the scrolling back to the first group of displayed data for the current page.
<space>	Clears any clearable status data on the current page.
h	Switches to a Help page listing the available commands for the current page. Pressing ‘h’ again will exit the help.

The screen shots below show the available Telnet screens.

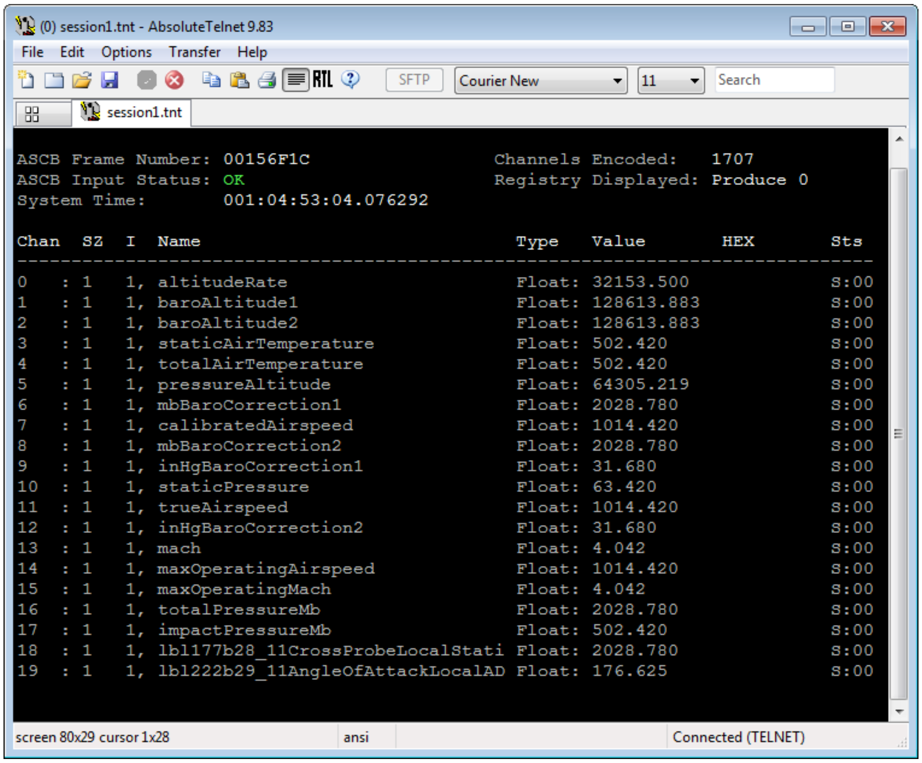
7.4.1. ASCB Consumed Parameter Page

This page shows the decoded parameters for both ASCB busses based on the TIS configuration file (tis.cfg) residing on the microSD card. Pressing ‘r’ while on the ASCB Parameter Page will cycle through the consumed parameters, the produced parameters for ASCB bus 0, and the produced parameters for ASCB bus 1, as shown in the following three images.

7.4.1.1.Consumed Parameters

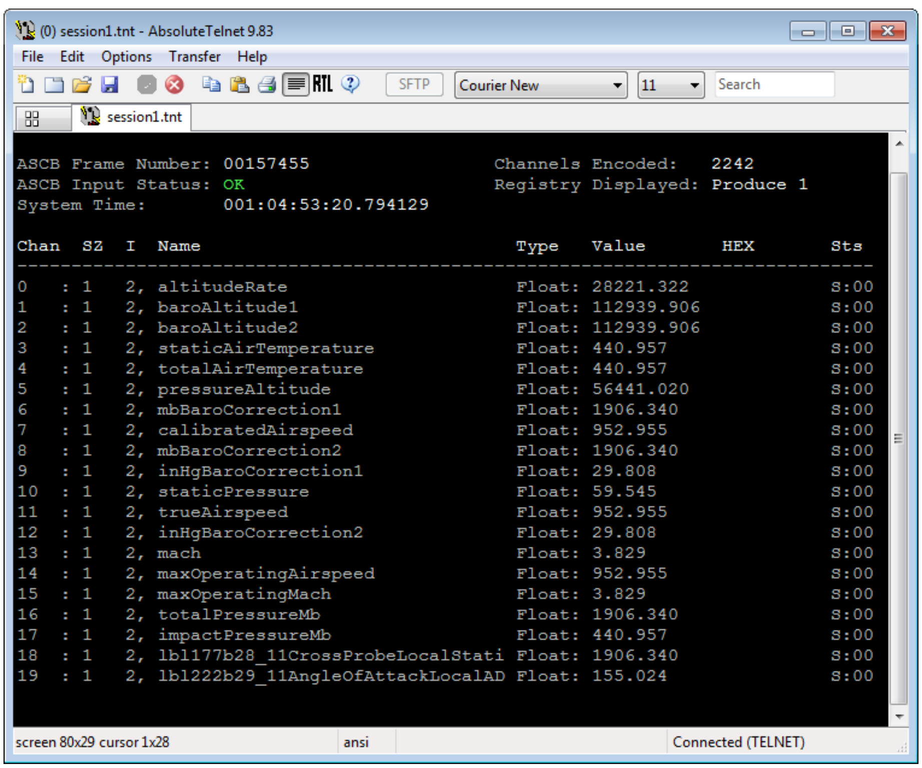


7.4.1.2.Bus 0 Produced Parameters



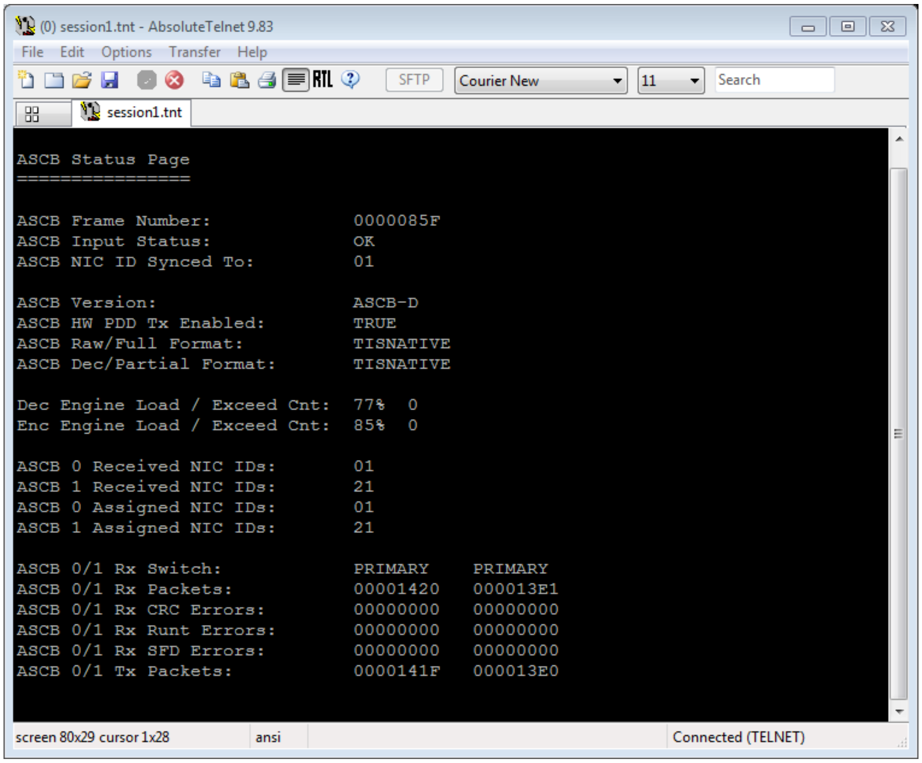
```
ASCB Frame Number: 00156F1C                    Channels Encoded:    1707
ASCB Input Status: OK                          Registry Displayed: Produce 0
System Time:       001:04:53:04.076292

Chan   SZ  I   Name                             Type   Value        HEX        Sts
--------------------------------------------------------------------------------
0    : 1    1, altitudeRate                     Float: 32153.500               S:00
1    : 1    1, baroAltitude1                    Float: 128613.883              S:00
2    : 1    1, baroAltitude2                    Float: 128613.883              S:00
3    : 1    1, staticAirTemperature             Float: 502.420                 S:00
4    : 1    1, totalAirTemperature              Float: 502.420                 S:00
5    : 1    1, pressureAltitude                 Float: 64305.219               S:00
6    : 1    1, mbBaroCorrection1                Float: 2028.780                S:00
7    : 1    1, calibratedAirspeed               Float: 1014.420                S:00
8    : 1    1, mbBaroCorrection2                Float: 2028.780                S:00
9    : 1    1, inHgBaroCorrection1              Float: 31.680                  S:00
10   : 1    1, staticPressure                   Float: 63.420                  S:00
11   : 1    1, trueAirspeed                     Float: 1014.420                S:00
12   : 1    1, inHgBaroCorrection2              Float: 31.680                  S:00
13   : 1    1, mach                             Float: 4.042                   S:00
14   : 1    1, maxOperatingAirspeed             Float: 1014.420                S:00
15   : 1    1, maxOperatingMach                 Float: 4.042                   S:00
16   : 1    1, totalPressureMb                  Float: 2028.780                S:00
17   : 1    1, impactPressureMb                 Float: 502.420                 S:00
18   : 1    1, lbl177b28_11CrossProbeLocalStati Float: 2028.780                S:00
19   : 1    1, lbl222b29_11AngleOfAttackLocalAD Float: 176.625                 S:00
```

7.4.1.3.Bus 1 Produced Parameters

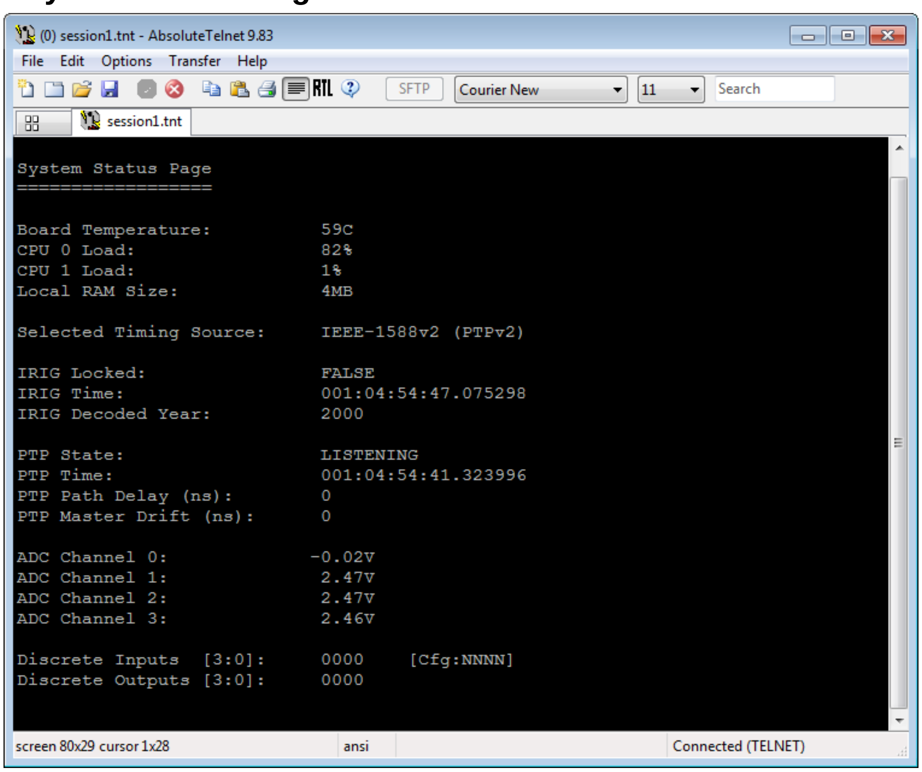


```
ASCB Frame Number: 00157455                    Channels Encoded:    2242
ASCB Input Status: OK                          Registry Displayed: Produce 1
System Time:       001:04:53:20.794129

Chan   SZ  I   Name                             Type   Value        HEX        Sts
--------------------------------------------------------------------------------
0    : 1    2, altitudeRate                     Float: 28221.322               S:00
1    : 1    2, baroAltitude1                    Float: 112939.906              S:00
2    : 1    2, baroAltitude2                    Float: 112939.906              S:00
3    : 1    2, staticAirTemperature             Float: 440.957                 S:00
4    : 1    2, totalAirTemperature              Float: 440.957                 S:00
5    : 1    2, pressureAltitude                 Float: 56441.020               S:00
6    : 1    2, mbBaroCorrection1                Float: 1906.340                S:00
7    : 1    2, calibratedAirspeed               Float: 952.955                 S:00
8    : 1    2, mbBaroCorrection2                Float: 1906.340                S:00
9    : 1    2, inHgBaroCorrection1              Float: 29.808                  S:00
10   : 1    2, staticPressure                   Float: 59.545                  S:00
11   : 1    2, trueAirspeed                     Float: 952.955                 S:00
12   : 1    2, inHgBaroCorrection2              Float: 29.808                  S:00
13   : 1    2, mach                             Float: 3.829                   S:00
14   : 1    2, maxOperatingAirspeed             Float: 952.955                 S:00
15   : 1    2, maxOperatingMach                 Float: 3.829                   S:00
16   : 1    2, totalPressureMb                  Float: 1906.340                S:00
17   : 1    2, impactPressureMb                 Float: 440.957                 S:00
18   : 1    2, lbl177b28_11CrossProbeLocalStati Float: 1906.340                S:00
19   : 1    2, lbl222b29_11AngleOfAttackLocalAD Float: 155.024                 S:00
```

7.4.2. ASCB Status Page

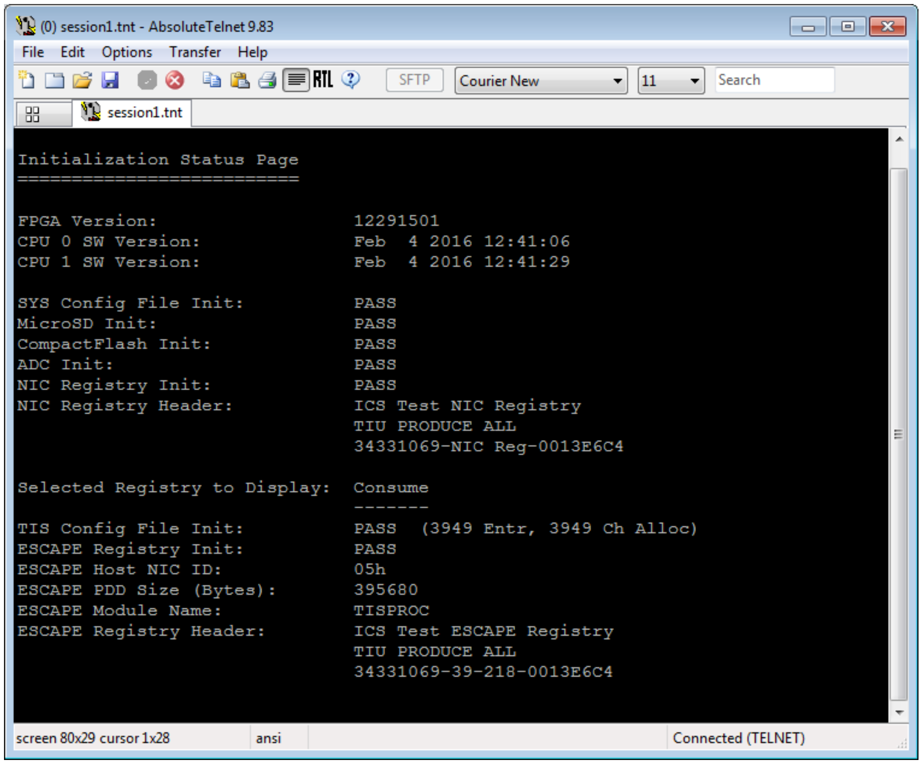


7.4.3. System Status Page

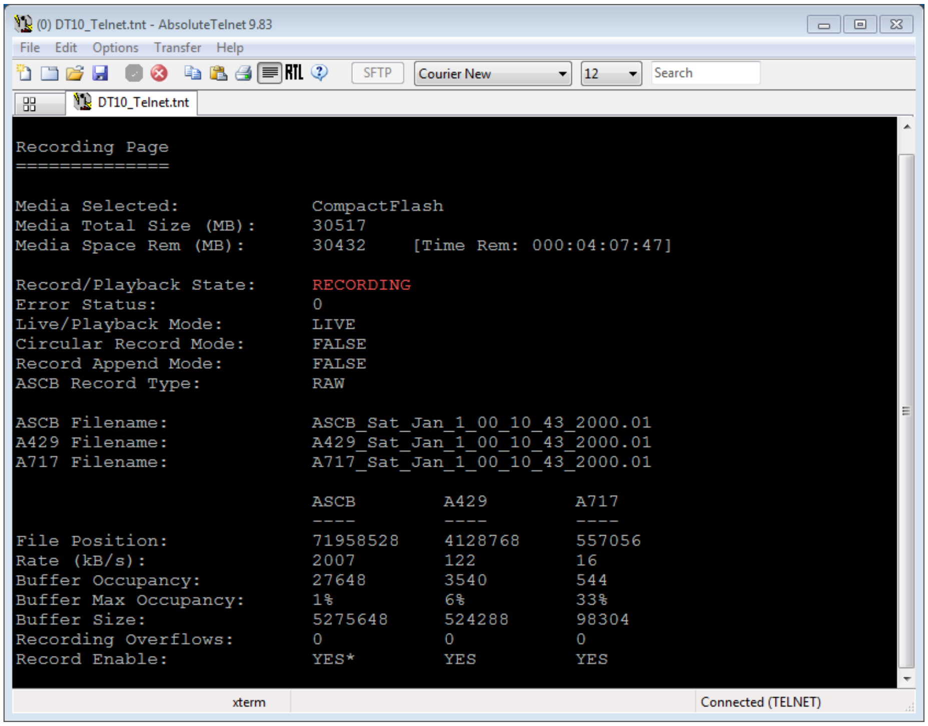


7.4.4. Initialization Status Page

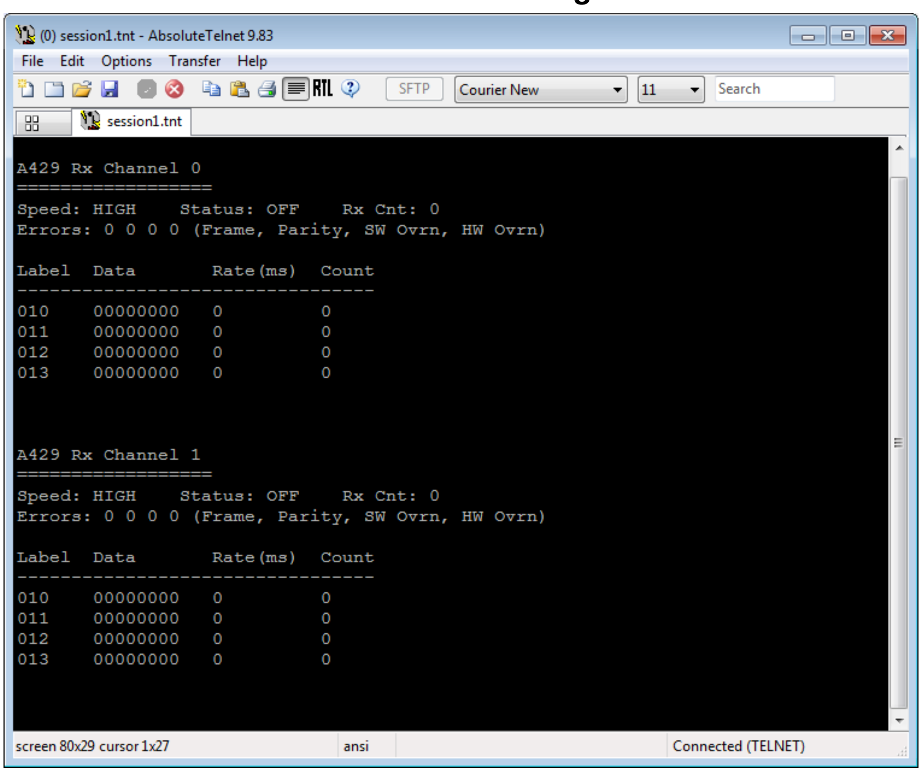
Pressing ‘r’ while on the Initialization Status Page will cycle through the ESCAPE registry and TIS configuration file information for the three types: Consume, Bus 0 Produce, and Bus 1 Produce. The ESCAPE registry information is pulled from the registry headers programmed into the DataTap-10’s non-volatile memory. The TIS configuration file information is pulled from the microSD card at power-up.



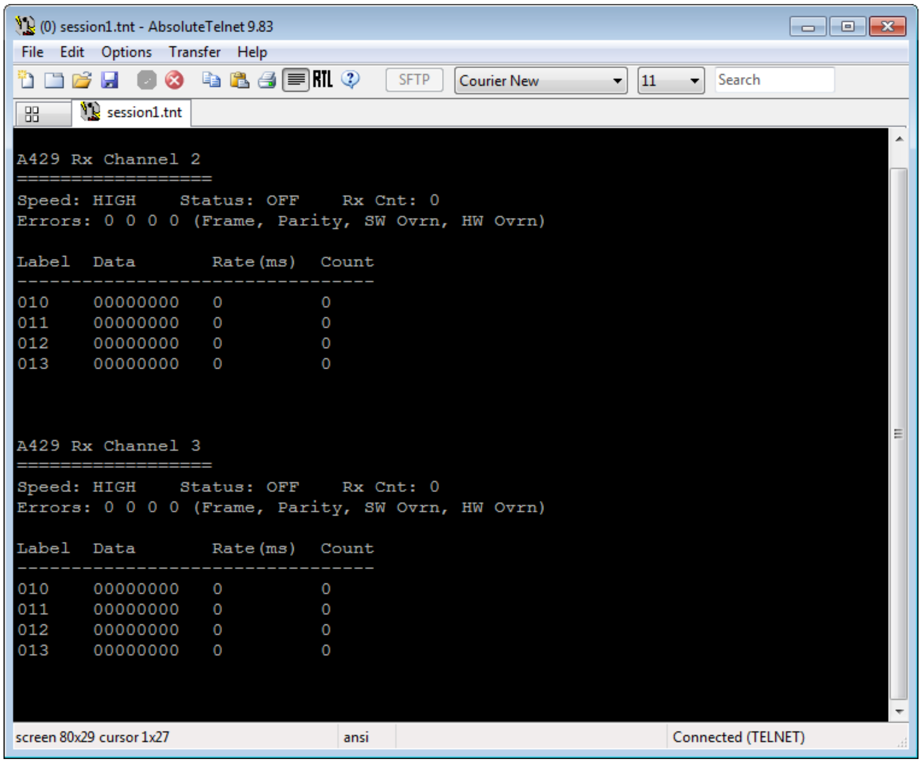
7.4.5. Recording Page



7.4.6. ARINC-429 Receive Channels 0/1 Page

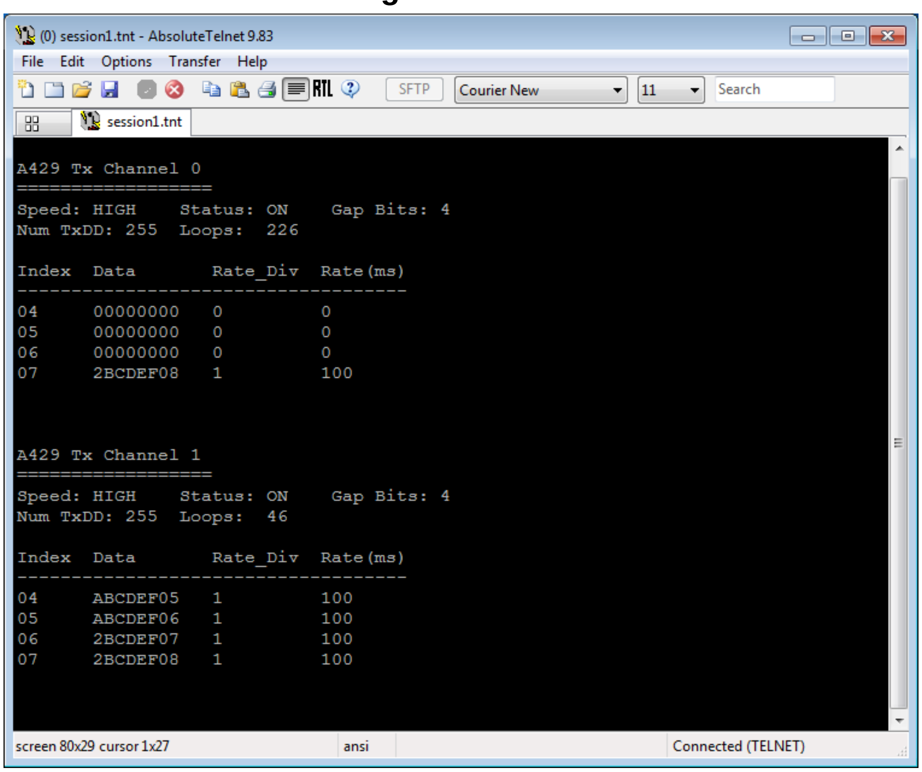


7.4.7. ARINC-429 Receive Channels 2/3 Page



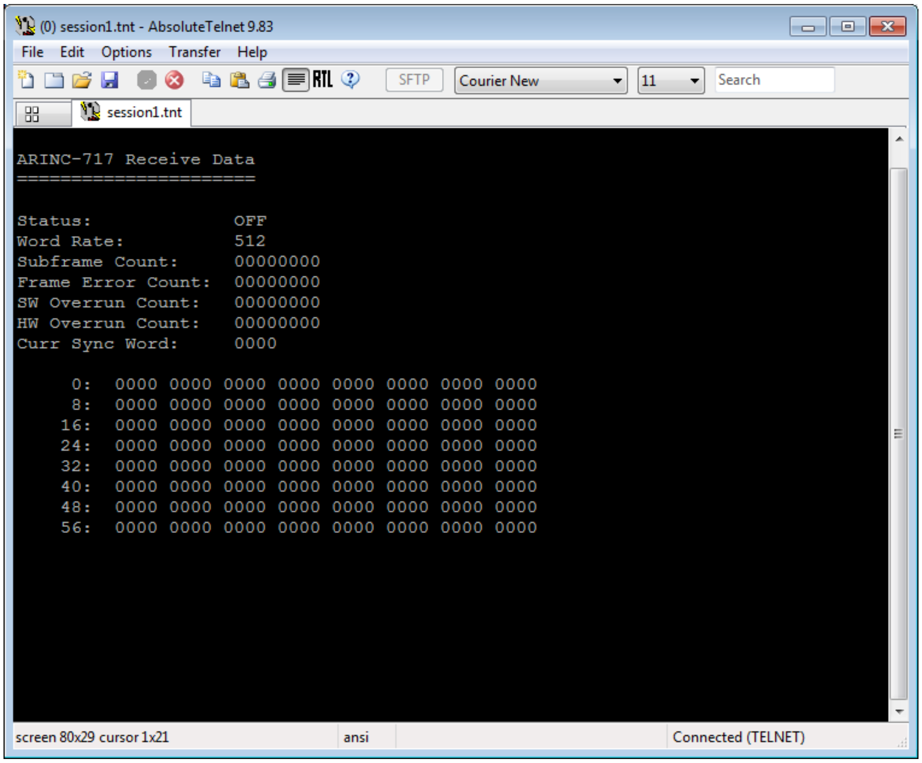
```
A429 Rx Channel 2
==================
Speed: HIGH    Status: OFF    Rx Cnt: 0
Errors: 0 0 0 0 (Frame, Parity, SW Ovrn, HW Ovrn)

Label  Data        Rate(ms)  Count
---------------------------------
010    00000000    0         0
011    00000000    0         0
012    00000000    0         0
013    00000000    0         0


A429 Rx Channel 3
==================
Speed: HIGH    Status: OFF    Rx Cnt: 0
Errors: 0 0 0 0 (Frame, Parity, SW Ovrn, HW Ovrn)

Label  Data        Rate(ms)  Count
---------------------------------
010    00000000    0         0
011    00000000    0         0
012    00000000    0         0
013    00000000    0         0
```

7.4.8. ARINC-429 Transmit Page

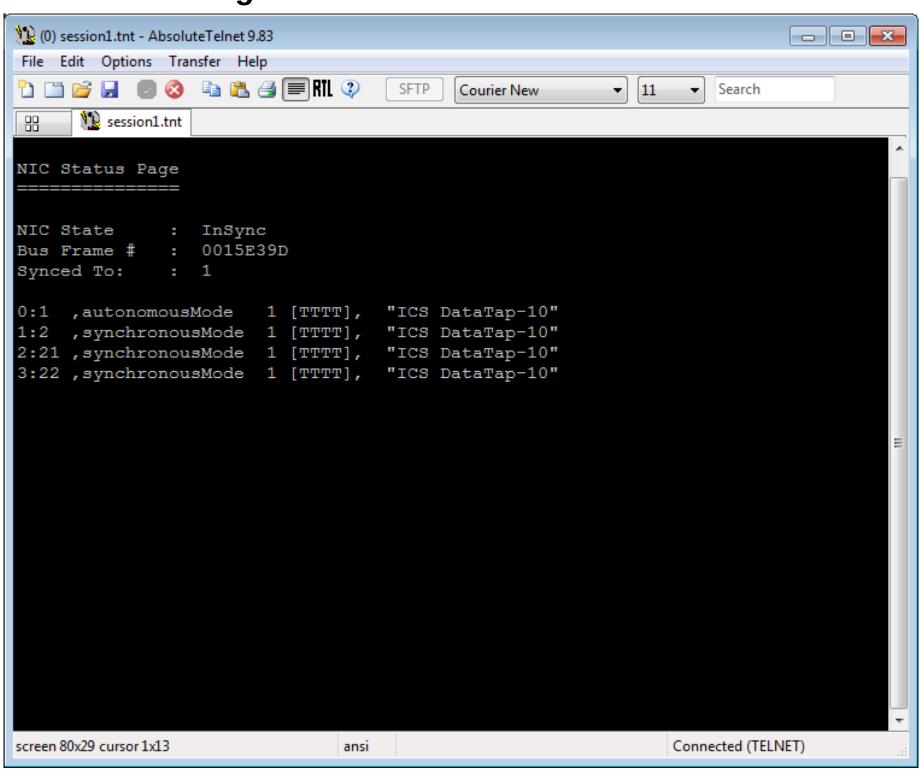


```
A429 Tx Channel 0
==================
Speed: HIGH    Status: ON    Gap Bits: 4
Num TxDD: 255  Loops:  226

Index  Data        Rate_Div  Rate(ms)
------------------------------------
04     00000000    0         0
05     00000000    0         0
06     00000000    0         0
07     2BCDEF08    1         100


A429 Tx Channel 1
==================
Speed: HIGH    Status: ON    Gap Bits: 4
Num TxDD: 255  Loops:  46

Index  Data        Rate_Div  Rate(ms)
------------------------------------
04     ABCDEF05    1         100
05     ABCDEF06    1         100
06     2BCDEF07    1         100
07     2BCDEF08    1         100
```

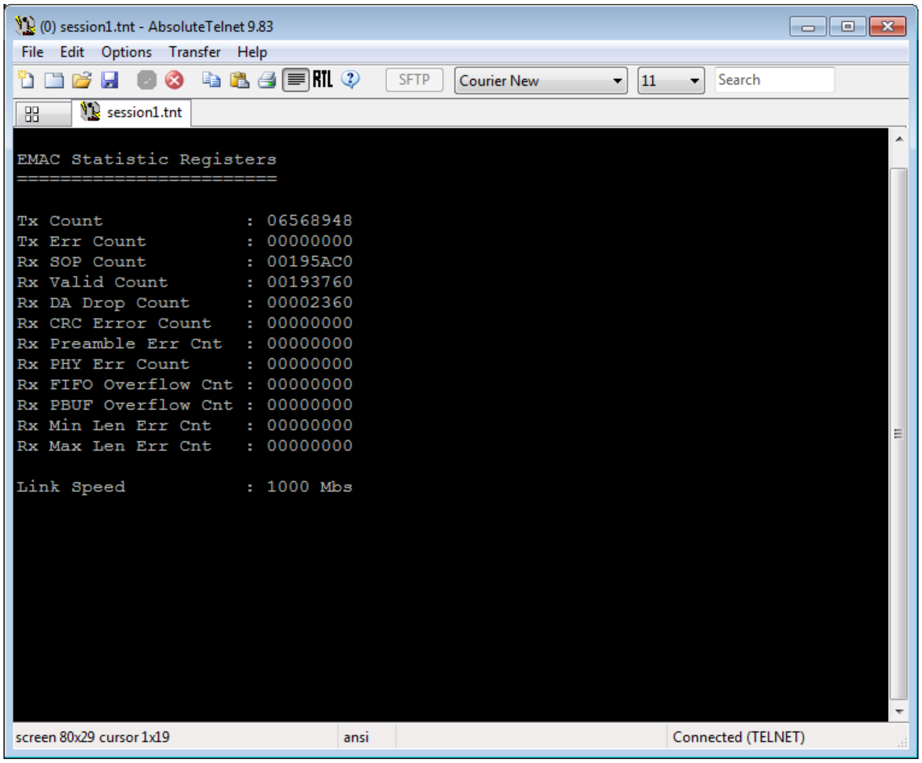
7.4.9. ARINC-717 Receive Page



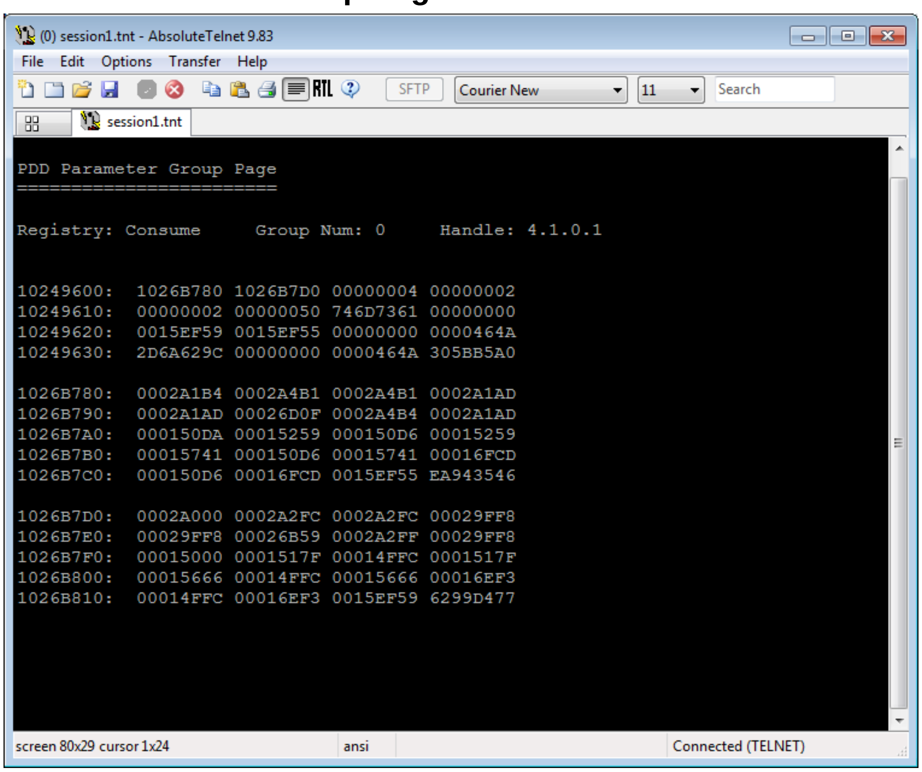
7.4.10.NIC Status Page



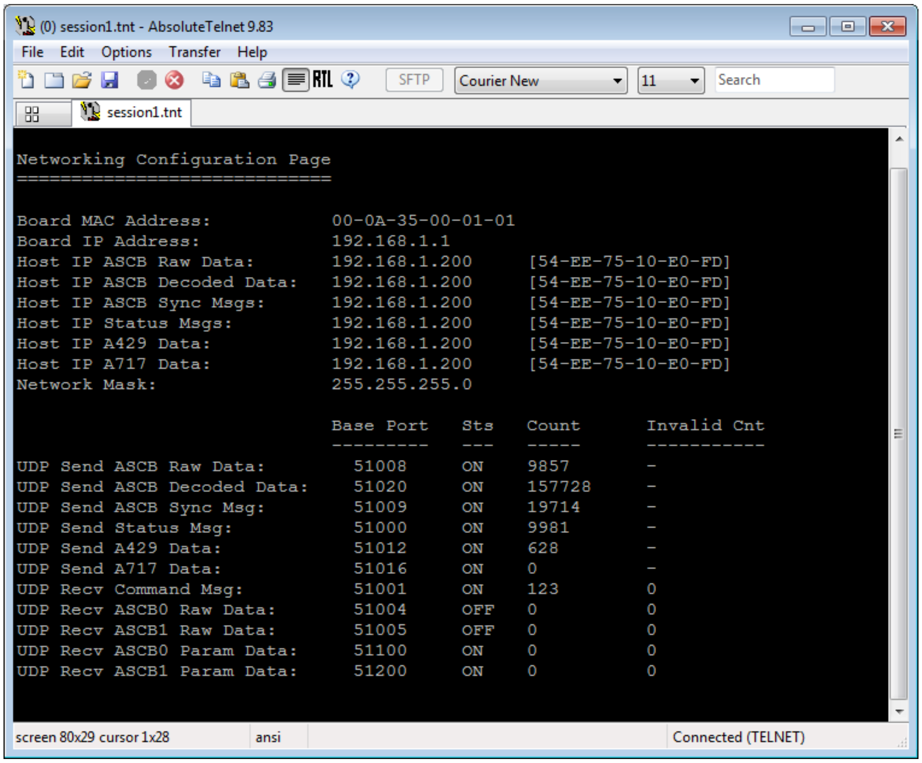
7.4.11.Ethernet MAC Statistics Page



7.4.12.PDD Parameter Group Page

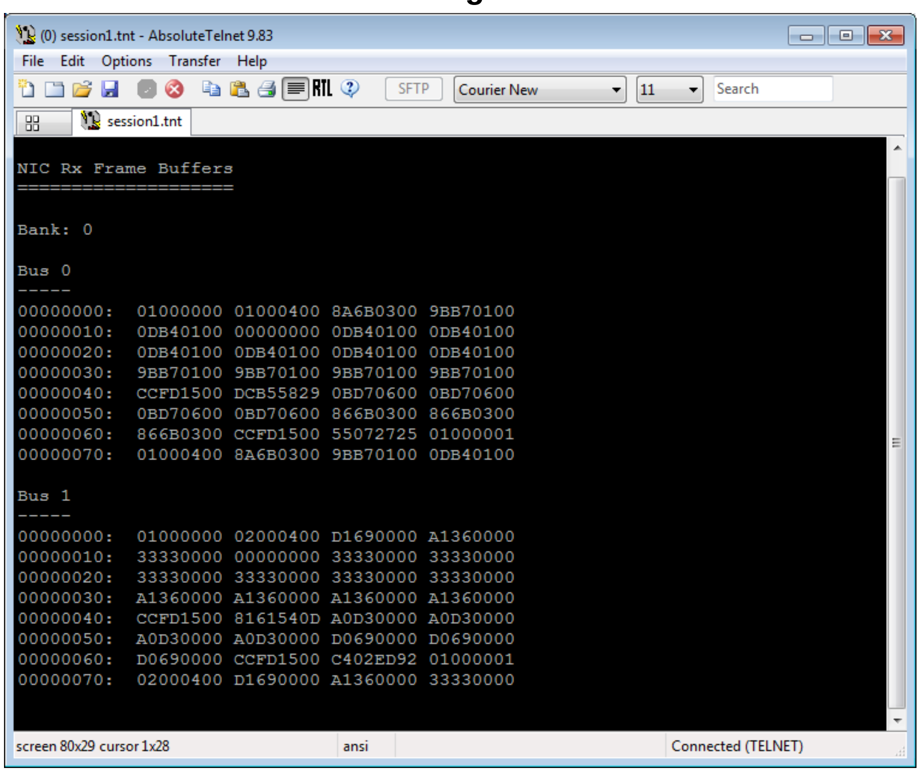


7.4.13.Networking Configuration Page



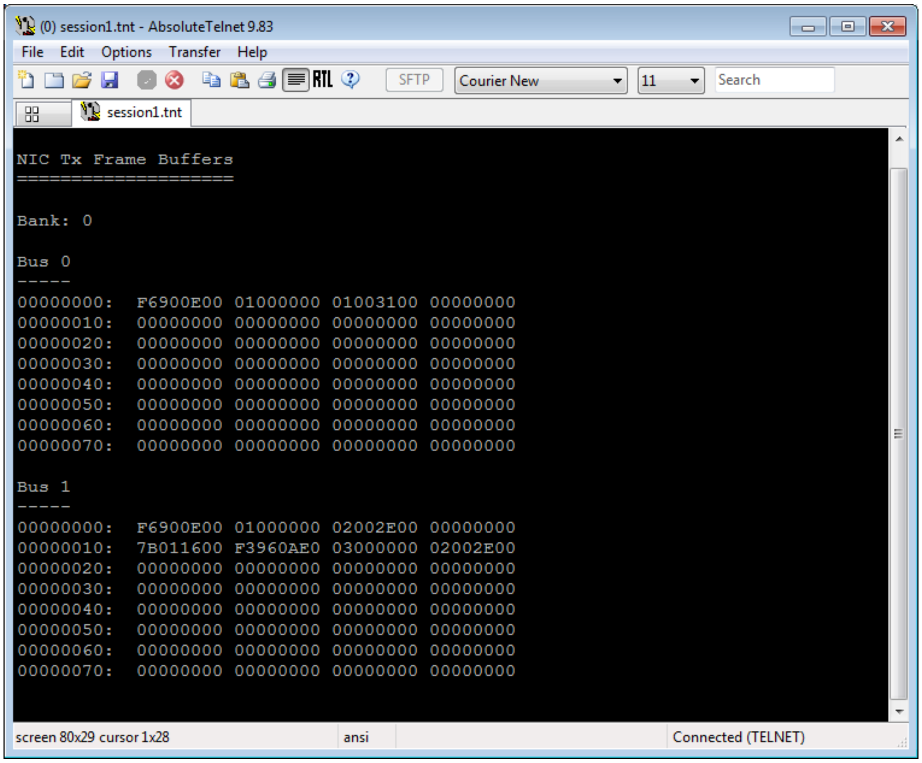
```
Networking Configuration Page
=============================

Board MAC Address:          00-0A-35-00-01-01
Board IP Address:           192.168.1.1
Host IP ASCB Raw Data:      192.168.1.200     [54-EE-75-10-E0-FD]
Host IP ASCB Decoded Data:  192.168.1.200     [54-EE-75-10-E0-FD]
Host IP ASCB Sync Msgs:     192.168.1.200     [54-EE-75-10-E0-FD]
Host IP Status Msgs:        192.168.1.200     [54-EE-75-10-E0-FD]
Host IP A429 Data:          192.168.1.200     [54-EE-75-10-E0-FD]
Host IP A717 Data:          192.168.1.200     [54-EE-75-10-E0-FD]
Network Mask:               255.255.255.0

                            Base Port   Sts   Count      Invalid Cnt
                            ---------   ---   -----      -----------
UDP Send ASCB Raw Data:       51008     ON    9857       -
UDP Send ASCB Decoded Data:   51020     ON    157728     -
UDP Send ASCB Sync Msg:       51009     ON    19714      -
UDP Send Status Msg:          51000     ON    9981       -
UDP Send A429 Data:           51012     ON    628        -
UDP Send A717 Data:           51016     ON    0          -
UDP Recv Command Msg:         51001     ON    123        0
UDP Recv ASCB0 Raw Data:      51004     OFF   0          0
UDP Recv ASCB1 Raw Data:      51005     OFF   0          0
UDP Recv ASCB0 Param Data:    51100     ON    0          0
UDP Recv ASCB1 Param Data:    51200     ON    0          0
```

7.4.14.NIC Receive Frame Buffer Page



```
NIC Rx Frame Buffers
====================

Bank: 0

Bus 0
-----
00000000:  01000000 01000400 8A6B0300 9BB70100
00000010:  0DB40100 00000000 0DB40100 0DB40100
00000020:  0DB40100 0DB40100 0DB40100 0DB40100
00000030:  9BB70100 9BB70100 9BB70100 9BB70100
00000040:  CCFD1500 DCB55829 0BD70600 0BD70600
00000050:  0BD70600 0BD70600 866B0300 866B0300
00000060:  866B0300 CCFD1500 55072725 01000001
00000070:  01000400 8A6B0300 9BB70100 0DB40100

Bus 1
-----
00000000:  01000000 02000400 D1690000 A1360000
00000010:  33330000 00000000 33330000 33330000
00000020:  33330000 33330000 33330000 33330000
00000030:  A1360000 A1360000 A1360000 A1360000
00000040:  CCFD1500 8161540D A0D30000 A0D30000
00000050:  A0D30000 A0D30000 D0690000 D0690000
00000060:  D0690000 CCFD1500 C402ED92 01000001
00000070:  02000400 D1690000 A1360000 33330000
```

7.4.15.NIC Transmit Frame Buffer Page



8. DataTap-10 Telemetry-Acquisition Interface System (TIS) Functionality

The Telemetry-Acquisition Interface System is a complex data translator and optionally a recorder. The platform itself decodes ASCB Version D or Enhanced ASCB data from the Pilot and/or Co-Pilot side of the aircraft and makes this data available to a telemetry or data acquisition system. The interface to the telemetry or data acquisition system is in the form of a high-speed UDP Ethernet based link that is easily changed to the needs of various user systems.

The platform must be directed which ASCB parameters to read and what to do with this data. An ICS written Windows based configuration utility is used to define what data channel (UDP Datagram position) the decoded ASCB data is to appear in to the Telemetry or Data Acquisition System. The functionality to select a set of parameters to decode in a stand-alone mode and program them into the DataTap-10 is built in to the FlightLine software. The program produces a file of parameters in a format defined in section 10.1 DataTap-10 CFG File Format. This file is renamed to tis.cfg and placed on the µSDHC card of the DataTap-10.

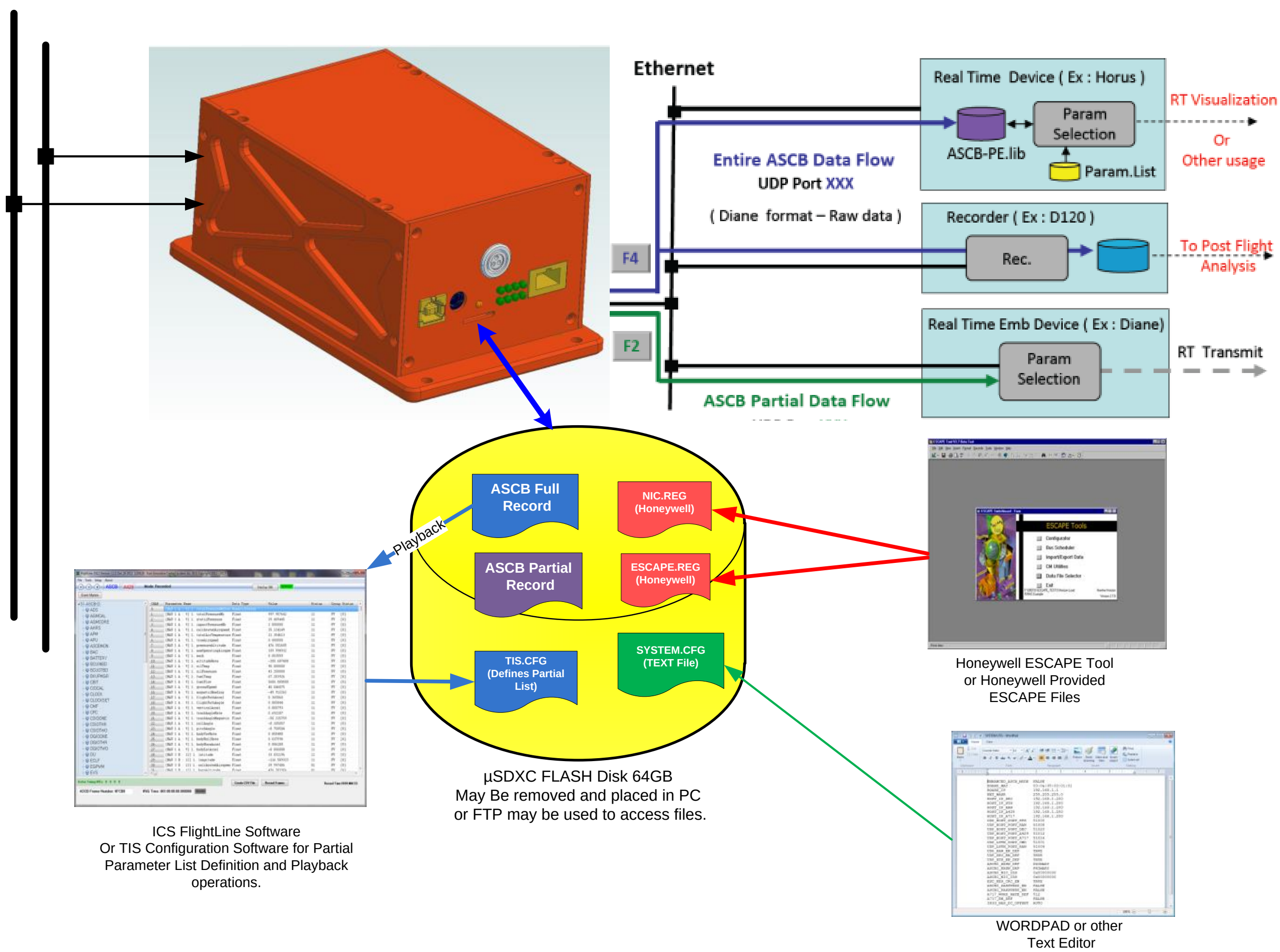


Figure 12 DT-10 Based TIS Conversion Platform Block Diagram

8.1. System Overview and Introduction

The DataTap-10 platform is designed to be a stand-alone conversion unit to read and decode parameters from the Honeywell ASCB Version D or Enhanced ASCB bus and to make these available to the end users telemetry equipment in a format more easily interfaced with.

The DataTap-10 is designed to be turned on and to come up converting data. The power-on operation is automatic. The unit will boot up and start the DataTap-10 software. Power off may be accomplished by simply turning off the power to the unit

8.2. System Power Up and Shutdown

The DataTap-10 unit may simply be turned on for power-up operation. The unit will come up running the DataTap-10 software automatically and will use the last configuration files that were loaded to direct its operation. There is more information on the configuration files later in this section. Typical power-up times are 10 seconds or less but is mainly dependent on the number of entries in the TIS.CFG configuration file.

For shutdown, the power is simply removed. To prevent file corruption of recording data, the recording file is opened, written, and closed at the 80HZ frame rate of the ASCB-D bus. It is possible to lose the last frame of data in a recording but this represents only the last 12.5 ms of the recording and does not prevent the remainder of the file from being downloaded or played back at a later time.

8.3. System File Requirements

The DataTap-10 platform must be directed which ASCB parameters to read and what to do with this data. A Windows XP, Vista, and Windows 7 based configuration utility is used to define what ASCB D data is to be read and placed onto a parallel interface or sent out an Ethernet based interface to the Telemetry and Acquisition equipment.

There are 3 basic software generated items from the Honeywell ESCAPE tool that are needed components for the DataTap-10 system before data conversion can take place.

These are:

ESCAPE generate NIC.le_reg for the aircraft configuration being tested
ESCAPE generated Consume all LE registry file (Escape.le_reg)
ESCAPE ASCB Configuration Database (Needed only for FlightLine software)

9. UDP Datagram Formats

9.1. Decoded Parameter UDP Output Datagrams (TISNATIVE Format)

Decoded parameter data from the DataTap-10 ASCB-D platform consists of from 1 to 8192 pieces of data (32-bits each) called "channels". The ASCB parameter that occurs in a particular channel is definable using the ICS provided Windows based configuration tool. This tool produces the tis.cfg file that is loaded into the DataTap-10 via the microSD card.

The configuration file defines which ASCB-D parameters will be converted and transmitted to the user-connected equipment and in what "channels" they will appear. This configuration file does not need to change from aircraft load to aircraft load unless a parameter is deleted from the new aircraft load, or changes data type.

The format of transmission from the DataTap-10 ASCB-D platform to the user's platform is via UDP datagrams, which is a well-defined protocol. Definition of UDP datagrams will not be discussed here except to define the payload of the individual datagrams as they pertain to the DataTap-10 ASCB-D platform data flow.

The DataTap-10 ASCB-D platform will transmit 1 to 32 datagrams of up to 1288 bytes at a rate defined for each datagram with the DEC_SEND_1_RATE through DEC_SEND_32_RATE definitions in the system configuration file. By sending up to 32 datagrams each containing 256 "channels", a maximum of 8192 channels may be sent. This is the maximum allowed on a single DataTap-10 ASCB-D platform.

Each Datagram contains up to 256 "channels" of 32-bit words of data from the ASCB D bus as well as 256 bytes of parameter status and some packet overhead. The format may be defined as TISNATIVE or as DIANE62, which is a Dassault format and has additional encapsulation around the data. The specific formats are referenced later in this section.

If the user configures the DataTap-10 ASCB-D platform for fewer than 256 channels of data (in positions 1-256), then only 1 datagram will be sent at the rate specified by PARTIAL_SEND_1_RATE. If more data is defined, then more datagrams will be sent as needed.

The size of the datagrams may be fixed or variable depending on the setting of "UDP_DEC_LEGACY_MODE" in the system configuration file. When set to TRUE, the datagrams will be a fixed 1288 bytes in length regardless of how many channels are occupied within the datagram (assuming at least one channel is occupied). When set to FALSE, the datagrams will only be as large as needed based on the highest configured channel within the datagram. As such, if only a few channels are occupied in a datagram, it is best to start with the first channel within the datagram and occupy consecutive channels from there. This way the datagram will not have wasted empty channels that will cause additional bandwidth over Ethernet. Both options are shown below in the structure definitions.

The UDP destination port base is configurable through the system configuration file (system.cfg) loaded on the DataTap-10's microSD card. Assuming the field UDP_HOST_PORT_DEC is set to 51020, the datagrams will look like the following:

Datagram 0 will contain channels [0..255] and will be sent as a UDP datagram at its programmed rate on UDP port 51020.
Datagram 1 will contain channels [256..511] and will be sent as a UDP datagram at its programmed rate on UDP port 51021.
…

Datagram 31 will contain channels [7935...8191] and will be sent as a UDP datagram at its programmed rate on UDP port 51051.

As each UDP datagram is on a different port, it is possible to know that data received on that port is always data from the channels associated with that datagram.

9.1.1. Data Structure of the Decoded Parameter UDP Datagrams (Fixed Length)

When SYSTEM.CFG value UDP_DEC_LEGACY_MODE = TRUE

```
typedef struct
{
   unsigned long LENGTH;
   unsigned long ASCBD_Frame;
   unsigned long CHANNEL[256];
   unsigned char STATUS[256];
   unsigned long PAD;

} DATAGRAM;
```

Where:

LENGTH	32-Bit value indicating the length of the UDP payload which includes the CHANNEL, STATUS, LENGTH, and ASCBD_Frame.
ASCBD_Frame	32-Bit increasing value that is the global frame number as read from the ASCB D bus.
CHANNEL	256 32-Bit values for the channels represented by the UDP datagram.
STATUS	256 8-Bit status values for the channels represented by the UDP datagram.
PAD	32-Bit PAD for possible future CRC value of the UDP payload, presently unused.

9.1.2. Data Structure of the Decoded Parameter UDP Datagrams (Variable Length)

When SYSTEM.CFG value UDP_DEC_LEGACY_MODE = FALSE

```
typedef struct
{
   unsigned long LENGTH;
   unsigned long ASCBD_Frame;
   unsigned long CHANNEL[up to 256];
   unsigned char STATUS[up to 256];

} DATAGRAM;
```

Where:

LENGTH	32-Bit value indicating the length of the UDP payload which includes the CHANNEL, STATUS, LENGTH, and ASCBD_Frame.
ASCBD_Frame	32-Bit increasing value that is the global frame number as read from the ASCB D bus.
CHANNEL	Up to 256 32-Bit values for the channels represented by the UDP datagram.
STATUS	Up to 256 8-Bit status values for the channels represented by the UDP datagram.

9.1.3. Status Byte Format

Each ASCB-D parameter has an associated status byte as shown above. These are 8-bit values with 4-bits actually having any meaning. Note that a status byte of 0xFF indicates the channel is unconfigured.

```
Bit                           Bit
7    6    5    4     3     2     1     0
----------------------------------
0    0    0    0    GS1   GS2   PS1   PS2
----------------------------------
```

Where

GS1 is Parameter Group Status (1 = **STALE**, 0 = **FRESH**)
GS2 is Parameter Group Status (1 = **CORRUPT**, 0 = **VALID**)

PS1 and PS2 come from the associated status parameter, if one exists. All ASCB parameter types (except status parameters themselves) can have an associated status parameter, and it is the decoding of this associated status parameter that sets PS1 and PS2. The associated status parameter is actually a separate parameter on the bus. For example, StaticPressure may have an associated status parameter with the name staticPressureStatus, both of which can be added to FlightLine or the DataTap-10 TIS configuration file. If a parameter does not have an associated status parameter, then PS1 and PS2 will always be zero. If a parameter does have an associated status parameter, then the type of the status parameter will always be one of "flag status", "numeric status", or "discrete status". Note that all status parameters are 1 or 2 bits in length.

Below is an example of a parameter and its associated status parameter as shown from a Telnet client connected to the DataTap-10. Note that staticPressureStatus decodes to a value of 3, which is indicated in the Status (Sts) column for StaticPressure and is also the value for the lower 2 bits in the status byte.

```
ASCB Frame Number: 00007FCE                  Channels Decoded:    2
ASCB Input Status: OK                        Registry Displayed: Consume
System Time:       001:00:06:53.873135       Status Error Cnt:    0

Chan  SZ  I   Name                             Type    Value        HEX        Sts
-------------------------------------------------------------------------------------
0    : 1   1, staticPressure                   Float: 0.000                    S:03
1    : 1   1, staticPressureStatus             Stat:  3            00000003   S:00
```

For a Numeric Status type, the meaning of PS1 and PS2 are:

nssmFailureWarning=0,
nssmNoComputedData=1,
nssmFunctionalTest=2,
nssmNormalOperation=3

For a Discrete Status type, the meaning of PS1 and PS2 are:

dssmNormalOperation=0,
dssmNoComputedData=1,
dssmFunctionalTest=2,
dssmFailureWarning=3

For a Flag Status type, the meaning of PS1 and PS2 are:

flagInvalid=0,
flagValid=1

Note that as described in the Honeywell PDD documentation, all status types are first converted to Discrete Status types prior to transmission onto the ASCB bus. They are then converted back to their original type during the decoding process.

9.2. Decoded Parameter UDP Output Datagrams (DIANE 62 Format)

Decoded Parameter UDP Output datagrams in DIANE62 format have additional encapsulation around the datagram described in section 9.1. Each datagram is broken up into 256 “channels” of data and each decoded (partial) datagram can be sent at different rates as defined in the system configuration file. Figure 13 below shows the encapsulation and is one example of how the different output rates can be used.

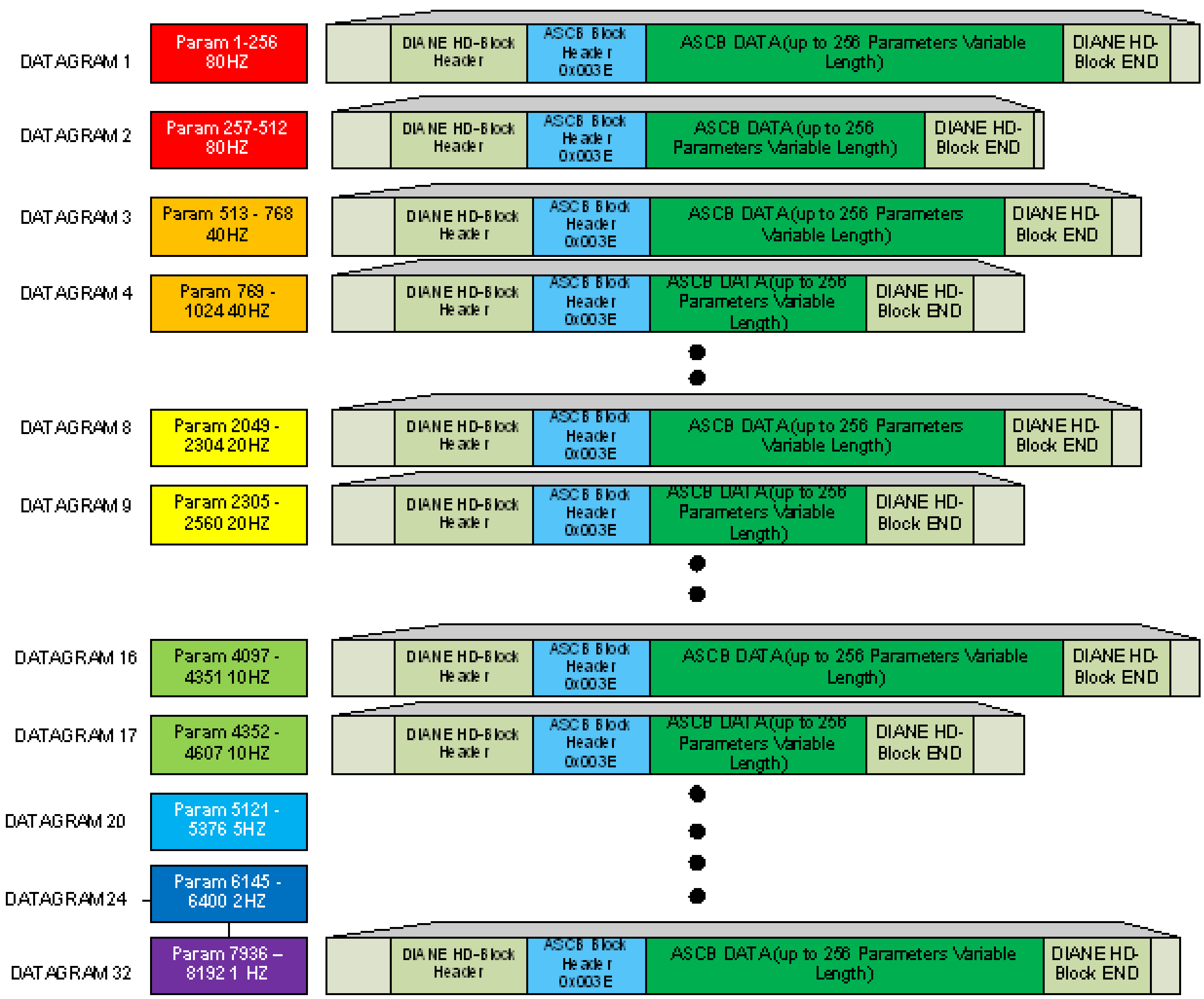


Figure 13 DIANE62 Data Output Example

9.2.1. DIANE Message Code 62 (Partial Send) Structure Definition

```
//-----------------------------------------------
// Dassault DIANE CODE 62 (0x003E) block format
//-----------------------------------------------

typedef struct
{

  //--------------------------------------------------------------
  //  High Density Block Header  ( 7 x 16-bits words)
  //--------------------------------------------------------------

   __u16 HD_block_sender_id;  // [15:11]DevNum, [10:8]SwVer, [5:0]0x03
   __u16 HD_block_size;       // Size of entire block (in 16-bit words)
   __u16 DATE_MS48;           // MS  16-bits of 48-bit 1us counter
   __u16 DATE_Mid48;          // Mid 16-bits of 48-bit 1us counter
   __u16 DATE_LS48;           // LS  16-bits of 48-bit 1us counter
   __u16 HD_block_status;     // DataTap-10 status
   __u16 HD_block_number;     // LS 16-bits of ASCB Frame Number

  //-------------------------------------------------------------
  //   ASCB Data Block Header
  //-------------------------------------------------------------

   __u16 msg_type;            // Message type (62)
   __u16 ASCB_msg_size;        // Message Size - 4 in bytes
   __u16 ASCB_status;          // ASCB status
   __u16 high_time;            // 14-bits 10s Resolution Bit 15 Even Parity
   __u16 low_time;             // 14-bits 1ms Resolution 0-9999, Bit 15 Even Parity
   __u16 micro_time;           // 14-bits 1us Resolution 0-999 Other bits 0.
   __u16 frag_num;             // Fragment number
   __u32 abs_seq_num;          // Absolute Sequence Number (ASCB Frame Number)

  //-------------------------------------------------------------
  //   ASCB Data Block
  //-------------------------------------------------------------

   __u32 ASCB_payload_length; // Length in bytes of ASCB Data Block
   __u32 ASCB_frame_number;   // ASCB Frame Number
   __u32 CHANNEL[256 max];    // Decoded ASCB Data
   __u8  STATUS[256 max];     // Decoded ASCB Status

  //--------------------------------------------------------------
  //  High Density Block Trailer
  //--------------------------------------------------------------

   __u16 DIANE_0xDEAD         // DIANE trailer (always starts on 16-bit boundary)


} DIANE_CODE_62_DATAGRAM;
```

Note that all fields in the High Density Block Header, the ASCB Data Block Header, and the High Density Block Trailer are sent in network byte order. As such, the functions ntohl and ntohs should be used when reading these fields. The ASCB Data Block is in Little Endian format, and therefore can be read directly when on a Little Endian host machine.

9.2.2. DIANE Message Code 62 Hard Coded Data Fields

The following structure fields are hard coded in each DIANE Message Code 62 UDP datagram. The values do not change from datagram to datagram.

Structure Field Name	Value	Comments
.msg_type	0062 (0x003E)	DIANE Message Code
.DIANE_0xDEAD	0xDEAD	Per Dassault Instructions 2008

9.2.3. DIANE Message Code 62 0x003E Variable Data Fields

The following structure fields are dynamically assigned based on the following descriptions.

9.2.4. DIANE ASCB-D UDP Payload Fields

The following fields are ASCB-D payload and DIANE block number related.

9.2.4.1.Field .HD_block_sender_id

This field provides information regarding the DataTap-10 and has the following meaning:

Bits [15:11] - DataTap-10 device number taken from the system configuration file.
Bits [10:8] - DataTap-10 software version.
Bits [7:6] - Scrambling key. Currently set to 0.
Bits [5:0] - Always 0x03, indicating DataTap-10 device.

9.2.4.2.Field .HD_block_size

This field indicates the size of the entire datagram, including the DIANE trailer, in 16-bit words. Because the number of channels per datagram is configurable, this field will vary depending on the datagram number.

9.2.4.3.Field .DATE_xx48

These fields represent the value of an incrementing 48-bit microsecond counter. The counter is derived from the PTPv2 or IRIG time depending on the specified synchronization mode using Dassault-supplied functions. The value represents the time at which the datagram was sent out of the DataTap-10.

9.2.4.4.Field .HD_block_status

This field provides status information regarding the DataTap-10. The bits are defined as follows in [15:0] format:

Bit	Definition
0	Raw/Full Recording is enabled.
1	Dec/Partial Recording is enabled.
2	Recording buffer overflow occurred. This indicates the microSD card write performance is too slow.
3	MicroSD recording error occurred. This indicates the card reported an error or that it may be full.
4	MicroSD card is full.
5	MicroSD warning is asserted, indicating recording time remaining is less than value specified in the system configuration file.
6	Error found while parsing system configuration file or system configuration file is missing.
7	Timing source is PTPv2.
8	Selected timing source is synchronized to its master.
9	NIC Registry is invalid.
10	ESCAPE Registry is invalid.

9.2.4.5.Field .HD_block_number

This field contains the lower 16-bits of the current ASCB frame number.

9.2.4.6.Field .ASCB_msg_size

This field contains the size of the ASCB data block minus 4 bytes. The ASCB data block consists of all bytes starting with the DIANE_type field and ending with the last configured STATUS byte. Because the number of channels in each datagram is configurable, the ASCB data block may end on any byte boundary.

9.2.4.7.Field .ASCB_status

This field provides status information regarding the DataTap-10's connection to the ASCB bus. The bits are defined as follows in [15:0] format:

Bit	Definition
0	Configured ASCB mode is Enhanced-ASCB.
1	The DataTap-10 is synchronized to the ASCB bus. This implies the configured ASCB mode

	matches the actual ASCB mode because synchronization is not possible if they are different due to the different bus encoding methods. Note that only one ASCB bus needs to be connected in order for synchronization to occur as long as a timing NIC exists on that bus.
2	Data is being received on ASCB 0.
3	Data is being received on ASCB 1.
4	ASCB connections are okay. This is determined by ensuring the correct NIC IDs are being received on each ASCB bus. If the status shows data is being received on both ASCB busses but this bit is not asserted, the ASCB bus connections are likely swapped. Note that both ASCB connections are required in order for this bit to be asserted.

9.2.4.8. Field .xxx_time

These fields represent the time at which the most recent parameter in the datagram was captured off of the ASCB bus. The time is derived from the PTPv2 or IRIG time depending upon the timing source specified in the system configuration file. The current PTPv2 or IRIG time is latched and saved with a parameter group as a complete group is captured off of the ASCB bus.

While parameters are being decoded each frame as directed by the tis.cfg file, the timestamps are compared for every parameter within a datagram with a valid status. Once decoding is complete, the most recent timestamp for each datagram is saved into 1 of 32 registers depending on the datagram number. Each timestamp is then converted using the Dassault-supplied functions and placed into the appropriate datagram for sending out over Ethernet.

Parameters in the datagram which have an invalid status are NOT included in the most recent parameter comparison. As a result, if all parameters in a datagram have an invalid status, these timestamp fields will all be zero in the datagram.

9.2.4.9. Field .frag_num

This field represents the fragment number of the datagram. Valid values are 0 to 31. This can be used to determine which channels a datagram contains:

frag_num Value	Represented Channels
0	0 – 255
1	256 – 511
…	…
31	7936 – 8191

9.2.4.10. Field .abs_seq_num

This field is a 32-bit frame count representing the number of the current ASCB-D frame. This number increments at an 80 Hz Rate and does not roll over during a flight segment. (The counter takes 621.3 days to rollover at 80Hz.) This can be used to determine missing UDP frames

9.2.4.11. Field .CHANNEL[256 max]

This field is an array of up to 256 32-bit converted ASCB parameter values. Each 32-bits is one logical channel. Channel assignments are derived from the tis.cfg file created by an ICS-developed TIS configuration Tool. The TIS configuration tool allows a user to pick one parameter from the approximately 65000 possible parameters on the aircraft bus and assign it to a logical channel number.

9.2.4.12.Field .STATUS[256 max]

This field is an array of up to 256 8-bit ASCB parameter status values. Each logical channel has an associated status byte in this array. As such, there will be as many STATUS values as there are CHANNEL values. See section 9.1.3 for decoding of the status byte values.

9.2.5. DIANE Message Code 62 timestamp fields

There are multiple time fields filled in with time information in the Message Code 62 DIANE messages, all derived from the PTPv2 or IRIG time depending on the selected timing source. The actual software source used to fill in these fields is listed below and based on the time functions provided by Dassault in 2008.

DIANE Message 62 Source Code for time stamp data.

```
/*-----------------------------------*/
/* Get DIANE time info and format it */
/*-----------------------------------*/

gettimeofday(tv,0);
setTimeVal(tv);
Hpf = getHeurePoidsFaible();
HPF = getHeurePoidsFort();
HHR = getHeureHauteResolution();
Cpt = getCpt48Bits();


/*--------------------------------------------------------*/
/* MS 16-bits of 48-bit 1us counter in field .DATE_MS48  */
/* DIANE Header Word 3                                    */
/*--------------------------------------------------------*/

DIANE_send_data[datagram_number].DATE_ms48 = htons((__u16)((Cpt & 0x0000ffff00000000ll)
>> 32)) ;

/*---------------------------------------------------------*/
/* Mid 16-bits of 48-bit 1us counter in field .DATE_Mid48 */
```

```
/* DIANE Header Word 4                                      */
/*---------------------------------------------------------*/

DIANE_send_data[datagram_number].DATE_midS48 = htons((__u16)((Cpt &
0x00000000ffff0000ll) >> 16)) ; // Mid 16-bits of 48-bit 1us counter

/*--------------------------------------------------------*/
/* LS 16-bits of 48-bit 1us counter in field .DATE_LS48   */
/* DIANE Header Word 5                                    */
/*--------------------------------------------------------*/

DIANE_send_data[datagram_number].DATE_ls48 = htons((__u16)((Cpt &
0x000000000000ffffll))) ;          // LS 16-bits of 48-bit 1us counter

/*-----------------------------------------------------------------*/
/* 15-bits and 1 Parity of 10 Sec Resolution in field ].high_time */
/* DIANE Word 2                                                    */
/*-----------------------------------------------------------------*/

DIANE_send_data[datagram_number].high_time = htons((__u16)HPF) ;


/*-----------------------------------------------------------------*/
/* 15-bits and 1 Parity of 1 Ms Resolution in field ].low_time    */
/* DIANE Word 3                                                    */
/*-----------------------------------------------------------------*/

DIANE_send_data[datagram_number].low_time = htons((__u16)Hpf) ;


/*-----------------------------------------------------------------------*/
/*  0..999 uSec other bits 0. of 1 us Resolution in field ].micro_time  */
/* DIANE Word 4                                                          */
/*-----------------------------------------------------------------------*/

DIANE_send_data[datagram_number].micro_time = htons((__u16)(tv->tv_usec%1000)) ;
```

9.3. Raw ASCB Frame Buffer UDP Output Datagrams (TISNATIVE Format)

The Raw NIC Frame Buffer UDP datagrams sent from the DataTap-10 ASCB-D consist of a 512-byte header followed by two 12KB buffers of data. The 12KB buffers contain the output of the NIC PDD for each of the two ASCB-D busses. One large 25,088-byte datagram containing all data is sent each ASCB-D frame (every 12.5 milliseconds). Each large datagram is fragmented into smaller packets to meet the Maximum Transfer Unit constraint of the Ethernet link and will be reassembled by the IP stack residing on the host. These datagrams are sent out of the DataTap-10 to the UDP port specified in the system configuration file (system.cfg) with the UDP_HOST_PORT_RAW parameter. Sending of these datagrams can be disabled through the UDP_RAW_EN_DEF parameter in the system configuration file.

Along with these datagrams, a separate UDP frame sync datagram is sent out by the DataTap-10 at the start of each ASCB-D frame. This is a short datagram and is typically used in ASCB-D transmission applications for the DataTap-10. The frame sync datagrams contain the current ASCB frame number and allow a host PC to begin producing the ASCB-D transmission buffers. The UDP frame sync datagrams are sent out of the DataTap-10 to the UDP port plus 1 specified in the system configuration file (system.cfg) with the UDP_HOST_PORT_RAW parameter. Sending of these datagrams can be disabled through the UDP_RAW_EN_DEF parameter in the system configuration file.

9.3.1. Data Structure of the Raw NIC Frame Buffer UDP Datagrams

```
typedef struct
{
   uint32 frame_tick;            // ASCB Frame Number
   uint32 discretes;             // [31:4]=RFU, [3:0]=Discrete Inputs 4:1
   uint32 unused1;               // RFU
   uint32 unused2;               // RFU

   uint32 IRIG1;                 // IRIG Time 1
   uint32 IRIG2;                 // IRIG Time 2
   uint32 IRIG3;                 // IRIG Time 3
   uint32 IRIG4;                 // IRIG Time 4

   char   format_str[8];         // Block formatting string

   uint08 unused3[472];          // RFU

   uint08 rx2_data[0x3000];      // RX2 Raw NIC Frame Buffer Data
   uint08 rx1_data[0x3000];      // RX1 Raw NIC Frame Buffer Data

} record_block_t;
```

Where:

- discretes: This field indicates the value of the 4 discrete inputs during the current ASCB frame, allowing them to be used for event marking when recording.

- The IRIG fields can be decoded using the following table:

Field	[31:16]	[15:12]	[11:8]	[7:4]	[3:0]
IRIG1	0	10^0 mS	10^2 uS	10^1 uS	10^0 uS
IRIG2	0	10^1 Seconds	10^0 Seconds	10^2 mS	10^1 mS
IRIG3	0	10^1 Hours	10^0 Hours	10^1 Minutes	10^0 Minutes
IRIG4	0	0	10^2 Days	10^1 Days	10^0 Days

- format_str : The format_str is used to determine the size of the datagram. If the string reads "eASCB   ", the datagram represents that of Enhanced-ASCB and each of the frame buffers will be 12288 bytes in length. If the string reads anything else (typically zeros), the frame buffers will each only be 12032 bytes in length. For the DataTap-10, the string always represents that of Enhanced-ASCB. Note that the larger datagram size is backwards-compatible with ASCB-D.

9.4. Raw ASCB Frame Buffer UDP Output Datagrams (DIANE 61 Format)

Raw ASCB-D frame buffers are 24Kbyte blocks of data and encapsulated in a single DIANE F4 HD-block. The block is sent as one large UDP datagram that is automatically fragmented by the DataTap-10 and reassembled by the receiving TCP/IP stack. These HD-blocks are shipped on a single UDP port number specified in the system configuration file. An incrementing frame sequence number is included in the ASCB-D RAW data payload to allow missing UDP data detection.

Note that all fields in the High Density Block Header, the ASCB Data Block Header, and the High Density Block Trailer are sent in network byte order. As such, the functions ntohl and ntohs should be used when reading these fields. The ASCB Data Block is in Little Endian format, and therefore can be read directly when on a Little Endian host machine.

The format of a RAW ASCB-D UDP Datagram is specified below:

9.4.1. DIANE Message Code 61 (Full Send) Structure Definition

```
//-----------------------------------------------
// Dassault DIANE CODE 61 (0x003D) block format
//-----------------------------------------------

typedef struct
{
   //----------------------------------------------------------------
   //  High Density Block Header  ( 7 x 16-bits words)
   //----------------------------------------------------------------

    __u16 HD_block_sender_id;  // [15:11]DevNum, [10:8]SwVer, [5:0]0x03
    __u16 HD_block_size;       // Size of entire block (in 16-bit words)
    __u16 DATE_MS48;           // MS  16-bits of 48-bit 1us counter
    __u16 DATE_Mid48;          // Mid 16-bits of 48-bit 1us counter
    __u16 DATE_LS48;           // LS  16-bits of 48-bit 1us counter
    __u16 HD_block_status;     // DataTap-10 status
    __u16 HD_block_number;     // LS 16-bits of ASCB Frame Number

   //---------------------------------------------------------------
   //   ASCB Data Block Header
   //---------------------------------------------------------------

    __u16 msg_type;            // Message type (61)
    __u16 ASCB_msg_size;       // Message Size - 4 in bytes
    __u16 ASCB_status;         // ASCB status
    __u16 high_time;           // 14-bits 10s Resolution Bit 15 Even Parity
    __u16 low_time;            // 14-bits 1ms Resolution 0-9999, Bit 15 Even Parity
    __u16 micro_time;          // 14-bits 1us Resolution 0-999 Other bits 0.
    __u16 frag_num;            // Fragment number
    __u32 abs_seq_num;         // Absolute Sequence Number (ASCB Frame Number)

   //---------------------------------------------------------------
   //   ASCB Data Block
   //---------------------------------------------------------------

    record_block_t RAW_FRAME_BUFFER;    // ASCB Data Block

   //----------------------------------------------------------------
   //  High Density Block Trailer
   //----------------------------------------------------------------

    __u16 DIANE_0xDEAD         // DIANE trailer

 } DIANE_RAW_ASCB;
```

9.4.2. DIANE Message Code 61 Hard-Coded Data Fields

The following structure fields are hard-coded in each DIANE Message Code 61 UDP datagram. The values do not change from datagram to datagram.

Structure Field Name	Value	Comments
.HD_block_size	0x3111	
.msg_type	0061 (0x003D)	DIANE Message Code
ASCB_msg_size	0x6207	
.frag_num	0x0000	Per DEV_101875_v3
.DIANE_0xDEAD	0xDEAD	Per Dassault Instructions 2008

9.4.3. DIANE Message Code 61 0x003D Variable Data Fields

The following structure fields are dynamically assigned based on the following descriptions.

9.4.4. DIANE ASCB-D UDP Payload Fields

The following fields are ASCB-D payload and DIANE block number related.

9.4.4.1. Field .HD_block_sender_id

This field provides information regarding the DataTap-10 and has the following meaning:

Bits [15:11] - DataTap-10 device number taken from the system configuration file.
Bits [10:8] - DataTap-10 software version.
Bits [7:6] - Scrambling key. Currently set to 0.
Bits [5:0] - Always 0x03, indicating DataTap-10 device.

9.4.4.2. Field .DATE_xx48

These fields represent the value of an incrementing 48-bit microsecond counter. The counter is derived from the PTPv2 or IRIG time depending on the specified synchronization mode using Dassault-supplied functions. The value represents the time at which the datagram was sent out of the DataTap-10.

9.4.4.3. Field .HD_block_status

This field provides status information regarding the DataTap-10. See the table in section 9.2.4.4 for the bit definitions.

9.4.4.4.Field .HD_block_number

This field contains the lower 16-bits of the current ASCB frame number.

9.4.4.5.Field .ASCB_status

This field provides status information regarding the DataTap-10's connection to the ASCB bus. See the table in section 9.2.4.7 for the bit definitions

9.4.4.6.Field .xxx_time

These fields represent the time latched at the start of the current frame. The timestamp is then converted using the Dassault-supplied functions and placed into the datagram for sending out over Ethernet.

9.4.4.7. Field .abs_seq_num

This field is a 32-bit frame count representing the number of the current ASCB-D frame. This number increments at an 80 Hz Rate and does not roll over during a flight segment. (The counter takes 621.3 days to rollover at 80Hz.) This can be used to determine missing UDP frames

9.4.4.8. Field .RAW_FRAME_BUFFER

This field represents a full record_block_t structure as defined in section 9.3.1.

9.5. Parameter UDP Input Datagrams

The Parameter UDP Input Datagrams allow a user to change parameter values transmitted by the DataTap-10 onto the ASCB bus. Similar to the Decoded Parameter UDP Output Datagrams (section 9.1), each datagram contains 256 channels worth of parameter values and status. Datagrams are sent to different UDP destination ports such that the DataTap-10 knows which datagram contains which channels. Different base UDP destination ports are provided for each ASCB bus in the system configuration file. Up to 32 input datagrams (8192 channels) between the two ASCB busses are supported, with no limitations on how the datagrams are split up between the busses. Datagrams to the same UDP destination port (i.e. same channels) can occur as often as every ASCB frame, which are 12.5 milliseconds apart. The host may listen for the UDP Frame Sync Datagram(s) sent out by the DataTap-10 at the start of each ASCB frame to synchronize sending of the datagrams, but this is not a requirement. The Parameter UDP Input Datagrams may also be sent asynchronously.

The UDP destination port base for each ASCB bus is configurable through the system configuration file (system.cfg) loaded on the DataTap-10's microSD card. Assuming the field UDP_LSTN_PORT_SIU0 is set to 51100, the incoming datagrams must look like the following:

Datagram 0 contains channels [0..255] and received by the DataTap-10 on port 51100.
Datagram 1 contains channels [256..511] and received by the DataTap-10 on port 51101.
…
Datagram 31 contains channels [7935..8191] and received by the DataTap-10 on port 51131.

9.5.1. Data Structure

```
typedef struct
{
   uint32 LENGTH;
   uint32 ASCBD_Frame;
   uint32 CHANNEL[256];
   uint08 STATUS[256];
   uint32 CRC32;    // Optional

} IN_DATAGRAM;
```

Where:

LENGTH	32-Bit value indicating the length of the UDP payload which includes the CHANNEL, STATUS, LENGTH, ASCBD_Frame, and optional CRC32 field.
ASCBD_Frame	Unused. This field is unused by the DataTap-10 and can be set to 0.
CHANNEL	32-Bit parameter values for each channel.
STATUS	8-Bit status values for each channel. This field may be used to corrupt the integrity check and/or freshness fields for a given parameter.
CRC32	Unused. This field is unused by the DataTap-10 and may be omitted. It can optionally be included for legacy purposes.

9.5.2. Status Byte Format

Each channel has an associated status byte as shown above. These are 8-bit values with 2-bits actually having any meaning. Setting GS1 or GS2 to a ‘1’ will cause the DataTap-10 to corrupt the integrity check field or freshness field, respectively. Both can be set. Note that because the integrity check and freshness fields are associated with a complete group of parameters, all parameters in the group will show the same error.

```
Bit                              Bit
7    6    5    4    3    2    1    0
---------------------------------------------
X    X    X    X   GS1  GS2   X    X
---------------------------------------------
```

Where:

GS1 is Parameter Group Status (1 = **STALE**, 0 = **FRESH**)
GS2 is Parameter Group Status (1 = **CORRUPT**, 0 = **VALID**)
X = Don't Care

9.6. DataTap-10 Frame Sync UDP Output Datagrams

At the start of each frame, the DataTap-10 will conditionally send out one or more UDP Frame Sync datagrams, mainly to communicate the current ASCB frame number to any listening applications. The number of datagrams to be sent out and the base UDP destination port are configurable through the system configuration file. When multiple datagrams are configured to be sent out, each will have an incrementing UDP destination port starting from the specified base port. The Frame Sync datagrams are typically used in DataTap-10 transmission applications where a host needs as early a signal as possible to begin producing the data to be transmitted. Having the ability to generate multiple frame syncs is useful when separate applications are used to produce the data for each ASCB bus.

9.6.1. Data Structure of the Frame Sync UDP Datagrams

```
typedef struct
{
   uint32 frame_num;         // ASCB Frame Number
   uint32 ascb0_nic_ids;     // ASCB 0 DT-10 Acting NIC IDs
   uint32 ascb1_nic_ids;     // ASCB 1 DT-10 Acting NIC IDs
   uint08 active_nics[4];    // NIC IDs received last frame
   uint08 nic_synced_to;     // NIC ID DT-10 synced to last frame
   uint08 rx_switch;         // DT-10 ASCB-D Rx Switch Settings

} udp_sync_pkt_t;
```

Where:

The "rx_switch" field indicates whether the DT-10 is receiving on the primary or backup bus for both ASCB 0 and ASCB 1. This field has the following definitions:

```
// ASCB 0 Rx Switch Setting (1=Primary, 0=Backup)
#define C_UDP_SYNC_RXSW0_PRIM     0x00000001

// ASCB 1 Rx Switch Setting (1=Primary, 0=Backup)
#define C_UDP_SYNC_RXSW1_PRIM     0x00000002
```

The "ascb0_nic_ids" field indicates the NIC ID(s) the DT-10 is acting as for ASCB Bus 0. This field has the following definitions:

```
0x00000001 – NIC ID 1
0x00000002 – NIC ID 2
0x00000004 – NIC ID 3
      …
0x40000000 – NIC ID 31
```

The “ascb1_nic_ids” field indicates the NIC ID(s) the DT-10 is acting as for ASCB Bus 1. This field has the following definitions:

```
0x00000001 – NIC ID 33
0x00000002 – NIC ID 34
0x00000004 – NIC ID 35
      …
0x40000000 – NIC ID 63
```

9.7. DataTap-10 Status UDP Output Datagrams

The Status UDP datagrams sent from the DataTap-10 ASCB-D contain system status and configuration information. These datagrams are sent out once every ASCB frame (i.e. every 12.5 milliseconds) and will begin as soon as the unit is booted up and initialized. A status datagram is also sent out in response to receiving a UDP Command Input datagram, with the command echoed in the command and cmd_data fields. All status datagrams are sent to the UDP port specified in the system configuration file (system.cfg) with the UDP_HOST_PORT_STS parameter. The periodic sending of these datagrams can be disabled through the UDP_STS_EN_DEF parameter in the system configuration file.

9.7.1. Data Structure of the Status UDP Datagrams

```
typedef struct
{
   uint32 tbl_ver;              // Structure/Table Version
   uint32 status;               // Status Field
   uint32 frame_num;            // Current ASCB Frame Number
   uint32 IRIG1;                // IRIG 1 Field
   uint32 IRIG2;                // IRIG 2 Field
   uint32 IRIG3;                // IRIG 3 Field
   uint32 IRIG4;                // IRIG 4 Field
   uint32 cpu0_load;            // CPU 0 Load Percentage
   uint32 board_temp;           // Temp Sensor Reading
   uint32 ascb0_pkt_cnt;        // ASCB 0 Packet Count
   uint32 ascb0_crc_err;        // ASCB 0 CRC Error Count
   uint32 ascb0_runt_err;       // ASCB 0 Runt Packet Count
   uint32 ascb0_sfd_err;        // ASCB 0 SFD Error Count
   uint32 ascb1_pkt_cnt;        // ASCB 1 Packet Count
   uint32 ascb1_crc_err;        // ASCB 1 CRC Error Count
   uint32 ascb1_runt_err;       // ASCB 1 Runt Packet Count
   uint32 ascb1_sfd_err;        // ASCB 1 SFD Error Count
   uint32 gpio_inputs;          // GPIO Input State
   uint32 gpio_outputs;         // GPIO Output State
   uint16 adc_val[4];           // ADC Readings
   uint08 timing_nics[4];       // Timing NIC IDs Last Frame
   uint32 nic_synced_to;        // NIC Synced To Last Frame
   uint32 ascb0_asgn_nids;      // ASCB 0 Assigned NIC IDs
   uint32 ascb1_asgn_nids;      // ASCB 1 Assigned NIC IDs
   uint08 nic_reg_hdr[144];     // NIC Registry Header String
```

```
   uint08 esc_reg_hdr[176];   // ESCAPE Registry Header String
   uint32 fpga_version;       // FPGA version
   uint08 cpu0_sw_date[16];   // CPU 0 SW Compile Date String
   uint08 cpu0_sw_time[16];   // CPU 0 SW Compile Time String
   uint08 cpu1_sw_date[16];   // CPU 1 SW Compile Date String
   uint08 cpu1_sw_time[16];   // CPU 1 SW Compile Time String
   uint08 cpu2_sw_date[16];   // CPU 2 SW Compile Date String
   uint08 cpu2_sw_time[16];   // CPU 2 SW Compile Time String
   uint32 command;            // Echo of Command Received
   uint08 cmd_data[96];       // Echo of Command Data Received
   uint32 ascb0_rcvd_nids[2]; // ASCB 0 Received NIC IDs
   uint32 ascb1_rcvd_nids[2]; // ASBB 1 Received NIC IDs
   uint32 rec_file_size;      // File size of current recording
   uint32 rec_mem_rem_mb;     // Media memory remaining in MB
   uint32 rec_mem_size_mb;    // Media memory size in MB
   uint08 rec_fn_base[32];    // Base filename of current recording
   uint08 rec_fn_ext[4];      // Extension of current recording
   uint32 dt10_sw_ver_num;    // DataTap-10 Software Version Number
   uint32 dt10_feature_mask;  // DataTap-10 Feature Mask
   uint08 rec_fn_base2[32];   // Continuation of base rec filename
   uint32 sts_err_cnt;        // Status Error Count
   uint32 rec_buf_size;       // Recording buffer size in bytes
   uint32 rec_buf_ovf;        // Recording buffer overflows
   uint32 rec_buf_occ;        // Recording buffer current occupancy
   uint32 rec_buf_max_occ;    // Recording buffer maximum occupancy
   int32  rec_err_code;       // Recording error code
   uint32 rec_info_sel;       // Recording information select
   uint32 dec_exceed_cnt;     // Parameter decode frame exceed count
   uint32 enc_exceed_cnt;     // Parameter encode frame exceed count
   uint32 dec_engine_load;    // Parameter Decode Engine load
   uint32 enc_engine_load;    // Parameter Encode Engine load
   uint32 ext_status;         // Extended Status Field

} udp_sts_pkt_t;
```

The following table indicates the structure version each field was added as well as the version of software:

Field	Structure Version Added	Software Version
tbl_ver	2	Nov 21, 2011
status	1	-
frame_num	1	-
IRIG1	1	-
IRIG2	1	-
IRIG3	1	-
IRIG4	1	-
cpu0_load	1	-
board_temp	1	-
ascb0_pkt_cnt	1	-
ascb0_crc_err	1	-
ascb0_runt_err	1	-
ascb0_sfd_err	1	-
ascb1_pkt_cnt	1	-

Field	Structure Version Added	Software Version
ascb1_crc_err	1	-
ascb1_runt_err	1	-
ascb1_sfd_err	1	-
gpio_inputs	2	Nov 21, 2011
gpio_outputs	2	Nov 21, 2011
adc_val	2	Nov 21, 2011
timing_nics	2	Nov 21, 2011
nic_synced_to	2	Nov 21, 2011
ascb0_asgn_nids	2	Nov 21, 2011
ascb1_asgn_nids	2	Nov 21, 2011
nic_reg_hdr	2	Nov 21, 2011
esc_reg_hdr	2	Nov 21, 2011
fpga_version	2	Nov 21, 2011
cpu0_sw_date	2	Nov 21, 2011
cpu0_sw_time	2	Nov 21, 2011
cpu1_sw_date	2	Nov 21, 2011
cpu1_sw_time	2	Nov 21, 2011
cpu2_sw_date	2	Nov 21, 2011
cpu2_sw_time	2	Nov 21, 2011
command	2	Nov 21, 2011
cmd_data	2	Nov 21, 2011
ascb0_rcvd_nids	2	Nov 21, 2011
ascb1_rcvd_nids	2	Nov 21, 2011
rec_file_size	5	Oct 30, 2012
rec_mem_rem_mb	5	Oct 30, 2012
rec_mem_size_mb	5	Oct 30, 2012
rec_fn_base	5	Oct 30, 2012
rec_fn_ext	5	Oct 30, 2012
dt10_sw_ver_num	6	August 6, 2014
dt10_feature_mask	6	August 6, 2014
rec_fn_base2	7	Sept 11, 2014
sts_err_cnt	7	Sept 11, 2014
rec_buf_size	9	Sept 15, 2015
rec_buf_ovf	9	Sept 15, 2015
rec_buf_occ	9	Sept 15, 2015
rec_buf_max_occ	9	Sept 15, 2015
rec_err_code	9	Sept 15, 2015
rec_info_sel	10	Feb 4, 2016
dec_exceed_cnt	10	Feb 4, 2016
enc_exceed_cnt	10	Feb 4, 2016
dec_engine_load	10	Feb 4, 2016
enc_engine_load	10	Feb 4, 2016

Field	Structure Version Added	Software Version
ext_status	11	July 28, 2016

9.7.1.1.Field .tbl_ver

This field indicates the structure version used by the datagram. The latest structure version is 10.

9.7.1.2.Field .status

The “status” field can have the following bits set with their meaning in [31:0] format:

Status Bit	Mask	Meaning
0	0x00000001	Ethernet is linked in 10 Mbs mode
1	0x00000002	Ethernet is linked in 100 Mbs mode
2	0x00000004	Ethernet is linked in 1000 Mbs mode
3	0x00000008	ARINC-717 recording is active.
4	0x00000010	Raw ASCB recording is active.
5	0x00000020	Decoded ASCB recording is active.
6	0x00000040	ARINC-429 recording is active.
7	0x00000080	DataTap-10 is in playback mode.
8	0x00000100	DataTap-10 power is from USB
9	0x00000200	DataTap-10 power is from 9V-36V input
10	0x00000400	DataTap-10 has a 2MB local RAM.
11	0x00000800	An ASCB recording is pending.
12	0x00001000	Available recording space is below threshold.
13	0x00002000	Recording media is full.
14	0x00004000	A recording error has occurred.
15	0x00008000	The recording media initialized correctly.
16	0x00010000	ASCB 0 receive switch is set to 1=primary, 0=backup
17	0x00020000	ASCB 1 receive switch is set to 1=primary, 0=backup
18	0x00040000	ASCB playback over UDP (Rx Raw) is active.
19	0x00080000	ASCB playback over ASCB (Tx Raw) is active.
20	0x00100000	Recording and playback is configured to use the Compact Flash rather than the microSD card. This is set in the system configuration file.
21	0x00200000	Targeted media for FTP transfers is Compact Flash rather than microSD. This is changed dynamically using the FTP server.
22	0x00400000	Compact Flash card is present.
23	0x00800000	Playback is currently seeking to desired start time.
24	0x01000000	ASCB mode is 1=Enhanced-ASCB, 0=ASCB-D
25	0x02000000	Timing source is locked to master.
26	0x04000000	Time source is 1= PTP (IEEE1588v2), 0=IRIG. This is

		set in the system configuration file.
27	0x08000000	DataTap-10 is in ASCB-A/B/C mode. If set, bit 24 should be ignored (ASCB mode) and the .ext_status field should be used to determine version A, B, or C.
28	0x10000000	Hardware transmit PDD is enabled.
29	0x20000000	Circular record mode is enabled.
30	0x40000000	Record append mode is enabled.
31	0x80000000	DataTap-10 has a 4MB local RAM.

9.7.1.3.Field .frame_num

This field indicates the current ASCB frame number.

9.7.1.4.Field .IRIGx

The IRIG fields can be decoded using the following table:

Field	[31:16]	[15:12]	[11:8]	[7:4]	[3:0]
IRIG1	0	10^0 mS	10^2 uS	10^1 uS	10^0 uS
IRIG2	0	10^1 Seconds	10^0 Seconds	10^2 mS	10^1 mS
IRIG3	0	10^1 Hours	10^0 Hours	10^1 Minutes	10^0 Minutes
IRIG4	0	0	10^2 Days	10^1 Days	10^0 Days

9.7.1.5.Field .cpu0_load

Indicates the current loading of CPU 0 expressed as a percentage of maximum load.

9.7.1.6. Field .board_temp

This field indicates the current value of the on-board temperature sensor of the DataTap-10. The temperature reading is expressed in 10-bit twos complement format, but must first be shifted to the right by 5 bits and masked with 0x3FF due to how the value is read out of the sensor. The value can then be decoded as follows:

Temperature	Digital Output DB9 . . . DB0
–128°C	10 0000 0000
–125°C	10 0000 1100
–100°C	10 0111 0000
–75°C	10 1101 0100
–50°C	11 0011 1000
–25°C	11 1001 1100
–0.25°C	11 1111 1111
0°C	00 0000 0000
+0.25°C	00 0000 0001
+10°C	00 0010 1000
+25°C	00 0110 0100
+50°C	00 1100 1000
+75°C	01 0010 1100
+100°C	01 1001 0000
+125°C	01 1111 0100
+127°C	01 1111 1100

9.7.1.7. Field .ascb0_pkt_cnt

This field indicates the number of valid packets received on ASCB bus 0. Packets which are rejected due to error are not included in this count.

9.7.1.8.Field .ascb0_crc_err

This field indicates the number of packets received on ASCB bus 0 which contained a CRC error, indicating corruption of the packet.

9.7.1.9.Field .ascb0_runt_err

This field indicates the number of packets received on ASCB bus 0 which had a packet length less than 24 bytes, which is the minimum required for processing by the NIC PDD. Note that the CRC is not verified on these packets.

9.7.1.10.Field .ascb0_sfd_err

This field indicates the number of packets received on ASCB bus 0 which had a Start-of-Frame Delimiter (SFD) error. This error occurs when the DataTap-10 detects a preamble to indicate the start of a packet, but the SFD is not encountered before the bus goes idle again.

9.7.1.11.Field .ascb1_pkt_cnt

This field indicates the number of valid packets received on ASCB bus 1. Packets which are rejected due to error are not included in this count.

9.7.1.12.Field .ascb1_crc_err

This field indicates the number of packets received on ASCB bus 1 which contained a CRC error, indicating corruption of the packet.

9.7.1.13.Field .ascb1_runt_err

This field indicates the number of packets received on ASCB bus 1 which had a packet length less than 24 bytes, which is the minimum required for processing by the NIC PDD. Note that the CRC is not verified on these packets.

9.7.1.14.Field .ascb1_sfd_err

This field indicates the number of packets received on ASCB bus 1 which had a Start-of-Frame Delimiter (SFD) error. This error occurs when the DataTap-10 detects a preamble to indicate the start of a packet, but the SFD is not encountered before the bus goes idle again.

9.7.1.15.Field .gpio_inputs

This field indicates the state of the four discrete inputs of the DataTap-10. The inputs are accessible through the optional P4 I/O header (section). This field has the following meaning:

Bit	Definition
0x00000001	State of Discrete Input 1
0x00000002	State of Discrete Input 2
0x00000004	State of Discrete Input 3
0x00000008	State of Discrete Input 4

9.7.1.16.Field .gpio_outputs

This field indicates the state of the four discrete outputs of the DataTap-10. The outputs are accessible through the optional P4 I/O header (section 3.8). This field has the following meaning:

Bit	Definition
0x00000001	State of Discrete Output 1
0x00000002	State of Discrete Output 2
0x00000004	State of Discrete Output 3
0x00000008	State of Discrete Output 4

9.7.1.17.Field .adc_val

This field contains the latest reading from each of the four channels of the on-board 16-bit Analog-To-Digital Converter. The four analog inputs are accessible through the optional P4 I/O header (section 3.8). Each channel has a range of -10V to 10V, but can accept up to +/-16.5V without affecting an adjacent channel and can handle an absolute maximum of up to +/-50V. The channels are configured to clamp to +/-10V in the event of an overvoltage condition. Each 16-bit value can be decoded to a voltage as follows:

Input (V)	Data (hex)
10.00000	FFFF
0.00031	8001
0.00000	8000
−0.00031	7FFF
−10.00000	0000

9.7.1.18.Field .active_nics

This field lists the NIC IDs of all timing NICs which were received in the last ASCB frame. The field is only 4 bytes because only 4 timing NICs can exist in an ASCB system. Each NIC ID received will occupy one byte, with the remaining bytes containing 0x00. If only one NIC ID was received, then only the first byte will be occupied regardless of the actual NIC ID value. This field should not change under normal operation once the DataTap-10 is synchronized to the ASCB bus. Each byte can contain one of the following values, and will be listed in ascending order (if non-zero) with no duplicates:

Field	Definition
0x01	NIC ID 1 received last frame
0x02	NIC ID 2 received last frame
0x21	NIC ID 33 received last frame
0x22	NIC ID 34 received last frame
0x00	End of list

9.7.1.19.Field .nic_synced_to

This field indicates the NIC ID of the master timing NIC for the entire ASCB bus. All NICs in the system, including the DataTap-10, are synchronized to the same master timing NIC under normal operation. Note that the DataTap-10 can be configured to act as one, multiple, all, or none of the timing NICs. Only one can become the master timing NIC and this is determined by the ASCB bus synchronization protocol.

This field can have the following values:

Field	Definition
0x01	NIC ID 1 is the timing master
0x02	NIC ID 2 is the timing master

0x21	NIC ID 33 is the timing master
0x22	NIC ID 34 is the timing master
0x00	No master timing NIC present

9.7.1.20.Field .ascb0_asgn_nids

This field indicates the NIC IDs which are currently assigned to the DataTap-10 for ASCB bus 0. It is represented as a 32-bit bitmask, with each bit indicating a single NIC ID. The DataTap-10 can act as multiple NIC IDs for each bus; therefore, multiple bits may be set in the bitmask.

Each bit has the following meaning, and any combination of bits may be set:

Bit	Definition
0x00000001	NIC ID 1
0x00000002	NIC ID 2
…	…
0x40000000	NIC ID 31
0x80000000	Invalid (no NIC ID 32 exists)

9.7.1.21.Field .ascb1_asgn_nids

This field indicates the NIC IDs which are currently assigned to the DataTap-10 for ASCB bus 1. It is represented as a 32-bit bitmask, with each bit indicating a single NIC ID. The DataTap-10 can act as multiple NIC IDs for each bus; therefore, multiple bits may be set in the bitmask.

Each bit has the following meaning, and any combination of bits may be set:

Bit	Definition
0x00000001	NIC ID 33
0x00000002	NIC ID 34
…	…
0x40000000	NIC ID 63
0x80000000	Invalid (no NIC ID 64 exists)

9.7.1.22.Field .nic_reg_hdr

This field contains the 144 bytes of the NIC registry header currently programmed into the DataTap-10.

9.7.1.23.Field .esc_reg_hdr

This field contains the 176 bytes of the consume ESCAPE registry header currently programmed into the DataTap-10.

9.7.1.24.Field .fpga_version

This field contains the DataTap-10 FPGA revision. See section 0 for FPGA revision history.

9.7.1.25.Field .cpu0_sw_date

This field contains the compile date of the software for processor 0. The format of this field is a string which is copied directly from the __DATE__ string in the C programming language.

9.7.1.26.Field .cpu0_sw_time

This field contains the compile time of the software for processor 0. The format of this field is a string which is copied directly from the __TIME__ string in the C programming language.

9.7.1.27.Field .cpu1_sw_date

This field contains the compile date of the software for processor 1. The format of this field is a string which is copied directly from the __DATE__ string in the C programming language.

9.7.1.28.Field .cpu1_sw_time

This field contains the compile time of the software for processor 1. The format of this field is a string which is copied directly from the __TIME__ string in the C programming language.

9.7.1.29.Field .cpu2_sw_date

This field contains the compile date of the software for processor 2. The format of this field is a string which is copied directly from the __DATE__ string in the C programming language. Note that FPGA revisions later than or equal to 08301301 do not contain a third processor because its functionality was moved into hardware.

9.7.1.30.Field .cpu2_sw_time

This field contains the compile time of the software for processor 2. The format of this field is a string which is copied directly from the __TIME__ string in the C programming language. Note that FPGA revisions later than or equal to 08301301 do not contain a third processor because its functionality was moved into hardware.

9.7.1.31.Field .command

This field is used to acknowledge receipt of a valid Command Input datagram and echoes the command received. In all other cases, this field is zero.

9.7.1.32.Field .cmd_data

This field is used to acknowledge receipt of a valid Command Input datagram and echoes the data of the received command. In all other cases, this field is zero.

9.7.1.33.Field .ascb0_rcvd_nids

This 64-bit field indicates all NIC IDs which were received on ASCB bus 0 in the previous ASCB frame. This includes timing NICs as well as non-timing NICs. It is represented as a bitmask, with each bit indicating a single NIC ID. No bits in the upper 32-bit word should ever be set under normal operation because these indicate the NIC IDs belonging to the copilot bus. If any are set, however, it indicates the ASCB cables are likely swapped going into the DataTap-10 (i.e. pilot is connected to ASCB bus 1 and copilot is connected to ASCB bus 0). Note that not all NICs transmit every frame, so the value of this field may vary from frame to frame.

Each bit has the following meaning, and any combination of bits may be set:

Bit	Definition
0x00000000_00000001	NIC ID 1
0x00000000_00000002	NIC ID 2
…	…
0x00000000_40000000	NIC ID 31
0x00000000_80000000	Invalid (no NIC ID 32 exists)
0x00000001_00000000	NIC ID 33
0x00000002_00000000	NIC ID 34
…	…
0x40000000_00000000	NIC ID 63
0x80000000_00000000	Invalid (no NIC ID 64 exists)

9.7.1.34.Field .ascb1_rcvd_nids

This 64-bit field indicates all NIC IDs which were received on ASCB bus 1 in the previous ASCB frame. This includes timing NICs as well as non-timing NICs. It is represented as a bitmask, with each bit indicating a single NIC ID. No bits in the lower 32-bit word should ever be set under normal operation because these indicate the NIC IDs belonging to the pilot bus. If any are set, however, it indicates the ASCB cables are likely swapped going into the DataTap-10 (i.e. pilot is connected to ASCB bus 1 and copilot is connected to ASCB bus 0). Note that not all NICs transmit every frame, so the value of this field may vary from frame to frame.

Each bit has the following meaning, and any combination of bits may be set:

Bit	Definition
0x00000000_00000001	NIC ID 1
0x00000000_00000002	NIC ID 2
…	…

0x00000000_40000000	NIC ID 31
0x00000000_80000000	Invalid (no NIC ID 32 exists)
0x00000001_00000000	NIC ID 33
0x00000002_00000000	NIC ID 34
…	…
0x40000000_00000000	NIC ID 63
0x80000000_00000000	Invalid (no NIC ID 64 exists)

9.7.1.35.Field .rec_file_size

This field contains the size of the current recording file in bytes.

9.7.1.36.Field .rec_mem_size_mb

This field contains the total size of the selected media in megabytes.

9.7.1.37.Field .rec_fn_base

This field contains the first 32 bytes of the current recording base filename.

9.7.1.38.Field .rec_fn_ext

This field contains the file extension of the current recording filename.

9.7.1.39.Field .dt10_sw_ver_num

This field contains the DataTap-10 software version number. It is used in the DIANE headers and is also used by FlightLine.

9.7.1.40.Field .dt10_feature_mask

This field contains the features supported by the DataTap-10 and is used by FlightLine. It has the following definitions:

Bit	Definition
0x00000001	Playback over UDP (Rx Raw) is supported.
0x00000002	RFU
0x00000004	FTP passive mode is supported.
0x00000008	ARINC-717 recording is supported.
0x00000010	Playback over ASCB (Tx Raw) is supported.

0x00000020	Multi-stream recording is supported.
0x00000040	Host Transmit Playback mode is supported.
0x00000080	Switch to Live command is supported.
0x00000100	ASCB-A is supported.

9.7.1.41.Field .rec_fn_base2

This field contains the second 32 bytes of the current recording base filename.

9.7.1.42.Field .sts_err_cnt

This field indicates the total number of decoded parameter status errors and matches the value shown on page 1 of the Telnet screens. It can be cleared via Telnet or by using a Command Input datagram.

9.7.1.43.Field .rec_buf_size

This field indicates the size of the recording buffer in bytes. It can be used to determine how full the buffer is getting during recording. Note that it must be 256KB or greater when doing ASCB recording or an error will be generated when the recording is started.

9.7.1.44.Field .rec_buf_ovf

This field indicates the number of buffer overflow errors that have occurred during the current recording. An overflow condition can occur once per ASCB frame (12.5 msec) during raw recording, or up to 32 times per frame during decoded recording. If the current data cannot fit into the buffer, the data is dropped and the overflow counter is incremented. Note that the recording file is not corrupted due to an overflow condition, but ASCB frame(s) will be missing. Note that this field is cleared each time a new recording is started.

9.7.1.45.Field .rec_buf_occ

This field indicates the current occupancy of the recording buffer in bytes.

9.7.1.46.Field .rec_buf_max_occ

This field indicates the maximum occupancy of the recording buffer in bytes during the current recording. It is cleared each time a new recording is started and can also be cleared via Telnet or a Command Input datagram.

9.7.1.47.Field .rec_err_code

This field indicates the error code of the current recording and is cleared each time a new recording is started. Note that many of the values are used only by the DataTap-10 software and should never be seen externally. The error code can be any one of the following values:

Error Code	Meaning
0	No error.
-1	End of File. This can only occur during Circular Recording Mode when the end of a file being overwritten is reached prior to 2GB.
-2	File I/O Error (non-hardware).
-3	Card Error
-4	No card is present.
-5	Card is full.
-6	Max file size has been exceeded.
-7	Card is not initialized.
-8	No file handler is available.
-9	File does not exist.
-10	File is already open.
-11	No device has been set.
-12	Card is invalid.
-13	Hardware logic error.
-14	File already exists.
-15	Invalid file system type on card.
-100	No oldest file found. This can only occur when Circular Recording Mode is enabled.
-101	Recording buffer is not large enough to support ASCB recording. This should only occur when a 2MB RAM is present.
-102	Playback file is corrupt.
-103	Maximum session number (999) exceeded. This can only occur when Circular Recording Mode is enabled.

9.7.1.48.Field .rec_info_sel

This field indicates the current recording stream selected for all recording-related fields in the Status UDP Output Datagrams, which consist of:

rec_file_size
rec_fn_base
rec_fn_ext
rec_fn_base2
rec_buf_size
rec_buf_ovf
rec_buf_max_occ

This allows the information to be sent out for all active recording streams without having to create separate fields for each stream. For each status datagram sent, which occurs every 12.5 milliseconds, the select will alternate to the next active recording stream. If a recording stream is not active, it will be skipped. For example, if only ASCB and ARINC-717 are active, one status datagram will show the ASCB recording information (select = 0) while the next will show the ARINC-717 information (select = 2). The next datagram will then go back to ASCB. If only one stream is active, no alternating will occur.

This field can have the following values:

0 – ASCB Recording Info
1 – ARINC-429 Recording Info
2 – ARINC-717 Recording Info

9.7.1.49.Field .dec_exceed_cnt

This field indicates the number of times the parameter decode engine has exceeded its allotted 12.5 millisecond frame time. This can result in inadvertent status errors being present in the Decoded Parameter UDP Output Datagrams. If this number is incrementing, the number of parameters present in any of the TIS configuration files loaded onto the DataTap-10’s microSD card should be reduced until the number is no longer incrementing. This field can also be viewed in FlightLine v12.10 or later.

9.7.1.50.Field .enc_exceed_cnt

This field indicates the number of times the parameter encode engine (used only when HW_TX_PDD_ENABLE is set to TRUE) has exceeded its allotted 12.5 millisecond frame time. This can result in bad data or status errors being generated on the ASCB bus during transmission. If this number is incrementing, the number of parameters present in any of the TIS configuration files loaded onto the DataTap-10’s microSD card should be reduced until the number is no longer incrementing. This field can also be viewed in FlightLine v12.10 or later.

9.7.1.51.Field .dec_engine_load

This field indicates the load of the parameter decode engine as a percentage of a single ASCB frame, which is 12.5 milliseconds. As mentioned for the “dec_exceed_cnt” field, the number of parameters in any of the TIS configuration files loaded onto the DataTap-10’s microSD card should be reduced until the peak load is below 100%.

9.7.1.52.Field .enc_engine_load

This field indicates the load of the parameter encode engine as a percentage of a single ASCB frame, which is 12.5 milliseconds. As mentioned for the “enc_exceed_cnt” field, the number of parameters in any of the TIS configuration files loaded onto the DataTap-10’s microSD card should be reduced until the peak load is below 100%.

9.7.1.53.Field .ext_status

This field contains the extended status and can have the following bits set with their meaning in [31:0] format:

Status Bit	Mask	Meaning
0	0x00000001	Host Tx Playback Buffer is half-full.
1	0x00000002	DataTap-10 is in Host Tx Playback Mode.
4	0x00000010	DataTap-10 is configured for ASCB-A.
5	0x00000020	DataTap-10 is configured for ASCB-B.
6	0x00000040	DataTap-10 is configured for ASCB-C.

9.8. DataTap-10 UDP Command Input Datagrams

The UDP command datagrams allow dynamic configuration of the DataTap-10 from a host PC. The UDP port that the DataTap-10 will listen on for these datagrams is configurable through the system.cfg file located on the microSD card. If the command is valid and accepted by the DataTap-10, it will result in an echo of the command in the next Status Output datagram sent out. Note that these datagrams are in Little Endian format rather than network byte order. The following is the basic UDP command datagram structure:

9.8.1. UDP Command Datagram Structure

```
typedef struct
{
   uint32 command;
   uint32 cmd_data[24];

} udp_cmd_pkt_t;
```

The following table lists the available commands and their associated command number:

Command	Command Number
None	0
DataTap-10 Reset	1
Set Receive Switch	2
UDP Output Control	6
ARINC-429 Transmit Control	7
ARINC-429 Transmit Data	8
ARINC-429 Receive Control	9
Set NIC IDs	10
Set Discrete Outputs	11
ARINC-717 Receive Control	12
Set ASCB Mode	13
Set Multiple NIC IDs	14
ARINC-429 Record Control	15
ASCB Record Control	16
ASCB Playback Control	17

Clear Status Error Count	18
Clear Rec Buffer Max Occupancy	19
ARINC-717 Record Control	20
Multi-Stream Record Control	21
Clear Frame Exceeded Counts	22
Switch to Live Mode	23
Set PTP Time	24

The command data consists of a number of 32-bit parameters associated with the chosen command. The payload must be padded to 100 bytes (including the command field) or it will be rejected by the DataTap-10. Each command requires different parameters, and these are specified in detail in the sections below.

9.8.2. DataTap-10 Reset Command

The DataTap-10 Reset command initiates a reset of the DataTap-10 and is similar to a power-cycle. This command does not require any parameters in the cmd_data field, as shown:

Payload Offset	Parameter
0	Command (0x00)

9.8.3. Set Receive Switch Command

The Set Receive Switch command allows setting of whether the DataTap-10 receives data on the primary or backup bus for each of the two ASCB busses. The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x01)
4	Configuration Word

- Configuration Word – 32-bit word which has the following bit definitions:

```
#define C_UDP_CMD_SET_RXSW0_PRIM  0x01  // Set ASCB 0 to Primary
#define C_UDP_CMD_SET_RXSW0_BKUP  0x02  // Set ASCB 0 to Backup
#define C_UDP_CMD_SET_RXSW1_PRIM  0x04  // Set ASCB 1 to Primary
#define C_UDP_CMD_SET_RXSW1_BKUP  0x08  // Set ASCB 1 to Backup
```

If neither of the bits are set for a given ASCB bus, no action is taken and the previous setting remains.

9.8.4. UDP Output Control Command

The UDP Output Control command allows enabling and disabling of each UDP output type. The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x01)
4	Configuration Word

- Configuration Word – 32-bit word which can be any combination of the following bit definitions:

```
#define C_UDP_CTL_RAW_ON      0x01      // Turn Raw output on
#define C_UDP_CTL_RAW_OFF     0x02      // Turn Raw output off
#define C_UDP_CTL_DEC_ON      0x04      // Turn Dec output on
#define C_UDP_CTL_DEC_OFF     0x08      // Turn Dec output off
#define C_UDP_CTL_STS_ON      0x10      // Turn Sts output on
#define C_UDP_CTL_STS_OFF     0x20      // Turn Sts output off
#define C_UDP_CTL_SYNC_ON     0x40      // Turn Sync output on
#define C_UDP_CTL_SYNC_OFF    0x80      // Turn Sync output off
#define C_UDP_CTL_A429_ON     0x100     // Turn ARINC-429 output on
#define C_UDP_CTL_A429_OFF    0x200     // Turn ARINC-429 output off
#define C_UDP_CTL_A717_ON     0x400     // Turn ARINC-717 output on
#define C_UDP_CTL_A717_OFF    0x800     // Turn ARINC-717 output off
```

9.8.5. ARINC-429 Transmit Control Command

The ARINC-429 Transmit Control command allows enabling and disabling of ARINC-429 data along with its speed and settings. The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x07)
4	Bus Number
8	Configuration Word
12	Frame Usec
16	Num Tx Desc
20	Tx Bit Gap

- Bus Number – 32-bit word indicating which ARINC-429 transmit bus (0 or 1) this command applies to.

- Configuration Word – 32-bit word which has the following bit definitions:

```
#define C_UDP_ATX_CFG_EN          0x01  // Enable
#define C_UDP_ATX_CFG_DIS         0x02  // Disable
#define C_UDP_ATX_CFG_HSPD        0x04  // Set to high speed
#define C_UDP_ATX_CFG_LSPD        0x08  // Set to low speed
#define C_UDP_ATX_CFG_FRM_USEC    0x10  // Frame Usec valid
#define C_UDP_ATX_CFG_NUM_TXDD    0x20  // Num Tx Desc valid
#define C_UDP_ATX_CFG_BIT_GAP     0x40  // Tx Bit Gap valid
```

Where:

C_UDP_ATX_CFG_FRM_USEC indicates the Frame Usec field is valid,
C_UDP_ATX_CFG_NUM_TXDD indicates the Num Tx Desc field is valid, and
C_UDP_ATX_CFG_BIT_GAP indicates the Tx Bit Gap field is valid.

If any individual bits are not set, then no action will be taken with the associated field and the previous setting remains.

- Frame Usec – 32-bit word indicating the rate in microseconds at which the ARINC-429 transmit descriptor ring is polled. See section 6.9 for more details.

- Num Tx Desc – 32-bit word indicating the number of descriptors in the ARINC-429 transmit descriptor ring. See section 6.9 for more details.

- Tx Bit Gap – 32-bit word indicating the gap in transmit bit times between ARINC-429 transmissions. A typical value for this field is 4.

9.8.6. ARINC-429 Transmit Data Command

The ARINC-429 Transmit Data command allows filling up of the transmit descriptor ring for ARINC-429 transmissions. The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x08)
4	Bus Number
8	Descriptor Index
12	Descriptor Transmit Rate
16	Descriptor Transmit Data

- Bus Number – 32-bit word indicating which ARINC-429 transmit bus (0 or 1) this command applies to.

- Descriptor Index - 32-bit word indicating the transmit descriptor for which to update. Valid values are 0 to 0xFF.

- Descriptor Transmit Rate – 32-bit word indicating the transmit rate at which the associated descriptor data will be transmitted. This rate is expressed in terms of descriptor ring polling intervals. For example, a value of 0 indicates the data should be transmitted each and every frame (i.e. one descriptor ring polling interval), a value of 1 indicates the data should be transmitted every other frame, a value of 2 indicates the data should be be transmitted every third frame, etc.

- Descriptor Transmit Data – 32-bit word representing the data to be transmitted for this descriptor. Note that parity is expected to already be calculated for the word, which allows testing of parity bit flipping, if desired.

9.8.7. ARINC-429 Receive Control Command

The ARINC-429 Receive Control command allows enabling and disabling of the four ARINC-429 receivers, as well as controlling its speed and polling rate. Once enabled and the polling rate expires for a channel, all received data will be sent out as a UDP datagram (see section 9.9). The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x09)
4	Bus Number
8	Configuration Word
12	Frame Usec

- Bus Number – 32-bit word indicating which ARINC-429 receive bus (0 to 3) this command applies to.

- Configuration Word – 32-bit word which has the following bit definitions:

```
#define C_UDP_ARX_CFG_EN          0x01  // Enable
#define C_UDP_ARX_CFG_DIS         0x02  // Disable
#define C_UDP_ARX_CFG_HSPD        0x04  // Set to high speed
#define C_UDP_ARX_CFG_LSPD        0x08  // Set to low speed
#define C_UDP_ARX_CFG_FRM_USEC    0x10  // Frame Usec valid
```

Where:

C_UDP_ARX_CFG_FRM_USEC indicates the Frame Usec field is valid,

If any individual bits are not set, then no action will be taken with the associated field and the previous setting remains.

- Frame Usec – 32-bit word indicating the rate in microseconds at which all ARINC-429 receive data for associated bus is gathered and transmitted as a single UDP datagram. The recommended setting for this field is 50000. Note that a value of 0 (default) indicates no UDP datagrams will be sent out for the channel.

9.8.8. Set NIC IDs Command

The Set NIC IDs command specifies which NIC IDs, if any, the DataTap-10 should act as. This effectively enables or disables ASCB transmission from the DataTap-10. If the field for a given bus is zero, the DataTap-10 is put into receive-only mode and no transmission occurs. If the field is non-zero, the DataTap-10 will act as the specified NIC ID for each bus and ASCB transmissions will occur. The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x0A)
4	ASCB 0 NIC ID
8	ASCB 1 NIC ID

Note that each NIC ID is specified as a number and not as a bitfield as is returned in the status datagrams. For example, if the DataTap-10 is to act as NIC ID 0x05 on ASCB bus 0 and NIC ID 0x22 on ASCB bus 1, the values 0x05 and 0x22 would be written directly into the payload, respectively.

9.8.9. Set Discrete Outputs Command

The Set Discrete Outputs allows controlling the state of the four discrete outputs of the DataTap-10. Note that if the health output is enabled via the system configuration file, this command will have no effect. The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x0B)
4	Discrete Outputs State

Discrete Outputs State is a 32-bit bitmask [31:0] with the lower four bits each representing a single discrete output (i.e. bit 0 represents discrete output 0).

9.8.10.ARINC-717 Receive Control Command

The ARINC-717 Receive Control allows control of the DataTap-10's ARINC-717 decoder, including enabling and disabling as well as specifying the word rate. Once enabled and synchronized to the incoming ARINC-717 data stream, the DataTap-10 will send out a UDP packet containing the received data each frame, which is always once per second. The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x0C)
4	Bus Number
8	Configuration Word
12	Word Rate

- Bus Number – 32-bit word indicating which ARINC-717 receive bus this command applies to. Note that only one decoder exists at this time, so this field must be 0.

- Configuration Word – 32-bit word which has the following bit definitions:

```
#define C_UDP_ARX_CFG_EN      0x01   // Enable
#define C_UDP_ARX_CFG_DIS     0x02   // Disable
```

- Word Rate - 32-bit word indicating the expected ARINC-717 word rate. Valid values are 64, 128 256, 512, 1024, and 2048. Note that no data will be received if this does not match the actual word rate of the bus.

9.8.11.Set ASCB Mode Command

The Set ASCB Mode allows setting whether the DataTap-10 decodes the ASCB bus as Enhanced-ASCB or as ASCB-D. The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x0D)
4	Configuration Word

- Configuration Word – 32-bit word which can contain the following values:

0 = ASCB-D Mode
1 = Enhanced-ASCB Mode

9.8.12.Set Multiple NIC IDs Command

The Set Multiple NIC IDs specifies which NIC IDs, if any, the DataTap-10 should act as. This effectively enables or disables ASCB transmission from the DataTap-10. This command is different from the Set NIC IDs command in that it allows the DataTap-10 to act as multiple NIC IDs specified as a bitmask for each ASCB bus. If the field for a given bus is zero, the DataTap-10 is put into receive-only mode and no transmission occurs. If the field is non-zero, the DataTap-10 will act as the specified NIC ID(s) for each bus and ASCB transmissions will occur. The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x0E)
4	ASCB 0 NIC IDs
8	ASCB 1 NIC IDs

- ASCB 0 NIC IDs – 32-bit word indicating which NIC ID(s) the DT-10 should act as for ASCB Bus 0. This word is specified as a bitmask with each bit representing an individual NIC ID. In this manner, any combination of desired NIC IDs can be specified. The bitmask is defined as follows:

```
0x00000001 – NIC ID 1
0x00000002 – NIC ID 2
0x00000004 – NIC ID 3
       …
0x40000000 – NIC ID 31
```

- ASCB 1 NIC IDs – 32-bit word indicating which NIC ID(s) the DT-10 should act as for ASCB Bus 1. This word is specified as a bitmask with each bit representing an individual NIC ID. In this manner, any combination of desired NIC IDs can be specified. The bitmask is defined as follows:

```
0x00000001 – NIC ID 33
0x00000002 – NIC ID 34
0x00000004 – NIC ID 35
       …
0x40000000 – NIC ID 63
```

9.8.13.ARINC-429 Record Control Command

The ARINC-429 Record Control command allows control of ARINC-429 recording to the DataTap-10's selected media card. Note that all enabled ARINC-429 channels will be recorded. The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x0F)
4	Control Word

- Control Word – 32-bit word which has the following bit definitions:

```
#define C_UDP_ARX_REC_EN     0x01   // Enable A429 recording
#define C_UDP_ARX_REC_DIS    0x02   // Disable A429 recording
```

9.8.14.ASCB Record Control Command

The ASCB Record Control command allows control of ASCB raw or decoded data recording to the DataTap-10's selected media card. The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x10)
4	Control Word
8	ASCB Record Type

- Control Word – 32-bit word which has the following bit definitions:

```
#define C_UDP_ASCB_REC_EN     0x01   // Enable ASCB recording
#define C_UDP_ASCB_REC_DIS    0x02   // Disable ASCB recording
```

- ASCB Record Type – 32-bit word which can have the following values:

  1 = Record ASCB Raw Data
  2 = Record ASCB Decoded Data

.

9.8.15.ASCB Playback Control Command

The ASCB Playback Control command controls playback of ASCB raw recordings currently residing on the DataTap-10's selected media card. This command will automatically switch the DataTap-10 into playback mode prior to searching for the specified playback file. If a playback is currently in progress, it will be restarted using the specified file upon receiving this command.

The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x11)
4	Control Word
8	[31:16] IRIG4, [15:0] IRIG3
12	[31:16] IRIG2, [15:0] IRIG1
16	Filename

- Control Word – 32-bit word which has the following bit definitions:

```
#define C_UDP_ASCB_PB_RX_RAW_EN    0x01   // Enable ASCB Rx Raw Playback
#define C_UDP_ASCB_PB_RX_DEC_EN    0x02   // Currently Unavailable
#define C_UDP_ASCB_PB_DIS          0x04   // Disable Playback
#define C_UDP_ASCB_PB_TX_RAW_EN    0x10   // Enable ASCB Tx Raw Playback
#define C_UDP_ASCB_PB_FL_TX_EN     0x20   // Enable ASCB FlightLine Tx Playback
```

- IRIG43 and IRIG21 – Indicates the playback start time. The DataTap-10 will search the recording for the specified IRIG time and begin playback from that point. Note that an exact time does not need to be specified because the DataTap-10 stops searching once a time in the recording later than the specified time is found. The IRIG fields can be decoded as follows:

Field	[15:12]	[11:8]	[7:4]	[3:0]
IRIG1	10^0 mS	10^2 uS	10^1 uS	10^0 uS
IRIG2	10^1 Seconds	10^0 Seconds	10^2 mS	10^1 mS
IRIG3	10^1 Hours	10^0 Hours	10^1 Minutes	10^0 Minutes
IRIG4	0	10^2 Days	10^1 Days	10^0 Days

- Filename – Indicates the playback filename. Note that because the payload is 100 bytes, only a filename less than or equal to 84 bytes can be specified.

9.8.16.Clear Status Error Count Command

The Clear Status Error Count command allows clearing of the ASCB parameter status error count. This count is displayed over Telnet and sent in the UDP Status Output datagrams. This command does not require any parameters in the cmd_data field, as shown:

Payload Offset	Parameter
0	Command (0x12)

9.8.17.Clear Recording Buffer Maximum Occupancy Statistic Command

This command is used to clear the recording buffer maximum occupancy statistic that is displayed over Telnet and sent out in the Status Output datagrams. It does not require any parameters in the cmd_data field, as shown:

Payload Offset	Parameter
0	Command (0x13)

9.8.18.ARINC-717 Recording Control Command

The ARINC-717 Record Control command allows control of ARINC-717 recording to the DataTap-10's selected media card. The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x14)
4	Control Word

- Control Word – 32-bit word which has the following bit definitions:

```
#define C_UDP_ARX_REC_EN     0x01    // Enable A717 recording
#define C_UDP_ARX_REC_DIS    0x02    // Disable A717 recording
```

9.8.19.Multi-Stream Record Control Command

The Multi-Stream Record Control command allows recording control of one or more data streams to the DataTap-10's selected media card. If a recording is already in progress and the Control Word indicates to enable recording, all recording will be stopped first. Recording will then be restarted based on the Record Type(s). If the Control Word indicates to disable recording, all recording will be stopped and will remain disabled until another command is received. Note that the ASCB Raw and ASCB Decoded recording bits may not be set simultaneously when enabling recording. If both are set, the ASCB Decoded bit will be ignored.

The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x15)
4	Control Word
8	Record Type(s)

- Control Word – 32-bit word which has the following bit definitions:

```
#define C_UDP_MULT_REC_EN      0x01    // Enable recording
#define C_UDP_MULT_REC_DIS     0x02    // Disable recording
```

- Record Type(s) – 32-bit word which can be any combination of the following values and is only valid when the Control Word indicates to enable recording:

```
#define C_UDP_MREC_ASCB_RAW    0x01    // ASCB Raw recording
#define C_UDP_MREC_ASCB_DEC    0x02    // ASCB Decoded recording
#define C_UDP_MREC_A429        0x04    // ARINC-429 recording
#define C_UDP_MREC_A717        0x08    // ARINC-717 recording
```

9.8.20.Clear Frame Exceeded Counts Command

The Clear Frame Exceeded Counts command allows clearing of the counters used to track how many times the parameter encode and decode engines have exceeded their allotted 12.5 millisecond frame time. See section 7.1 for details regarding the monitoring of these counts. It does not require any parameters in the cmd_data field, as shown:

Payload Offset	Parameter
0	Command (0x16)

9.8.21.Switch to Live Mode Command

The Switch to Live Mode command switches the DataTap-10 from Playback Mode to Live Mode. It does not require any parameters in the cmd_data field, as shown:

Payload Offset	Parameter
0	Command (0x17)

9.8.22. Set PTP Time Command

The Set PTP Time command allows setting of the PTPv2 (IEEE1588-v2) time within the DataTap-10. Note that in order for this time to be used as the system time, the "USE_TIMING_SOURCE" parameter in the system.cfg file must be set to "IEEE1588v2".

The command and cmd_data fields have the following meaning:

Payload Offset	Parameter
0	Command (0x18)
4	[31:16] IRIG4, [15:0] IRIG3
8	[31:16] IRIG2, [15:0] IRIG1

- IRIG4, IRIG3, IRIG2, IRIG1 – Indicates the time to set based on the following bit definitions:

Field	[15:12]	[11:8]	[7:4]	[3:0]
IRIG1	10^0 mS	10^2 uS	10^1 uS	10^0 uS
IRIG2	10^1 Seconds	10^0 Seconds	10^2 mS	10^1 mS
IRIG3	10^1 Hours	10^0 Hours	10^1 Minutes	10^0 Minutes
IRIG4	0	10^2 Days	10^1 Days	10^0 Days

9.9. DataTap-10 ARINC-429 Rx UDP Output Datagrams

The ARINC-429 Rx UDP datagrams are sent out at the configured polling rate whenever an ARINC-429 receive channel is enabled, the polling rate is non-zero, and labels are being received. Each label has an associated rate in microseconds that is calculated by the DataTap-10 and placed into the datagram. The packet may be fragmented at the IP level by the DataTap-10 if it exceeds the maximum transfer unit (MTU) of the Ethernet link.

The maximum number of labels in a datagram is 738 when the MTU of the Ethernet link is at its standard value of 1500 bytes. Any labels received beyond this in a single polling period will be dropped. This means the polling period should be set fast enough to avoid any overflow. If the polling period is left at its default and recommended value of 50 milliseconds, no overflow is possible.

9.9.1. Data Structure of the ARINC-429 Rx UDP Output Datagrams

```
typedef struct
{
   uint32   num_labels;      // Number of Labels in this Datagram
   RX_DATA  labels[max 738]; // Received Data and Rate

} A429_RX_DATAGRAM;
```

Where RX_DATA is defined as:

```
typedef struct
{
   uint32 rx_data;
   uint32 rate_usec;

} RX_DATA;
```

Note that all fields in the datagram are in Little Endian format rather than network byte order, meaning they can be read directly when on a Little Endian machine. The functions ntohl and ntohs should not be used.

9.10. DataTap-10 ARINC-717 Rx UDP Output Datagrams

The ARINC-717 Rx UDP datagrams are sent out once per second whenever the ARINC-717 decoder is enabled and it is synchronized to the incoming bitstream at the configured word rate. Each received 12-bit word is zero-extended to 16-bits and placed into the datagram. Each datagram contains exactly one full frame of data, meaning its size will always be the configured word rate times 2 bytes each word. The packet may be fragmented at the IP level by the DataTap-10 if it exceeds the maximum transfer unit of the Ethernet link.

9.10.1.Data Structure of the ARINC-717 Rx UDP Output Datagrams

```
typedef struct
{
   uint32 IRIG1;
   uint32 IRIG2;
   uint32 IRIG3;
   uint32 IRIG4;

   uint16 rx_data[max 8192];   // Received Data

} A717_RX_DATAGRAM;
```

Where the IRIG time can be decoded as follows:

Field	[31:16]	[15:12]	[11:8]	[7:4]	[3:0]
IRIG1	0	10^0 mS	10^2 uS	10^1 uS	10^0 uS
IRIG2	0	10^1 Seconds	10^0 Seconds	10^2 mS	10^1 mS
IRIG3	0	10^1 Hours	10^0 Hours	10^1 Minutes	10^0 Minutes
IRIG4	0	0	10^2 Days	10^1 Days	10^0 Days

Note that all fields in the datagram are in Little Endian format rather than network byte order, meaning they can be read directly when on a Little Endian machine. The functions ntohl and ntohs should not be used.

9.11. DataTap-10 ASCB-A/B/C Output Datagrams

The ASCB-A/B/C Output Datagrams each contain a full frame of data from one ASCB bus and always have a payload of 2048 bytes. The first 16 bytes of the payload contain the IRIG time. Note that no datagrams for a given bus will be sent out if there is no activity on that bus. The datagram for ASCB bus 0 is sent out on the port specified by "UDP_HOST_PORT_RAW" in the system configuration file and the datagram for ASCB bus 1 is sent out on that port number plus 1.

9.11.1.Data Structure of the ASCB-A/B/C Output Datagrams

```
typedef struct
{
   uint32 IRIG1;
   uint32 IRIG2;
   uint32 IRIG3;
   uint32 IRIG4;

   uint08 data[2048-16];

} ASCBC_DATAGRAM;
```

Where the IRIG time can be decoded as follows:

Field	[31:16]	[15:12]	[11:8]	[7:4]	[3:0]
IRIG1	0	10^0 mS	10^2 uS	10^1 uS	10^0 uS
IRIG2	0	10^1 Seconds	10^0 Seconds	10^2 mS	10^1 mS
IRIG3	0	10^1 Hours	10^0 Hours	10^1 Minutes	10^0 Minutes
IRIG4	0	0	10^2 Days	10^1 Days	10^0 Days

10. Configuration Data Structures

10.1. DataTap-10 CFG File Format

The Telemetry Interface Configuration file that is produced by the Windows based configuration tool is a binary file of 96-byte records that contain information about each parameter placed in the channel map.

The first entry in the .CFG file is a single 32-bit signed int value indicating the total number of records in the file. The remaining entries are 96-byte records in an Intel Little Endian format as defined in the structure below. Note that the structure is the same for ASCB-D and Enhanced ASCB.

```
/*--------------------------------------------------------------*/
/* This Structure was intentionally made 96 bytes long so that   */
/* each record would line up on a 16 byte boundary in the config */
/* file.                                                         */
/*--------------------------------------------------------------*/

struct Config_struct
       {
         unsigned long Function_ID ;
         unsigned long Instance ;
         unsigned long Channel ;
         unsigned long Parameter_ID ;
         unsigned long PType ;                // Parameter type for PDD API Decode
         unsigned long PRate ;                // Parameter Rate for PDD API Decode
         unsigned long DPRAM_Address ;        // Address in DPRAM of parameter
         unsigned long DPRAM_Size ;           // Parameter Size in Words (32-bit)
         char Parameter_Name[64] ;

       } Config ;
```

See field definitions on following pages.

Parameter Handle Definitions:

```
Function_ID ;
Instance ;
Channel ;
Parameter_ID ;
```

These 4 32-bit words are combined to form the 64-bit parameter handle for the parameter that was selected for monitoring and conversion. These are 4 separate 32-bit fields in the TIS configuration file, they are combined by the DataTap-10 during initialization into a form usable for decoding the ASCB parameters as shown below.

Each instance of a Parameter has a 64-bit handle that can be used internally to uniquely identify it in a Primus Epic system. The following diagram shows the four fields that compose of a parameter handle. Please note that bit 0 is the least significant bit and bit 63 is the most significant bit.

63 – 48	47 – 32
Function ID	Instance

31 – 24	23 – 0
Channel	Parameter ID[1]

This encoding of the parameter handle results in the following maximum limits on these configuration elements in Primus Epic:

Functions	65,535 in Primus Epic, all configurations (0 is used for system-wide definitions like Global Enumeration Sets)
Parameters	16,777,215 in each Function (0 designates the Function class)
Instances (Function instance)	65,535 of each Function, Application, and Parameter (Instance 0 designates the template or class)
Channels (Application instance)	256 of each Application and Parameter, for a Function instance. 0 corresponds to channel “a”, 1 to channel “b”, etc. If a Function instance has only one instance of an application, the channel will be set to 0.

The Function IDs and Parameter IDs are created automatically as new Functions and their associated Parameters are defined. The Parameter ID in the handle uniquely identifies a Parameter within a Function’s scope. This means that each Parameter (and Parameter Group) in a Function has a distinct Parameter ID, but that Parameters in other Functions could have the same ID. Each Function is configured to specify how many instances of each of its Applications are actually allowed, which then determines the number of instances and channels of its Parameters.

[1] The Parameter ID value used in handles is stored in the FunctionParameterId field of the Parameter table in ESCAPE databases. Users must take care not to mistakenly use the Parameter table’s id field.

Datatypes

Ptype Definitions:

Value	Type
1	32-bit float
2	Binary (Binary data array)
3	32-bit signed int
4	Text (Character data array)
5	Uint 32-bit unsigned int
6	Flagstatus, Discrete Status, and Numeric Status
7	All other Enumerated Sets
8	Bitfields

Each parameter is defined to have a fixed datatype, which affects the way values of the parameter are encoded for transmission on the ASCB and, usually, the way the parameter values are formatted in the API between the applications and the ASCB I/O Software. For ASCB-D, each parameter to be transmitted periodically over the bus must have a datatype of Integer, Unsigned Integer, Float, Enum, Text, or Binary.
ASCB-D does not support a BCD datatype other than by using an Enumeration Set to represent single digits. Applications should convert BCD values to one of the support numeric datatypes prior to transmission on ASCB-D.

Table 12. Parameter Datatypes Available in ASCB-D

Datatype	Usage and Representation	Required Parameter Attributes
Integer	Numbers in the range of –2,147,483,648 to 2,147,483,647 that have no fractional component (i.e., have a resolution of 1).	Minimum and Maximum Scale Values
Unsigned Integer	Numbers in the range of 0 to 4,294,967,295 that have no fractional component (i.e., have a resolution of 1)	Minimum and Maximum Scale Values
Float	Numbers that have a fractional component or that have a range beyond that supported by the Integer type	Minimum and Maximum Scale Values; Minimum required Resolution
Enum	Parameters that have multiple (up to 256) discrete states.	A list of the possible states for the parameter. An unsigned integer value (0-255) and a text string define each state.
Text	A fixed size text string of ASCII characters.	Size in bytes
Binary	A fixed size array of bytes, the encoding of which is unknown except by the producer and the consumers of the values.	Size in bytes

Enumeration Set Definition and Naming

The ASCB-D data definition supports the definition of Enumeration Sets as a convenience to simplify the specification of multiple Enum parameters that have the same set of possible states. An Enumeration Set is similar to a typedef of an enum in the C and C++ languages. It allows a user to give a name to a set of states that some variables can have, and then allows the name to be used like a datatype for defining the variables that can have those states.

An Enumeration Set is defined as a parameter with a datatype of “EnumSet”. An Enumeration Set has four attributes: a Name, a Description, a Function, and a list of possible states. The scope of an EnumSet’s name is normally a single Epic Function, meaning that two different functions can create Enumeration Sets with the same Name attribute, but with different sets of possible states. However, when an EnumSet is defined in the System function, it’s scope is the entire Primus Epic system, meaning that all Epic functions will share the use of the Enumeration Set and that individual functions may not create an EnumSet with the same Name attribute. Enumeration Set names can consist of up to sixteen characters composed of letters (A-Z, a-z) and numeric digits (0-9). The letters can be uppercase or lowercase, but uniqueness within scope is not case sensitive. The first character must be a letter, not a number, and will always be stored and displayed in uppercase. Spaces and other special characters are not allowed.

Table 13. Enumeration Set Defining Construct

Datatype	Usage	Required Parameter Attributes
EnumSet	A pseudo-parameter used to define a set of possible states shared by multiple parameters of type Enum produced by a Function. Once declared, the name of an EnumSet can be used as a datatype for parameters in the Function that have its defined states. EnumSets defined in the System function have a system-wide scope, meaning that they can be used as datatypes for Parameters in all Functions.	A list of the possible states, each defined by an integer value in the range 0-255 and a text string.

After a Enumeration Set has been created, parameters of type Enumeration can be defined to use the Enumeration Set by setting the datatype of the parameter to the Name of the Enumeration Set.

For example, there may be many two-state parameters that are defined such that a value of 0 means OFF and a value of 1 means ON. To ease the definition of these parameters, first use ESCAPE to define an Enumeration Set with the Name “OffOn” and possible states of “0=OFF, 1=ON”. Then define each of the parameters with the datatype of OffOn, and leave their possible states blank. In this way the parameters will be effectively defined as parameters of type Enum with the possible states of “0=OFF, 1=ON”.

Enumeration Set Defined for Status Parameters

When a Parameter is defined in ASCB-D, it can be associated with a separate *Status Parameter*, whose value indicates the validity or quality of the value of the Parameter. More than one Parameter can be associated with the same Status Parameter. Parameters must be in the same Parameter Group as their associated Status Parameter. When an application reads a Parameter using the PDD, the PDD returns the Parameter's value and, optionally, it's status. If the Parameter read had an associated status parameter, the returned status will reflect the value of the status parameter, plus provide indicators for stale data and data integrity failures. If the Parameter had no associated Status Parameter, the returned status only indicates stale data and integrity failures.
Table 14 lists several formats of status variables that are supported by the PDD and the datatype that have been defined to indicate Status Parameters. These datatypes are predefined in the system as global Enumeration Sets. A Parameter must be defined with one of these types to be used as a Status Parameter.

Table 14. Global Enumeration Sets Defined as Status Parameter Datatypes

Datatype	Description	Possible States
FlagStatus	A two state value indicating good or bad for use as PDD status parameter	0 = Invalid 1 = Valid
NumericStatus	A four state value encoded like an ARINC 429 Binary Numeric SSM for use as PDD status parameter	0 = Failure Warning 1 = No Computed Data 2 = Functional Test 3 = Normal Operation
DiscreteStatus	A four state value encoded like an ARINC 429 Discrete SSM for use as PDD status parameter	0 = Normal Operation 1 = No Computed Data 2 = Functional Test 3 = Failure Warning

The values of status parameters are always mapped to a two-bit value with the format of the DiscreteStatus datatype prior to transmission on the sending side. Hence status parameters always occupy two bits on the ASCB_D.

PRate Definitions:

This parameter defines the rate in integer Milliseconds of the parameter rate on the ASCB-D Bus. Note that one valid ASCB-D rate is 12.5 ms. The 12.5 ms rate is defined as 12 ms in the file. All other rates are representable in an integer value. i.e. 25 ms, 50 ms, 100ms, …

DPRAM_Address Definition:

This parameter defines what address in Dual Port Memory of an APIC interface card the parameter is programmed to start at. The DataTap-10 TIS does not have an APIC interface card, but instead outputs that data in UDP datagrams. This value in this field is 8 times the programmed channel number. If you divide this number by 8 you will get the programmed UDP channel number.

DPRAM_Size Definition:

This parameter normally defines the number of words of Dual Port RAM a parameter occupies in an APIC card. In the TISDT1 application, this defines the number of channels the parameter occupies. It is possible to have a parameter larger than 32-bits on the ASCB D bus. If you have selected one of these parameters, then this number will be the number of channels the parameter occupies. It is normally 1 channel or 32-bits.

Parameter_Name:

Parameter_Name[64] is the character string that names the parameter being converted.

11. Electrical, Mechanical and Environmental Specifications

11.1. DataTap-10 Standard Enclosure Environmental

Operating Temperature: -45°C to +70°C
Storage Temperature: -55°C to +125°C
Humidity: 10 to 90% RH, noncondensing (optional conformal coating available for condensing environments)
Altitude: -1500ft to +20,000ft

11.2. DataTap-10 Flight Test Enclosure Environmental

Operating Temperature: -45°C to +70°C
Storage Temperature: -55°C to +125°C
Humidity: 10 to 90% RH, noncondensing (optional conformal coating available for condensing environments)
Altitude: -1500ft to +20,000ft

11.3. Datatap-10 Standard Enclosure Mechanical

Weight: < 13 oz (370 Grams) in standard configuration
138 mm L x 80mm W x 48 mm H

11.4. Datatap-10 Flight Test Enclosure Mechanical

Weight: < 1 Kg in standard configuration

11.5. Electrical Requirements (Standard and Optinal Enclosure)

Power Requirements: 4.5-6.5VDC (USB Power Connector) ≈500 ma. (≈750 ma. ASCB-D TX)
Power Requirements: 9-36VDC (External Power Connector) ≈2.5W. (≈4W ASCB-D TX)

One or both power sources may be supplied simultaneously, but only one source is required.

12. FPGA Revision History

- Revision 12091601 – December 9, 2016
    - Added support for ASCB-A.
    - Fixed a bug in the IRIG logic which in rare cases caused locking to the IRIG source with exactly a 1 or 2 millisecond offset.
    - Fixed another bug in the IRIG logic where ADC channels 1 through 3 (zero-based) were not automatically read by the logic when synced to an IRIG source with a DC offset.
- Revision 06141601 – June 14, 2016
    - Increased the sampling frequency in the ASCB-D and eASCB receive logic from 120MHz to 140MHz to match the recent change for the MAU NICs.
    - Made the ASCB-D and eASCB sampling points configurable in software.
    - Added interrupt coalescing timers to the ARINC-429, ARINC-717, and ASCB-C receive modules to significantly reduce CPU usage.
    - Modified the Real-Time Clock logic in the Ethernet controller to allow a much more accurate IEEE1588v2 (PTPv2) implementation by allowing the master offset to be specified in hardware rather than having software make the adjustment.
    - Increased overall performance by increasing the size of the instruction caches by sacrificing a portion of the Processor 0 data cache.
- Revision 12291501 – December 29, 2015
    - Added an 8-input mux to allow any discrete input or ARINC-429 channel to be used for the ARINC-717 input.
    - Added an additional 7 sync detection units to the ARINC-717 decoder to allow faster synchronization to the incoming data stream.
    - Increased the depth of the ARINC-429 receive FIFOs from 16 to 32.
    - Tied the SD controller logic directly into the memory controller rather than over the main bus to increase performance.
- Revision 09141501 – September 14, 2015
    - Increased the hardware buffering for ASCB-C to prevent buffer overruns.
    - Added modes for production testing (internal use only).
- Revision 01061501 – January 6, 2015
    - Added the ability to receive ASCB version C data.
    - Added a truncation mode when encoding float parameters for transmission onto the ASCB bus. Previous FPGA revisions used a rounding mode. Software-based encoding algorithms such as FlightLine and Honeywell TIUs use truncation, leading to slightly different results in some cases. Consequently, an optional truncation mode was added to the DataTap-10. Note that the DataTap-10 must have the optional ASCB transmission capability enabled in order to transmit onto the bus. The transmitters are removed on receive-only units.

- Revision 09031401 – September 3, 2014
  - Made a change to the CompactFlash ATA interface to avoid long delays in returning the CRC during Ultra DMA reads when the system gets busy. This allows playback from the CompactFlash to occur at full speed using Ultra DMA.
  - Added support for Big Endian format NIC and PDD registries.
  - Integrated reading of all ADC channels into the IRIG decode module to significantly reduce CPU usage and allow fast reading of all channels without interfering with the IRIG decode logic.
  - Added memory access contraints to all NIC and PDD modules to prevent bad registries from corrupting system memory.
  - Added a hardware clock used during playback which prevents the need to borrow the PTP clock as in earlier revisions.
- Revision 05121401 – May 12, 2014
  - Fixed a bug in the CompactFlash interface logic where the DMACK signal was left asserted if the card was pulled out in the middle of a UDMA write.
- Revision 04231401 – April 23, 2014
  - Added support for recording to CompactFlash.
  - Added support for encoding ASCB parameters in hardware rather than software.
- Revision 08301301 – August 30, 2013
  - Added support for decoding ASCB parameters in hardware rather than software.
  - Added support for 4MB and 8MB local RAMs.
  - Added PTPv2 hardware timestamping support to the Ethernet MAC.
- Revision 10301201 – October 30, 2012
  - Added support for Enhanced-Mode ASCB (eASCB) transmission and reception.
  - Added support for the 4-bit SD interface to support ASCB recording to the microSD card.
- Revision 02131201 – February 13, 2012
  - Initial revision.

13. System Configuration File

```
##############################################################
# Configuration File Version: 1.13                           #
##############################################################

##############################################################
# Define the MAC address, IP address, and network mask for   #
# the DataTap-10.                                            #
#                                                            #
# Defaults:                                                  #
#                                                            #
# BOARD_MAC                 00:0a:35:00:01:01                #
# BOARD_IP                  192.168.1.1                      #
# NET_MASK                  255.255.255.0                    #
#                                                            #
##############################################################

BOARD_MAC                   00:0a:35:00:01:01
BOARD_IP                    192.168.1.1
NET_MASK                    255.255.255.0


##############################################################
# Define the host IP addresses. The specified addresses will #
# be put into the IP destination address field of the IP     #
# header of the associated output packets. Note that         #
# broadcast IP addresses may be used such that multiple PCs  #
# can receive the same data.                                 #
#                                                            #
# The following are used for all ASCB versions:              #
#                                                            #
# HOST_IP_STS  - Host IP for status messages                 #
# HOST_IP_SYNC - Host IP for ASCB frame sync messages        #
# HOST_IP_A429 - Host IP for ARINC-429 data output           #
# HOST_IP_A717 - Host IP for ARINC-717 data output           #
#                                                            #
# The following are only used for ASCB-D and eASCB:          #
#                                                            #
# HOST_IP_DEC  - Host IP for ASCB decoded data output        #
# HOST_IP_RAW  - Host IP for ASCB raw data output            #
#                                                            #
# Default: 192.168.1.200                                     #
#                                                            #
##############################################################

HOST_IP_STS                 192.168.1.200
HOST_IP_DEC                 192.168.1.200
HOST_IP_RAW                 192.168.1.200
HOST_IP_SYNC                192.168.1.200
```

```
HOST_IP_A429              192.168.1.200
HOST_IP_A717              192.168.1.200


##############################################################
# Indicate the ASCB version. ASCB versions A, B, C, and D,   #
# as well as Enhanced-ASCB are all supported with the        #
# DataTap-10. Note that ASCB-B and ASCB-C are only           #
# supported in FPGA version 01061501 and later, and ASCB-A   #
# is only supported in FPGA version 12091601 and later.      #
#                                                            #
# This parameter has superseded ENHANCED_ASCB_MODE to allow  #
# support for ASCB-A/B/C. To maintain backwards-             #
# compatibility with older configuration files,              #
# ENHANCED_ASCB_MODE can still be used, with a value of      #
# FALSE indicating ASCB-D and TRUE indicating Enhanced-ASCB  #
# (eASCB). ENHANCED_ASCB_MODE and ASCB_VERSION should not    #
# both be present in the same configuration file as only the #
# one furthest down in the file will be used.                #
#                                                            #
# Default:        ASCB-D                                     #
#                                                            #
# Valid Values:   ASCB-A   (ASCB version A)                  #
#                 ASCB-B   (ASCB version B)                  #
#                 ASCB-C   (ASCB version C)                  #
#                 ASCB-D   (ASCB version D)                  #
#                 eASCB    (Enhanced-ASCB)                   #
#                                                            #
##############################################################

ASCB_VERSION              ASCB-D


##############################################################
# Set the NIC IDs which this DataTap-10 should act as for    #
# each ASCB bus. This applies to ASCB transmission only. The #
# 32-bit fields represent a bit mask where the LSB indicates #
# NIC ID 1 and the MSB represents NIC ID 32. For the copilot #
# bus (ASCB 1), an offset of 32 is added to the NIC ID       #
# representations. Note that multiple NIC IDs can be         #
# specified.                                                 #
#                                                            #
# A value of AUTO can also be specified for either bus. In   #
# this case, the host NIC ID field will automatically be     #
# extracted from the associated transmit ESCAPE registry and #
# the DataTap-10 will be assigned that NIC ID value. Note    #
# that this is only valid when the HW_PDD_ENABLE parameter   #
# is set to TRUE. If HW_PDD_ENABLE is set to FALSE or a      #
# valid transmit ESCAPE registry is not found for a given    #
# bus, no ASCB transmission will occur on that bus.          #
#                                                            #
```

```
# Note that at least one timing NIC (IDs 1, 2, 33, or 34)    #
# must be present on the bus for ASCB transmission to occur. #
# The NIC IDs can be hard-coded below to include a timing    #
# NIC, if needed, since multiple NIC IDs can be specified.   #
#                                                            #
# These parameters are ignored when the ASCB version is      #
# set to ASCB-A/B/C.                                         #
#                                                            #
# Default:      0x00000000 (no transmission)                 #
# Valid Values: 0x00000000 to 0xFFFFFFFF or AUTO             #
#                                                            #
##############################################################

ASCB0_NIC_IDS             0x00000000
ASCB1_NIC_IDS             0x00000000


##############################################################
# Indicate whether ARINC-429, ARINC-717, and/or ASCB         #
# recording is selected. Recording of all 3 streams is       #
# supported, but only with the media cards recommended by    #
# ICS. If using a discrete input for recording control,      #
# these parameters indicate which streams will be started or #
# stopped based on the discrete input. If AUTO_REC_ENABLE is #
# set to TRUE, they indicate which streams will be recorded  #
# on power-up. Any of these values can be changed            #
# dynamically through Telnet, UDP, or FlightLine.            #
#                                                            #
# ASCB recording can be in either raw or decoded format by   #
# setting RAW_REC_EN or DEC_REC_EN to TRUE, respectively.    #
# Both cannot be set to TRUE or an error will be generated.  #
#                                                            #
# If the parameter AUTO_REC_ENABLE is set to TRUE, recording #
# will begin on power-up after certain conditions are met,   #
# which is referred to as automatic recording. Note that to  #
# maintain compatibility with older config files, the        #
# default value of AUTO_REC_ENABLE is TRUE such that it can  #
# be omitted. Automatic recording will not begin until the   #
# conditions listed below are met in the following order:    #
#                                                            #
#    1) If REC_FILENAME_FORMAT is set to TISNATIVE, the      #
#       selected time source (IEEE1588v2 or IRIGB) is locked #
#       or the timeout has been reached (5 seconds for IRIGB #
#       and 20 seconds for IEEE1588v2).                      #
#                                                            #
#    2) The delay since power-on specified by                #
#       AUTO_REC_DELAY_SEC has expired.                      #
#                                                            #
#    3) If ASCB_REC_WAIT_DATA is set to TRUE, ASCB traffic   #
#       is present.                                          #
#                                                            #
```

```
# Note that for legacy support, FULL_REC can be used in      #
# place of RAW_REC_EN and PARTIAL_REC can be used in place   #
# of DEC_REC_EN. In addition, "yes" can used in place of     #
# TRUE and "no" can be used in place of FALSE, all of which  #
# are case-insensitive.                                      #
#                                                            #
# The DEC_REC_EN parameter is ignored when the ASCB version  #
# is set to ASCB-A/B/C.                                      #
#                                                            #
# Defaults:                                                  #
#                                                            #
# RAW_REC_EN     FALSE                                       #
# DEC_REC_EN     FALSE                                       #
# A429_REC_EN    FALSE                                       #
# A717_REC_EN    FALSE                                       #
#                                                            #
# Valid Values:  TRUE | FALSE                                #
#                                                            #
##############################################################

RAW_REC_EN                 FALSE
DEC_REC_EN                 FALSE
A429_REC_EN                FALSE
A717_REC_EN                FALSE


##############################################################
# Define the base filenames for the raw and decoded          #
# recording. Note that the extension may be omitted since    #
# recording files always start out with .01. If the filename #
# already exists when recording begins, the base name will   #
# be appended with "_2" and incremented from there until a   #
# unique filename is found. This does not apply to circular  #
# recording mode since the filenames are always forced to be #
# unique. Long filenames are supported.                      #
#                                                            #
# Note that "FULL_" may be used in place of "RAW_" and       #
# "PARTIAL_" may be used in place of "DEC_" in the parameter #
# names for legacy support.                                  #
#                                                            #
# The DEC_REC_FILE parameter is ignored when the ASCB        #
# version is set to ASCB-A/B/C.                              #
#                                                            #
# Defaults:                                                  #
#                                                            #
# RAW_REC_FILE  ASCB_RAW                                     #
# DEC_REC_FILE  ASCB_DEC                                     #
#                                                            #
##############################################################

RAW_REC_FILE               ASCB_RAW
```

```
DEC_REC_FILE                ASCB_DEC


##############################################################
# Set whether the DataTap-10's hardware-driven transmit     #
# Periodic Device Driver (PDD) and PDD API are enabled. If   #
# set to TRUE, the modules will be enabled. If set to FALSE, #
# the associated hardware modules will be disabled and       #
# transmission can only occur by having FlightLine or custom #
# software send the required NIC frame buffers into the      #
# DataTap-10 each ASCB frame. Note that in either case, the  #
# correct NIC IDs must be set in order for proper            #
# transmission to occur. This parameter replaces the legacy  #
# HW_PDD_ENABLE parameter. Either parameter can be used and  #
# both have the exact same meaning. The new parameter was    #
# created for clarification since the hardware PDD and PDD   #
# API for reception are always enabled regardless of this    #
# parameter setting.                                         #
#                                                            #
# If set to TRUE, the following TIS configuration files must #
# exist on the microSD if values of specific parameters are  #
# to be dynamically changed via UDP input packets:           #
#                                                            #
# tis_tx0.cfg - TIS Configuration File for ASCB Bus 0 Tx     #
# tis_tx1.cfg - TIS Configuration File for ASCB Bus 1 Tx     #
#                                                            #
# In addition, the following registry files must be         #
# programmed into the DataTap-10 by placing them on the      #
# microSD card at least once:                                #
#                                                            #
# esc_tx0.reg - ESCAPE Registry for ASCB Bus 0 Tx            #
# esc_tx1.reg - ESCAPE Registry for ASCB Bus 1 Tx            #
#                                                            #
# This parameter is ignored when the ASCB version is set     #
# to ASCB-A/B/C.                                             #
#                                                            #
# Default:        FALSE                                      #
# Valid Values:   TRUE | FALSE                               #
#                                                            #
##############################################################

HW_TX_PDD_ENABLE            FALSE


##############################################################
# Indicate whether to record to the DataTap-10's Compact     #
# Flash interface rather than to the microSD card. Recording #
# to the Compact Flash requires a special daughter card and  #
# enclosure that can only be purchased through Innovative    #
# Control Systems. Note that configuration files such as the #
# system configuration file and any TIS configuration files  #
```

```
# must exist on the microSD card regardless of this setting. #
# Any registry files and software updates can only exist on  #
# the microSD card as well.                                  #
#                                                            #
# Default:       FALSE                                       #
# Valid Values:  TRUE | FALSE                                #
#                                                            #
##############################################################

USE_COMPACT_FLASH         FALSE


##############################################################
# Indicates the use of the status LED after initialization,  #
# assuming no errors have occurred. See the DataTap-10       #
# User's Manual for more details on the various states of    #
# the status LED during and after initialization. This       #
# parameter can have any of the following values:            #
#                                                            #
#    FL_CONN - FlightLine Connection. When green, it         #
#              indicates there is a valid connection to      #
#              FlightLine. This is the default setting to    #
#              maintain backwards compatibility.             #
#    REC_STS - Recording Status. See the DataTap-10 User's   #
#              Manual for the state definitions.             #
#    CFG_VLD - Configuration Valid. Once initialization is   #
#              complete, the LED turns static green if there #
#              were no initialization errors. This can be    #
#              useful when the DataTap-10 is used in a       #
#              stand-alone configuration without FlightLine. #
#    OFF     - Off. Once initialization is complete, the LED #
#              turns off if there were no initialization     #
#              errors. This can be useful when the           #
#              DataTap-10 is used in a stand-alone           #
#              configuration without FlightLine.             #
#                                                            #
# Note that the legacy parameter USE_STS_LED_FOR_REC_STS is  #
# still valid, however, it should not be used in the same    #
# configuration file as STS_LED_USE or only the one further  #
# down in the file will be used.                             #
#                                                            #
# Default:       FL_CONN                                     #
# Valid Values:  REC_STS | FL_CONN | CFG_VLD | OFF           #
#                                                            #
##############################################################

STS_LED_USE               FL_CONN


##############################################################
# Define the default state for each of the output message    #
```

```
# types. If the functionality is not required, each can be   #
# individually turned off to save CPU and network bandwidth. #
# They can also be turned off dynamically via the Telnet     #
# server or UDP command packets.                             #
#                                                            #
# UDP_RAW_EN_DEF  - Enable raw ASCB data output. Valid for   #
#                   for all ASCB versions.                   #
# UDP_DEC_EN_DEF  - Enable decoded ASCB data output. Valid   #
#                   only for ASCB-D and eASCB.               #
# UDP_STS_EN_DEF  - Enable status message output. Valid for  #
#                   all ASCB versions.                       #
# UDP_SYNC_EN_DEF - Enable ASCB frame sync message output.   #
#                   Valid only for ASCB-D and eASCB. These   #
#                   frame sync messages are typically only   #
#                   required by a host application when      #
#                   doing NIC-only ASCB transmission with    #
#                   the DataTap-10.                          #
# UDP_A429_EN_DEF - Enable ARINC-429 output. Note that data  #
#                   is only sent out when the associated     #
#                   ARINC-429 bus is active and enabled.     #
# UDP_A717_EN_DEF - Enable ARINC-717 output. Note that data  #
#                   is only sent out when the ARINC-717 bus  #
#                   is synced.                               #
#                                                            #
# Note that FULL_SEND can be used in place of UDP_RAW_EN_DEF #
# and PARTIAL_SEND can be used in place UDP_DEC_EN_DEF for   #
# legacy support, however, the associated send/record        #
# format must be set to DIANE6x in this case.                #
#                                                            #
# Default:        TRUE                                       #
# Valid Values:   TRUE | FALSE                               #
#                                                            #
##############################################################

UDP_RAW_EN_DEF            TRUE
UDP_DEC_EN_DEF            TRUE
UDP_STS_EN_DEF            TRUE
UDP_SYNC_EN_DEF           TRUE
UDP_A429_EN_DEF           TRUE
UDP_A717_EN_DEF           TRUE


##############################################################
# Define the UDP host (outgoing) and listen (incoming)       #
# ports. These typically should not need to be changed       #
# unless the ports are already being used on the host        #
# machine. Note that both the host and listen ports indicate #
# the value in the UDP Destination Port field of the UDP     #
# header.                                                    #
#                                                            #
# The following are used for all ASCB versions:              #
```

```
#                                                            #
# UDP_HOST_PORT_STS  - Host port for status message output.  #
# UDP_HOST_PORT_RAW  - Host port for ASCB raw data output.   #
#                      In ASCB-D and eASCB mode, both bus 0  #
#                      and bus 1 data is sent out on this    #
#                      port. In ASCB-A/B/C mode, bus 0 data  #
#                      will be sent on the specified port,   #
#                      and bus 1 data will be sent on the    #
#                      specified port number plus 1.         #
# UDP_HOST_PORT_A429 - Base host port for ARINC-429 data     #
#                      output. This port and the next 3      #
#                      ports are used to support the 4       #
#                      ARINC-429 receive channels of the     #
#                      DataTap-10.                           #
# UDP_HOST_PORT_A717 - Host port for ARINC-717 data output.  #
# UDP_LSTN_PORT_CMD  - Listen port for command messages.     #
#                                                            #
# The following are only used for ASCB-D and eASCB:          #
#                                                            #
# UDP_HOST_PORT_DEC  - Base host port for ASCB decoded data  #
#                      output. This port and the next 31     #
#                      ports are used to support up to 8192  #
#                      channels.                             #
# UDP_HOST_PORT_SYNC - Base host port for ASCB frame sync    #
#                      messages. This port and the next 2    #
#                      ports are used to support up to 3     #
#                      frame sync messages.                  #
# UDP_LSTN_PORT_RAW  - Base listen port for ASCB raw data    #
#                      input. Raw bus 0 data should be sent  #
#                      in on this port and bus 1 data should #
#                      be sent on the next port.             #
# UDP_LSTN_PORT_SIU0 - Base listen port for ASCB Bus 0       #
#                      Simulator Interface Unit (SIU)        #
#                      datagrams, which are also known as    #
#                      Parameter Input datagrams. This port  #
#                      and the next 31 ports are used to     #
#                      support up to 8192 channels.          #
# UDP_LSTN_PORT_SIU1 - Base listen port for ASCB Bus 1       #
#                      Simulator Interface Unit (SIU)        #
#                      datagrams, which are also known as    #
#                      Parameter Input datagrams. This port  #
#                      and the next 31 ports are used to     #
#                      support up to 8192 channels           #
#                                                            #
# Defaults:                                                  #
#                                                            #
# UDP_HOST_PORT_STS         51000                            #
# UDP_HOST_PORT_DEC         51020                            #
# UDP_HOST_PORT_RAW         51008                            #
# UDP_HOST_PORT_SYNC        51009                            #
# UDP_HOST_PORT_A429        51012                            #
```

```
# UDP_HOST_PORT_A717       51016                               #
# UDP_LSTN_PORT_CMD        51001                               #
# UDP_LSTN_PORT_RAW        51004                               #
# UDP_LSTN_PORT_SIU0       51100                               #
# UDP_LSTN_PORT_SIU1       51200                               #
#                                                              #
################################################################

UDP_HOST_PORT_STS          51000
UDP_HOST_PORT_DEC          51020
UDP_HOST_PORT_RAW          51008
UDP_HOST_PORT_SYNC         51009
UDP_HOST_PORT_A429         51012
UDP_HOST_PORT_A717         51016
UDP_LSTN_PORT_CMD          51001
UDP_LSTN_PORT_RAW          51004
UDP_LSTN_PORT_SIU0         51100
UDP_LSTN_PORT_SIU1         51200


################################################################
# Indicate whether the UDP decoded output datagrams are in    #
# legacy format. Legacy format means that full 256-channel    #
# datagrams are always sent out regardless of the number of   #
# occupied channels in each datagram, assuming at least one   #
# channel is occupied. It also means that a 32-bit CRC is     #
# appended to the datagram and is accounted for in the        #
# length field. In non-legacy mode, the datagram length is    #
# based on the highest occupied channel and does not          #
# include the 32-bit CRC. Note that this option also affects #
# the format of the datagrams being recorded when decoded     #
# recording is enabled.                                        #
#                                                              #
# This parameter is ignored when the PARTIAL_SEND_REC_FORMAT #
# parameter is set to DIANE62.                                 #
#                                                              #
# NOTE: The 32-bit CRC field is always zero when enabled by   #
# the legacy mode.                                             #
#                                                              #
# This parameter is ignored when the ASCB version is set to   #
# ASCB-A/B/C.                                                  #
#                                                              #
# Default:       FALSE                                         #
# Valid Values:  TRUE | FALSE                                  #
#                                                              #
################################################################

UDP_DEC_LEGACY_MODE        FALSE


################################################################
```

```
# Indicate the filename format for ASCB recording. TISNATIVE #
# format appends a timestamp to the base filename specified  #
# by RAW_REC_FILE or DEC_REC_FILE for raw and decoded        #
# recording, respectively. The timestamp is derived from the #
# IRIG or PTP time source and is in the following format:    #
#                                                            #
#    "<base>_dow_mon_dom_hh_mm_ss_yyyy"                      #
#                                                            #
#    where:  dow = Day of week, i.e. Wed                     #
#            mon = 3-letter month, i.e. Aug                  #
#            dom = Day of the month, i.e. 31                 #
#                                                            #
# STANDARD format uses only the value specified by           #
# RAW_REC_FILE or DEC_REC_FILE for the filename. If the file #
# already exists on the card, "_2" will be appended to the   #
# base filename and the 2 will be incremented until a unique #
# filename is found.                                         #
#                                                            #
# In both formats, the filename extension always starts with #
# .01 and is incremented from there as each subsequent file  #
# segment is opened. A file segment is closed just before it #
# reaches 2GB in size and then a new one is opened.          #
#                                                            #
# Note that because the TISNATIVE format relies on the IRIG  #
# or PTP time depending upon the USE_TIMING_SOURCE value),   #
# the DataTap-10 will wait for time synchronization to occur #
# prior to starting a recording if recording is enabled at   #
# power-up through RAW_REC_EN or DEC_REC_EN. If time         #
# synchronization does not occur within 5 seconds for IRIG   #
# or 20 seconds for PTP after the DataTap-10 finishes its    #
# initialization procedure, it will start the recording      #
# anyways using the unsynchronized time. The unsynchronized  #
# time is the elapsed time since power-on. The DataTap-10    #
# does not wait for time synchronization to occur if the     #
# STANDARD format is chosen.                                 #
#                                                            #
# Default:       STANDARD                                    #
# Valid Values:  STANDARD | TISNATIVE                        #
#                                                            #
##############################################################

REC_FILENAME_FORMAT       STANDARD


##############################################################
# Indicate the year for the IRIG time. Many of the IRIG-B    #
# formats do not include the year, so this parameter         #
# compensates for that. The year is only used when creating  #
# filenames in TISNATIVE format and when creating the file   #
# date/time for recordings or FTP "put" operations. This     #
# parameter is ignored if USE_TIMING_SOURCE is not set to    #
```

```
# IRIGB.                                                      #
#                                                             #
# The following IRIG-B formats include the year:              #
#                                                             #
#    B124, B125, B126, B127                                   #
#                                                             #
# If the format of the IRIG-B source is unknown, the year as #
# decoded from the IRIG-B signal is displayed on page 3 of    #
# the Telnet screens. Because only the lower-byte of the      #
# year is transmitted, the decoded year will read 2000" if    #
# it does not exist in the IRIG-B frames.                     #
#                                                             #
# Note that a value of 0 for the year indicates to use the    #
# year from the decoded IRIG-B input. Any non-zero value      #
# greater than or equal to 2014 indicates to use the          #
# specified value. Any other value is invalid.                #
#                                                             #
# Default:       0                                            #
# Valid Values:  0, or 2014 or greater                        #
#                                                             #
###############################################################

IRIG_YEAR                 2016


###############################################################
# Indicate whether the DataTap-10 should verify the CRC on    #
# the PDD ESCAPE Registries during initialization. Older      #
# registries tend to have the CRC appended to the end of the #
# file while registries generated more recently do not. If    #
# the CRC is not present, this value should be set to FALSE. #
#                                                             #
# This parameter is ignored when the ASCB version is set      #
# to ASCB-A/B/C.                                              #
#                                                             #
# Default:       FALSE                                        #
# Valid Values:  TRUE | FALSE                                 #
#                                                             #
###############################################################

ESC_REG_CRC_EN            FALSE


###############################################################
# Indicate the default setting for each ASCB receive switch. #
# Note that the receive switch settings can be changed        #
# dynamically through Telnet, the FlightLine software, or     #
# UDP command packets.                                        #
#                                                             #
# Default:       PRIMARY                                      #
# Valid Values:  PRIMARY | BACKUP                             #
```

```
#                                                             #
###############################################################

ASCB0_RXSW_DEF            PRIMARY
ASCB1_RXSW_DEF            PRIMARY


###############################################################
# Indicate whether to place the DataTap-10 into Passthru     #
# Mode, where all ASCB traffic received on the associated    #
# bus is sent out over Ethernet, similar to a DataTap-5.     #
# Note that the DataTap-10 cannot be used for any other      #
# purpose if in Passthru Mode on either ASCB bus.            #
#                                                             #
# These parameters are ignored when the ASCB version is set  #
# to ASCB-A/B/C.                                              #
#                                                             #
# Default:       FALSE                                        #
# Valid Values:  TRUE | FALSE                                 #
#                                                             #
###############################################################

ASCB0_PASSTHRU_EN         FALSE
ASCB1_PASSTHRU_EN         FALSE


###############################################################
# Indicate the mode for encoding of ASCB float parameters    #
# when the hardware PDD transmission is enabled. Float       #
# parameters are scaled and then converted to integers as    #
# part of the encoding process prior to being sent on the    #
# ASCB bus. In FPGA versions prior to 01061501, rounding was #
# used by the DataTap-10 during the conversion to an         #
# integer. Honeywell TIUs and the software-based ASCB        #
# transmission in FlightLine, however, use truncation based  #
# on the standard C library, potentially leading to slightly #
# different results. Consequently, a new logic core was      #
# developed and added to the DataTap-10 to allow either      #
# rounding or truncation.                                     #
#                                                             #
# This parameter is ignored when the ASCB version is set     #
# to ASCB-A/B/C.                                              #
#                                                             #
# This parameter will have no effect for FPGA versions prior #
# to 01061501.                                                #
#                                                             #
# Default:       ROUND                                        #
# Valid Values:  TRUNCATE | ROUND                             #
#                                                             #
###############################################################
```

```
ASCB_FLOAT_ENCODE_MODE    ROUND


##############################################################
# Configure the settings for the playback functionality.     #
# Playback allows any ASCB raw recording file set residing   #
# on the selected media type to either be transmitted onto   #
# the ASCB bus (TXRAW) or to be fed in as ASCB input data    #
# with no transmission onto the ASCB bus (RXRAW).            #
#                                                            #
# For ASCB-D and eASCB, decoded parameters can be viewed     #
# over Telnet, through FlightLine, or through custom         #
# software just as with live mode. Because raw recording     #
# records the entire ASCB bus, any set of decoded parameters #
# can be viewed post-recording simply by changing the TIS    #
# configuration file in the DataTap-10 and/or FlightLine.    #
# Whether the file was recorded in TISNATIVE or DIANE format #
# will automatically be detected by the DataTap-10. The UDP  #
# datagrams will be sent out in whatever format is specified #
# with the RAW_SEND_REC_FORMAT and DEC_SEND_REC_FORMAT       #
# parameters.                                                #
#                                                            #
# The PLAYBACK_IRIG parameter is used to indicate the exact  #
# point in the file specified as an IRIG time at which to    #
# begin playback. A value of 001:00:00:00 or 000:xx:xx:xx    #
# indicates to begin playback at the very start of the       #
# recording. For any other value, the DataTap-10 will        #
# search through all segments and frames of the recording    #
# using a a fast search algorithm until an IRIG time is      #
# found which is later than the specified time. If a later   #
# time is never found, playback will stop. The search        #
# typically only takes a few seconds regardless of the       #
# number of file segments which exist for the recording.     #
#                                                            #
# NOTE: In RXRAW mode, no data will be transmitted onto the  #
# ASCB bus. The Consume ESCAPE registry belonging to the     #
# same aircraft load used for the recording MUST be          #
# programmed into the DataTap-10 or the data will not be     #
# valid.                                                     #
#                                                            #
# In TXRAW mode, all data in the recording is transmitted    #
# onto the ASCB bus by the DataTap-10 acting as every NIC in #
# the system. Note that no other modules that transmit       #
# should be connected to the ASCB bus when TXRAW mode is     #
# enabled. The NIC registry belonging to the same aircraft   #
# load used for the recording MUST programmed into the       #
# DataTap-10 or the transmitted data will not be valid. If   #
# receive processing is desired (i.e. UDP output or viewing  #
# over Telnet), then the correct Consume ESCAPE registry     #
# must be programmed as well. Note that TXRAW mode is not    #
# valid for ASCB-A/B/C.                                      #
```

```
#                                                             #
# Defaults:                                                   #
#                                                             #
# PLAYBACK_MODE         OFF                                   #
# PLAYBACK_FILE         ASCB_RAW.01                           #
# PLAYBACK_IRIG         001:00:00:00                          #
#                                                             #
# Valid Values:                                               #
#                                                             #
# PLAYBACK_MODE         OFF | RXRAW | TXRAW                   #
# PLAYBACK_IRIG         001:00:00:00 to 366:23:59:59          #
#                                                             #
###############################################################

PLAYBACK_MODE             OFF
PLAYBACK_FILE             ASCB_RAW.01
PLAYBACK_IRIG             001:00:00:00


###############################################################
# Configure the default settings for the ARINC-717 decoder.   #
# These can be changed dynamically through Telnet,            #
# FlightLine, or UDP command packets.                         #
#                                                             #
# Defaults:                                                   #
#                                                             #
# A717_EN_DEF          FALSE                                  #
# A717_WORD_RATE_DEF   512                                    #
#                                                             #
# Valid Values:                                               #
#                                                             #
# A717_EN_DEF          TRUE | FALSE                           #
# A717_WORD_RATE_DEF   64   | 128   | 256   | 512   |         #
#                      1024 | 2048 | 4096 | 8192              #
#                                                             #
###############################################################

A717_EN_DEF               FALSE
A717_WORD_RATE_DEF        512


###############################################################
# Indicate the synchronization source for timing information #
# sent in any output packets. If IEEE-1588v2 is specified,    #
# the time will be converted to IRIG format for any ICS-      #
# defined output packet headers.                              #
#                                                             #
# Default:       IRIGB                                        #
# Valid Values:  IRIGB | IEEE1588v2                           #
#                                                             #
###############################################################
```

```
USE_TIMING_SOURCE          IRIGB


##############################################################
# Indicate whether the IRIG input has a DC offset. IRIG      #
# source outputs which are capacitively-coupled rather than  #
# transformer coupled may be pulled up to 2.5V by the input  #
# buffer to the ADC of the DataTap-10. Though it is          #
# recommended to use a small audio transformer at the input  #
# to the DataTap-10 in this case, the offset can             #
# alternatively be subtracted out by the DataTap-10. If the  #
# offset is known to exist, this value should be set to      #
# TRUE. If the offset is known NOT to exist, this value      #
# should be set to FALSE. If the value is set to AUTO, the   #
# DataTap-10 will switch between accounting and not          #
# accounting for the offset every 3 seconds until the IRIG   #
# signal is locked onto.                                     #
#                                                            #
# Default:       AUTO                                        #
# Valid Values:  TRUE | FALSE | AUTO                         #
#                                                            #
##############################################################

IRIG_HAS_DC_OFFSET        AUTO


##############################################################
# Configure the PTP announce timeout. This is expressed in   #
# terms of announce message intervals. If an announce        #
# message is not received within the specified number of     #
# announce message intervals, the PTP state machine will     #
# return to the listening state.                             #
#                                                            #
# Default:       3 (per the IEEE-1588v2 spec)                #
# Valid Values:  2 to 255                                    #
#                                                            #
##############################################################

PTP_ANNOUNCE_TIMEOUT      3


##############################################################
# Configure the PTP time zone offset in hours. Because the   #
# PTP time represents UTC, this field allows the time to be  #
# corrected for the current time zone.                       #
#                                                            #
# Default:       0                                           #
# Valid Values:  -12 to 12                                   #
#                                                            #
##############################################################
```

```
PTP_TIMEZONE_OFFSET       0


##############################################################
# Indicate the number of UDP frame sync messages which are   #
# to be sent out of the DataTap-10 at the start of each ASCB #
# frame. Having multiple sync messages sent out with         #
# different UDP destination ports can be useful on a host    #
# where separate programs or instances are used to create    #
# the data for each ASCB bus when doing ASCB transmission.   #
# FlightLine, for example, requires 2 sync messages to be    #
# sent out because it uses 2 separate instances of a C++     #
# class to perform the transmission. Note that the UDP       #
# destination port of the first sync message is              #
# UDP_HOST_PORT_SYNC and it increments by 1 from there for   #
# each succeeding sync message in the same ASCB frame. Also  #
# note that UDP_SYNC_EN_DEF must be set to TRUE for any UDP  #
# sync messages to be sent out.                              #
#                                                            #
# This parameter is ignored when the ASCB version is set     #
# to ASCB-A/B/C.                                             #
#                                                            #
# Default:       2                                           #
# Valid Values:  0 to 3                                      #
#                                                            #
##############################################################

NUM_UDP_SYNC_MSGS         2


##############################################################
# Define whether the discrete outputs of the DataTap-10 are  #
# used as general purpose outputs, where the values can be   #
# set via UDP command packets, or whether they are used to   #
# output health and status information. If set to TRUE, the  #
# discrete outputs will be used for the following:           #
#                                                            #
#    - Discrete Output 1 will output a one pulse per second  #
#      (PPS) based on the IRIG or IEEE-1588 synchronization. #
#    - Discrete Output 2 will indicate the health status of  #
#      the DataTap-10,                                       #
#    - Discrete Output 3 will go active when the microSD     #
#      card is getting low on recording space.               #
#    - Discrete Output 4 will indicate whether recording is  #
#      active.                                               #
#                                                            #
# Default:       FALSE                                       #
# Valid Values:  TRUE | FALSE                                #
#                                                            #
##############################################################
```

```
OUTPUT_HEALTH_INFO        FALSE


##############################################################
# Define the number of minutes when a warning should be      #
# issued via the discrete outputs (if configured) and/or the #
# status LED (if configured) due to limited recording space. #
# The warning can also be viewed over Telnet. This parameter #
# applies to recording to Compact Flash or microSD depending #
# on the setting of USE_COMPACT_FLASH.                       #
#                                                            #
# Default:       10                                          #
# Valid Values:  0 to 480                                    #
#                                                            #
##############################################################

SD_MEMORY_RECORD_WARNING  10


##############################################################
# Define the DataTap-10 device number. This value does not   #
# affect any ICS-defined output packet headers.              #
#                                                            #
# Default:       0                                           #
# Valid Values:  0 to 31                                     #
#                                                            #
##############################################################

DEVICE_NUM                0


##############################################################
# Define the UDP output and recording format. DIANE61 and    #
# DIANE62 are custom formats, so typically TISNATIVE is used #
# here.                                                      #
#                                                            #
# Note that "FULL_" may be used in place of "RAW_" and       #
# "PARTIAL_" may be used in place of "DEC_" in the parameter #
# names for legacy support.                                  #
#                                                            #
# These parameters are ignored when the ASCB version is set  #
# to ASCB-A/B/C.                                             #
#                                                            #
# Defaults:                                                  #
#                                                            #
# RAW_SEND_REC_FORMAT   TISNATIVE                            #
# DEC_SEND_REC_FORMAT   TISNATIVE                            #
#                                                            #
# Valid Values:                                              #
#                                                            #
```

```
# RAW_SEND_REC_FORMAT   TISNATIVE | DIANE61             #
# DEC_SEND_REC_FORMAT   TISNATIVE | DIANE62             #
#                                                       #
#########################################################

RAW_SEND_REC_FORMAT       TISNATIVE
DEC_SEND_REC_FORMAT       TISNATIVE


#########################################################
# Define the send and record rate in Hz for each UDP decoded #
# datagram.                                             #
#                                                       #
# Note that "PARTIAL_" may be used in place of "DEC_" in the #
# parameter names for legacy support.                   #
#                                                       #
# These parameters are ignored when the ASCB version is set #
# to ASCB-A/B/C.                                        #
#                                                       #
# Default:       80                                     #
# Valid Values:  80 | 40 | 20 | 10 | 5 | 2 | 1          #
#                                                       #
#########################################################

DEC_SEND_1_RATE           80
DEC_SEND_2_RATE           80
DEC_SEND_3_RATE           80
DEC_SEND_4_RATE           80
DEC_SEND_5_RATE           80
DEC_SEND_6_RATE           80
DEC_SEND_7_RATE           80
DEC_SEND_8_RATE           80
DEC_SEND_9_RATE           80
DEC_SEND_10_RATE          80
DEC_SEND_11_RATE          80
DEC_SEND_12_RATE          80
DEC_SEND_13_RATE          80
DEC_SEND_14_RATE          80
DEC_SEND_15_RATE          80
DEC_SEND_16_RATE          80
DEC_SEND_17_RATE          80
DEC_SEND_18_RATE          80
DEC_SEND_19_RATE          80
DEC_SEND_20_RATE          80
DEC_SEND_21_RATE          80
DEC_SEND_22_RATE          80
DEC_SEND_23_RATE          80
DEC_SEND_24_RATE          80
DEC_SEND_25_RATE          80
DEC_SEND_26_RATE          80
DEC_SEND_27_RATE          80
```

```
DEC_SEND_28_RATE          80
DEC_SEND_29_RATE          80
DEC_SEND_30_RATE          80
DEC_SEND_31_RATE          80
DEC_SEND_32_RATE          80


##############################################################
# Define the string embedded in the timing messages sent out #
# by the DataTap-10 when it is configured for transmission   #
# as a timing NIC. Timing NICs are defined as having a NIC   #
# ID of 1, 2, 33, or 34. All timing NICs send out timing     #
# messages at the start of each ASCB frame.                  #
#                                                            #
# Recent releases of the Honeywell MAU software verify the   #
# string sent in the timing messages at power-up as part of  #
# its Configuration Management System (CMS). If the string   #
# from each timing message does match the expected value,    #
# the MAU NIC(s) will go into an idle state.                 #
#                                                            #
# The string in each received timing message is available    #
# in quotes on the NIC Status Page over Telnet. If an MAU    #
# NIC is stuck in the idle state due to a CMS failure,       #
# either power-up the DataTap-10 after the MAU cabinets are  #
# already running or temporarily put the DataTap-10 into a   #
# receive-only mode such that the expected string can be     #
# retrieved over Telnet.                                     #
#                                                            #
# Note that the quotes around the string are not considered  #
# a part of the actual message string and may be omitted.    #
#                                                            #
# Default:                                                   #
#                                                            #
# TMG_MSG_STRING   "ICS DataTap-10"                          #
#                                                            #
# Valid Values:                                              #
#                                                            #
# TMG_MSG_STRING   <any string up to 35 characters>          #
#                                                            #
##############################################################

TMG_MSG_STRING            "ICS DataTap-10"


##############################################################
# Indicate whether record append mode is enabled. When       #
# enabled and a new recording is started within the same     #
# power cycle as a previous recording, the previous          #
# recording file(s) will be opened and appended to with the  #
# new recording data. This can be used with or without       #
# circular recording mode enabled (see option CIRC_REC_MODE  #
```

```
# below). This setting applies to ASCB, A429, and A717       #
# recording.                                                 #
#                                                            #
# Note that if ASCB recording is enabled and recording is    #
# restarted via Telnet/UDP/FlightLine with a different type  #
# (raw or decoded) than the previous recording, a new ASCB   #
# recording session will be started and the previous         #
# recording will NOT be appended to. If the media card is    #
# removed and then reinserted, new recording session(s) will #
# always be started.                                         #
#                                                            #
# This parameter has superseded ASCB_REC_APPEND because it   #
# applies to all recording types, but either parameter can   #
# can used.                                                  #
#                                                            #
# Default:        FALSE                                      #
# Valid Values:   TRUE | FALSE                               #
#                                                            #
##############################################################

REC_APPEND_MODE            FALSE


##############################################################
# Indicate whether circular record mode is enabled. Circular #
# recording mode means that whenever a new recording file or #
# file segment is opened and less than 2GB remains on the    #
# media card, the DataTap-10 will search for the oldest      #
# recording file on the card and overwrite it. The oldest    #
# file is determined using the filenames residing in the     #
# base directory.                                            #
#                                                            #
# When circular recording mode is enabled, filenames are     #
# created in the form of <base>_m.n, where m is a session    #
# number between 1 and 999 and n is the standard             #
# incrementing extension per 2GB file segment (.01, .02,     #
# etc.). The session number m is incremented each time a     #
# new recording is started, unless a file is to be appended  #
# based on the setting of REC_APPEND_MODE. The DataTap-10    #
# determines the new session number by first searching the   #
# media for the current highest session number and then      #
# incrementing it for use in the new filename. Files not in  #
# the form of _m.n are ignored during the search.            #
#                                                            #
# Note that the card must first be filled with empty 2GB     #
# files when in Circular Recording Mode, otherwise an error  #
# may be generated during recording. The fill can be         #
# executed using the command "quote site fill" in the        #
# DataTap-10 FTP server. A fill will automatically be        #
# executed on power-up if Circular Recording Mode is enabled #
# and no valid _m.n files exist on the card.                 #
```

```
#                                                             #
# Note that REC_FILENAME_FORMAT is still valid when in        #
# circular recording mode. If set to TRUE, the timestamp      #
# will be added as a suffix to <base> as described in         #
# REC_FILENAME_FORMAT. When circular recording mode is        #
# disabled, filenames will be created in the legacy           #
# DataTap-10 format as described in the xxx_REC_FILE and      #
# REC_FILENAME_FORMAT options.                                #
#                                                             #
# This parameter has superseded ASCB_CIRC_REC_MODE            #
# because it applies to all recording types, but either       #
# parameter can be used.                                      #
#                                                             #
# Default:        FALSE                                       #
# Valid Values:   TRUE | FALSE                                #
#                                                             #
###############################################################

CIRC_REC_MODE                FALSE


###############################################################
# Indicate whether a recording set to occur at power-up       #
# (i.e. automatic recording) should wait for ASCB data to be  #
# present on the bus before starting to record. Automatic     #
# recording is enabled by setting at least one of the record  #
# enables to TRUE as well as setting AUTO_REC_ENABLE to       #
# TRUE.                                                       #
#                                                             #
# Note that this only applies to automatic recording and      #
# has no effect if a recording is started manually via the    #
# the discrete inputs, Telnet, UDP, or FlightLine.            #
#                                                             #
# This parameter has superseded ASCB_REC_WAIT_DATA because    #
# it applies to all enabled recording types (though it only   #
# monitors for ASCB data present on the bus), but either      #
# parameter can be used.                                      #
#                                                             #
# Default:        FALSE                                       #
# Valid Values:   TRUE | FALSE                                #
#                                                             #
###############################################################

AUTO_REC_WAIT_ASCB           FALSE


###############################################################
# Indicate the number of seconds to delay from power-up       #
# before starting an automatic recording (i.e. RAW_REC_EN     #
# or DEC_REC_EN set to TRUE). Once the delay expires,         #
# automatic recording will begin UNLESS:                      #
```

```
#                                                             #
#   - ASCB_REC_WAIT_DATA is set to TRUE and there is no data #
#     on the bus, or                                          #
#   - A discrete input is set for manual recording control    #
#     and is low.                                             #
#                                                             #
# Note that this option is ignored when automatic recording  #
# is not enabled.                                             #
#                                                             #
# Default:       0                                            #
# Valid Values:  0 to 65535                                   #
#                                                             #
###############################################################

AUTO_REC_DELAY_SEC        0


###############################################################
# Indicate whether automatic recording is enabled, meaning   #
# recording will begin automatically at power-up once all    #
# optional conditions have been met. This parameter was      #
# added to allow manual recording control via the discrete   #
# inputs without requiring automatic recording to be enabled #
# as well. The default value of this parameter is set to     #
# TRUE to maintain compatibility with older configuration    #
# files by allowing it to be omitted. Note that if set to    #
# TRUE and none of the individual recording enables are set  #
# to TRUE, no automatic recording will begin at power-up.    #
#                                                             #
# Default:       TRUE                                         #
# Valid Values:  TRUE | FALSE                                 #
#                                                             #
###############################################################

AUTO_REC_ENABLE           TRUE


###############################################################
# Indicate whether active recordings are automatically       #
# stopped when no ASCB data is present. When set to TRUE and #
# no ASCB data is present, any active ASCB, A429, and A717   #
# recordings will all be stopped.                            #
#                                                             #
# Note that this parameter cannot be used simultaneously     #
# with level-sensitive discrete input recording control or   #
# a configuration error will be generated.                   #
#                                                             #
# Default:       FALSE                                        #
# Valid Values:  TRUE | FALSE                                 #
#                                                             #
###############################################################
```

```
REC_STOP_NO_ASCB          FALSE


##############################################################
# Indicate the configuration of each of the four discrete    #
# inputs. Note that using a discrete input for recording     #
# control is termed as manual recording control. These       #
# parameters can be set to the following values:             #
#                                                            #
#        NONE:  This indicates a discrete input is           #
#               unconfigured and can only be used as a       #
#               general purpose input whose value can be     #
#               read via Telnet or in the UDP status         #
#               output messages.                             #
#                                                            #
# REC_CTL_LVL:  This indicates a discrete input is used for  #
#               level-sensitive recording control. A high    #
#               level on the discrete input indicates        #
#               recording is enabled while a low level       #
#               indicates recording is disabled. Note that   #
#               recording control using Telnet, UDP, or      #
#               FlightLine cannot be used when this mode     #
#               is enabled.                                  #
#                                                            #
# REC_CTL_EDGE: This indicates a discrete input is used for  #
#               edge-triggered recording control. A rising   #
#               edge detected on the discrete input will     #
#               start recording and a falling edge will stop #
#               recording. Using edge-triggered control      #
#               allows recording control using Telnet, UDP,  #
#               or FlightLine to be used as well.            #
#                                                            #
# Note that only one discrete input can be configured for    #
# manual control. If multiple are configured, only the first #
# one encountered in this file will be used and a            #
# configuration error will be generated on any others.       #
#                                                            #
# When recording is started via the discrete inputs, the     #
# streams to be recorded are determined by the current       #
# streams enabled for recording (i.e. the record select). At #
# power-up, the record select is determined by the values of #
# RAW_REC_EN, DEC_REC_EN, A429_REC_EN, and A717_REC_EN.      #
# After power-up, the record select can then be changed      #
# through Telnet, UDP, and/or FlightLine.                    #
#                                                            #
# Default:       NONE                                        #
# Valid Values:  NONE | REC_CTL_LVL | REC_CTL_EDGE           #
#                                                            #
##############################################################
```

```
DIN1_CONFIG                 NONE
DIN2_CONFIG                 NONE
DIN3_CONFIG                 NONE
DIN4_CONFIG                 NONE


##############################################################
# Indicate the configuration of each ARINC-429 channel at    #
# power-up. These values can be changed over Telnet, UDP, or #
# FlightLine. A value of "HIGH" indicates the channel is     #
# high-speed, "LOW" indicates the channel is low-speed, and  #
# "OFF" indicates the channel is not enabled on power-up.    #
#                                                            #
# Default:       OFF                                         #
# Valid Values:  OFF | HIGH | LOW                            #
#                                                            #
##############################################################

A429_RX1_CONFIG             OFF
A429_RX2_CONFIG             OFF
A429_RX3_CONFIG             OFF
A429_RX4_CONFIG             OFF


##############################################################
# Indicate the input for the ARINC-717 decoder. The input    #
# can be any discrete or ARINC-429 input. When using a       #
# discrete input, the maximum supported rate is 512 words    #
# per second due to the limited speed of the opto-couplers   #
# used in the DataTap-10 hardware. When using an ARINC-429   #
# input, any rate between 64 and 8192 words per second is    #
# supported.                                                 #
#                                                            #
# Default:       DIN1                                        #
#                                                            #
# Valid Values:  DIN1 | DIN2 | DIN3 | DIN4 |                 #
#                A429_RX1 | A429_RX2 | A429_RX3 | A429_RX4   #
#                                                            #
##############################################################

A717_INPUT                  DIN1
```