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

XMC-A825-16 ARINC-825 16 Channel XMC / CANFlight-16 User's Manual REV C

Driven by CAN Aerospace

ARINC 825

CAN AVIATION ALLIANCE

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

XMC-A825-16 ARINC-825 16 Channel XMC / CANFlight-16 User's Manual REV C

Table of Contents

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

XMC-A825-16 ARINC-825 16 Channel XMC / CANFlight-16 User's Manual REV C

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

XMC-A825-16 ARINC-825 16 Channel XMC / CANFlight-16 User's Manual REV C

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

XMC-A825-16 ARINC-825 16 Channel XMC / CANFlight-16 User's Manual REV C

1 Introduction

The XMC-A825-16 interface system offers 16 electrically isolated CAN 2.0B interfaces according to ISO 11898, an IRIG-B time code input and a 10/100/1000 BaseT Ethernet interface. To minimize CPU load on host computers, the XMC-A825-16 uses an onboard Xilinx Artix 7 FPGA with Five 128 MHz Microblaze processors and 24 MByte SRAM to process and store CAN messages. The CAN bus interfaces are accessible via a VHD-68 front panel connector as well as two 40-pin rectangular on-board connectors behind the VHD-68 connector.

Figure 1-1 XMC-A825-16 Hardware

The XMC-A825-16 features are:

- Laptop-ready, 5 Processor Based Embedded Realtime Control System for Ground and Airborne Use
- Sixteen (16) isolated, fully independent Controller Area Network (ISO 11898), ARINC825 and CANaerospace protocol compliant interfaces
- Integrated µSDHC card-based Flight Data Recording Option available
- IRIG-B Time Code Input for High Precision Data Synchronization
- Auto-negotiating 10/100/1000 BaseT Ethernet interface with CANaerospace over Ethernet (CoE) and ARINC825 over Ethernet (A825oE) protocol and Application Programmer Interface (API)
- IEEE 802.11 Wireless LAN Option with CANaerospace over Wireless (CoW) and ARINC825 over Wireless (A825oW) protocol available
- Frontpanel Activity LEDs for CAN and Ethernet
- Power Supply 5V through XMC interface or through ICS Stand Alone Aircraft power supply. Aircraft Power (9-36VDC)
- Mechanical Dimensions 149mm L x 74mm W x 47mm H
- CAN/ARINC825/CANaerospace toolbox for Linux, MacOS and Windows
- Custom Software and FPGA firmware Options available on request

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

XMC-A825-16 ARINC-825 16 Channel XMC / CANFlight-16 User's Manual REV C

The XMC-A825-16 is a specific purpose embedded computer system that utilizes its processing power to relief external computer systems from the tasks of transmitting, receiving, buffering and pre/postprocessing low, medium or high-speed CAN, CANaerospace and ARINC 825 messages. It can handle up to 100% bus load at the maximum CAN data rate of 1MBit/s on both channels without data loss. The driver software provides an easy-to-handle function call interface for CAN bus message transmission and reception including support for the CANaerospace and ARINC 825 higher layer protocols. The XMC-A825-16 software consists of host drivers for various operating systems and platforms, sample “C” source code and the XCT toolbox connected to the XMC-A825-16 via Ethernet/UDP/IP. The Illustration below shows the XMC-A825-16 block diagram.

The XMC-A825-16 CAN 2.0B interfaces are implemented with licensed Bosch M_CAN controller IP cores to ensure compatibility with the Bosch CAN standard and to allow precise hardware timing and control over the transmission and reception of CAN/ARINC825/CANaerospace messages. The Xilinx FPGA and the XMC-A825-16 firmware provide local buffering and 60ns time stamp resolution for all CAN messages and implement ARINC825/CANaerospace specific protocol functions. High precision time synchronization of CAN messages is accomplished through an IRIG-B time code input providing 1µs resolution. An integrated µSDHC interface is used for data acquisition storage, system configuration information and firmware upgrades.

XMC-A825-16 can be integrated into a rugged aluminum box and powered from 9-36 VDC allowing it to run from standard 14V or 28V DC aircraft power buses according to the EN2282 specification. In this configuration it is called a CANFlight-16 The power input lines are protected against transient overvoltage and electromagnetic interference. The total power consumption of a XMC-A825-16 unit is less than 5Watts. The CAN and Ethernet interfaces are serviced by different processors so that all interfaces may be used at the same time without any loss of data. XMC-A825-16 units may be connected to host computers using CANaerospace/ARINC825 and the auto-negotiating 10/100/1000 BaseT Ethernet interface with the CAN over Ethernet protocol standard developed by Stock Flight Systems. The Wireless LAN (IEEE 802.11) option for CANFlight-16 version provides an integrated DHCP server and a web-based configuration interface. The µSDHC card slot accepts cards with sizes up to 128 GBytes.

XMC-A825-16 units can work as either standalone systems or as intelligent nodes within simple or complex CAN, CANaerospace and ARINC825 networks, They are interoperable with the Dual Channel CANFlight and the 4-channel PowerNECS in any combinations on the same network. The interface between XMC-A825-16 and host computer applications via Ethernet/UPD/IP, either wired or wireless, is realized through a corresponding Application Programming Interface (API) for a variety of operating systems including Linux, Solaris, MacOS, Windows and VxWorks.

XMC-A825-16 ARINC-825 16 Channel XMC / CANFlight-16 User's Manual REV C

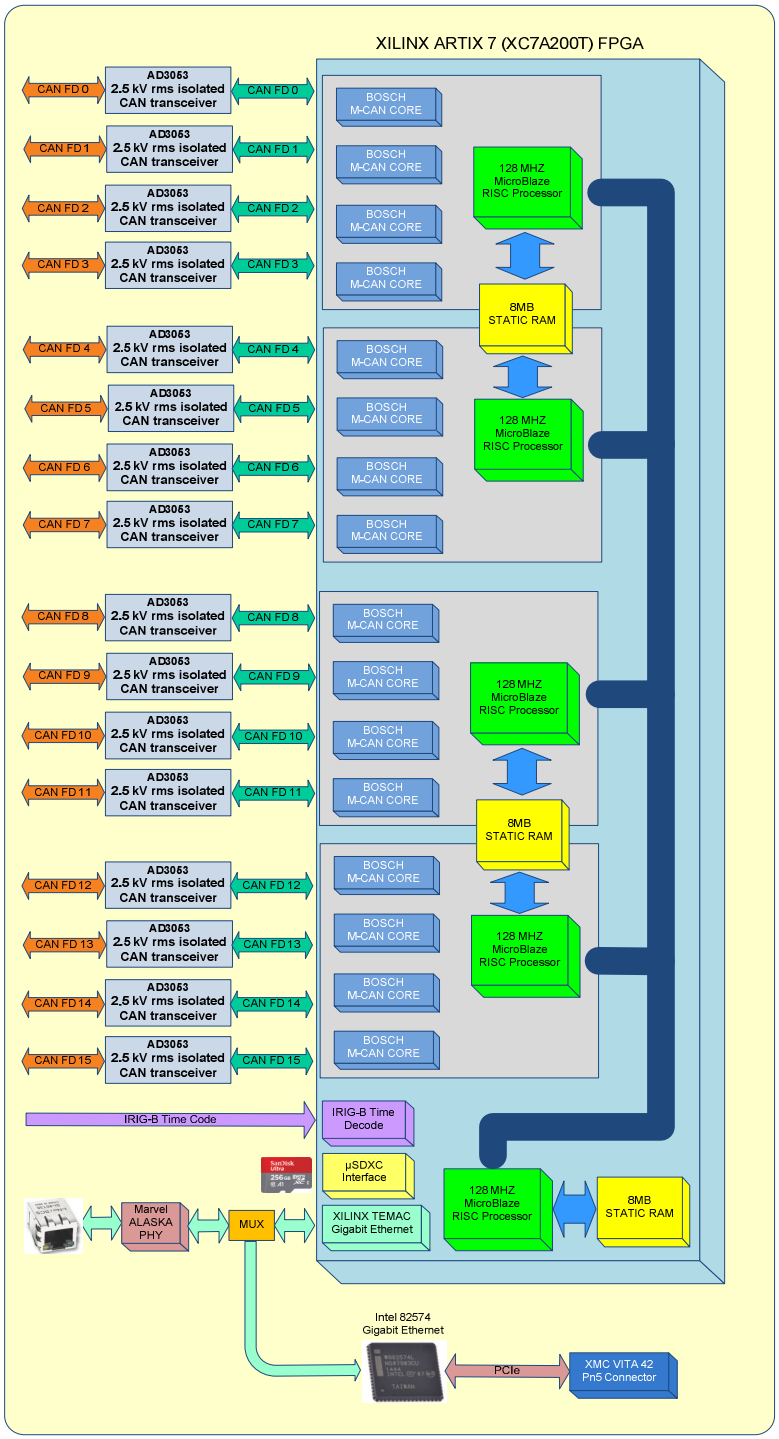


Figure 1-2 XMC-A825-16 Block Diagram

2 Environmental Specifications

- Passively Cooled. No moving parts, -40°C to +70°C (+85°C with airflow)
- Weight: 100 grams or less in standard configuration (XMC Card Only)
- Power Consumption < 10W all 16-channels driven
- 155 mm L x 75mm W x 13.4mm H (Overall Dimensions XMC Card)
- Operating Altitude Less than 25.000ft (7600m)
- Humidity 95% or less, non-condensing
- 5V or 12V power from VPWR Pins of XMC Card Connector (<10 watts)
- Vibration, surpasses requirements of DO-160 Category S

3 Front Panel Connectors and Indicators

The XMC-A825-16 front panel is shown in ***Figure 3-1*** the usage of the various connectors and LEDs is described in Table. Note that the CAN channels are numbered CH1-CH16 for indicators on the Optional TP9060 I/O Panel and connectors but are referred to as channels 0 through 15 for all software functions including the configuration file (section 8 and 10).

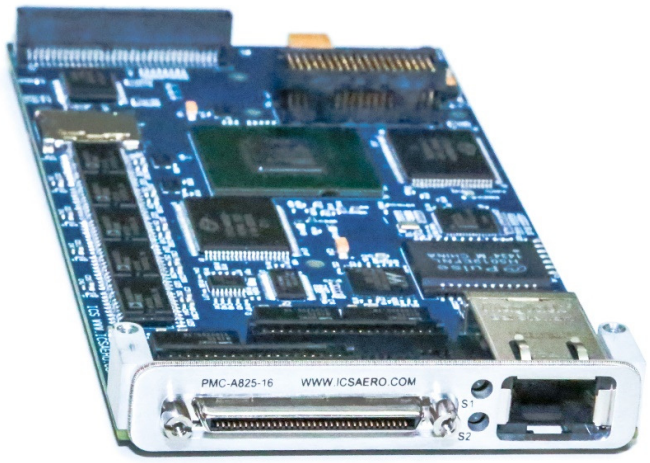


Figure 3-1 XMC-A825-16 Front Panel

XMC-A825-16 Front Panel Element	Description
ETHERNET Connector	Standard RJ-45 connector used for the interface between XMC-A825-16 and other computer systems using 10/100/1000 BaseT Ethernet. The XMC-A825-16 Ethernet supports auto-negotiating of the bus speed as well as support for ICMP echo requests so that XMC-A825-16 systems respond to “ping” commands over the network. This connector is only used when the on-board Intel Gigabit Ethernet Controller is not enabled. To use this interface, the option PCI=0 must be set in the configuration file “XMC_A825.CFG” on the micro SD card installed. See the Section entitled XMC-A825-16 Configuration
Status LED S1	Bicolor LED which is illuminated in red during the start of the XMC-A825-16 firmware or in case of an internal system failure. This LED flashes green during normal operation (with 90% on-time versus 10% off-time).
Status LED S2	Bicolor LED which is illuminated in red during the start of the XMC-A825-16 firmware or in case of an internal system failure. This LED flashes green during normal operation (with 90% on-time versus 10% off-time).

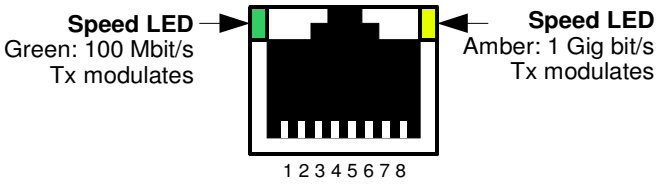
Table 1: XMC-A825-16 Front Panel Elements

The CAN activity LEDs located on the front panel (TX, RX) indicate if a CAN channel transmits and/or receives CAN messages. Every CAN channels has a dedicated RX/TX activity LED which flashes once for every transmitted/received message or continuously at a rate of 2Hz in case of a steady message flow.

If the RX activity LED of a particular CAN channel stays dark in an active network, the physical interface correctness of all network nodes including the used baud rates, sample points and termination resistors should be verified.

3.1 Gigabit Ethernet Connection

The XMC-A825-16 ETHERNET Connector has a pinout according to the established RJ-45 standard as shown in Figure Illustration. This connector also contains two Status LEDS which indicate the negotiated Ethernet data rate and interface transmit activity.



This connector is only used when the on-board Intel Gigabit Ethernet Controller is not enabled. To use this interface, the option PCI=0 must be set in the configuration file “XMC_A825.CFG” on the micro SD card installed. See the Section entitled XMC-A825-16 Configuration

Figure 3-2 Gigabit Ethernet Connector layout

Table shows the corresponding Ethernet signal assignment for the RJ-45 connector.

Pin	Signal Name	Signal Description
1	TX+	Transmit Data +
2	TX-	Transmit Data -
3	RX+	Receive Data +
4	N.C.	Not Connected
5	N.C.	Not Connected
6	RX-	Receive Data -
7	N.C.	Not Connected
8	N.C.	Not Connected

Table 2: RJ-45 Connector Signal Definition

3.2 CAN I/O Connector (VHD-68)

Name	VHD 68 Pin	Description	Notes
CAN HI CH0	1	CAN 0 High	No on board Terminations
CAN LO CH0	35	CAN 0 Low	No on board Terminations
CAN HI CH1	2	CAN 1 High	No on board Terminations
CAN LO CH1	36	CAN 1 Low	No on board Terminations
CAN HI CH2	3	CAN 2 High	No on board Terminations
CAN LO CH2	37	CAN 2 Low	No on board Terminations
CAN HI CH3	4	CAN 3 High	No on board Terminations
CAN LO CH3	38	CAN 3 Low	No on board Terminations
CAN HI CH4	5	CAN 4 High	No on board Terminations
CAN LO CH4	39	CAN 4 Low	No on board Terminations
CAN HI CH5	6	CAN 5 High	No on board Terminations
CAN LO CH5	40	CAN 5 Low	No on board Terminations
CAN HI CH6	7	CAN 6 High	No on board Terminations
CAN LO CH6	41	CAN 6 Low	No on board Terminations
CAN HI CH7	8	CAN 7 High	No on board Terminations
CAN LO CH7	42	CAN 7 Low	No on board Terminations
CAN HI CH8	26	CAN 8 High	No on board Terminations
CAN LO CH8	60	CAN 8 Low	No on board Terminations
CAN HI CH9	27	CAN 9 High	No on board Terminations
CAN LO CH9	61	CAN 9 Low	No on board Terminations
CAN HI CH10	28	CAN 10 High	No on board Terminations
CAN LO CH10	62	CAN 10 Low	No on board Terminations
CAN HI CH11	29	CAN 11 High	No on board Terminations
CAN LO CH11	63	CAN 11 Low	No on board Terminations
CAN HI CH12	31	CAN 12 High	No on board Terminations
CAN LO CH12	65	CAN 12 Low	No on board Terminations
CAN HI CH13	32	CAN 13 High	No on board Terminations
CAN LO CH13	66	CAN 13 Low	No on board Terminations
CAN HI CH14	33	CAN 12 High	No on board Terminations
CAN LO CH14	67	CAN 12 Low	No on board Terminations
CAN HI CH15	34	CAN 13 High	No on board Terminations
CAN LO CH15	68	CAN 13 Low	No on board Terminations
CAN0 GND	9	CAN Bus 0 GND	ISO CAN GND
CAN1 GND	43	CAN Bus 1 GND	ISO CAN GND
CAN2 GND	10	CAN Bus 2 GND	ISO CAN GND
CAN3 GND	44	CAN Bus 3 GND	ISO CAN GND
CAN4 GND	11	CAN Bus 4 GND	ISO CAN GND
CAN5 GND	45	CAN Bus 5 GND	ISO CAN GND
CAN6 GND	12	CAN Bus 6 GND	ISO CAN GND

CAN7 GND	46	CAN Bus 7 GND	ISO CAN GND
CAN8 GND	22	CAN Bus 8 GND	ISO CAN GND
CAN9 GND	56	CAN Bus 9 GND	ISO CAN GND
CAN10 GND	23	CAN Bus 10 GND	ISO CAN GND
CAN11 GND	57	CAN Bus 11 GND	ISO CAN GND
CAN12 GND	24	CAN Bus12 GND	ISO CAN GND
CAN13 GND	58	CAN Bus 13 GND	ISO CAN GND
CAN14 GND	25	CAN Bus 14 GND	ISO CAN GND
CAN15 GND	59	CAN Bus 15 GND	ISO CAN GND
RS232-TX	13	TX Data Output	RS232 Debug Port (Reference Digital Ground)
RS232-RX	47	RX DATA Input	RS232 Debug Port (Reference Digital Ground)
IRIG IN +	15	IRIG Time Code Input +	IRIG B Timecode Input +
IRIG IN -	49	IRIG Time Code Input -	IRIG B Timecode Input -
GND	21	Digital Ground	Ground Return
GND	30	Digital Ground	Ground Return
GND	55	Digital Ground	Ground Return
GND	64	Digital Ground	Ground Return
+3.3V	17	3.3V power from Board	This is reserved for use with ICS designed I/O panel.
+3.3V	18	3.3V power from Board	This is reserved for use with ICS designed I/O panel.
+3.3V	50	3.3V power from Board	This is reserved for use with ICS designed I/O panel.
+3.3V	51	3.3V power from Board	This is reserved for use with ICS designed I/O panel.
+3.3V	52	3.3V power from Board	This is reserved for use with ICS designed I/O panel.
LED I/O	16	LED SREG RST N	This is reserved for use with ICS designed I/O panel.
LED I/O	19	LED SREG OE N	This is reserved for use with ICS designed I/O panel.
LED I/O	20	LED SREG SDAT	This is reserved for use with ICS designed I/O panel.
LED I/O	53	LED SREG LCLK	This is reserved for use with ICS designed I/O panel.
LED I/O	54	LED SREG SCLK	This is reserved for use with ICS designed I/O panel.
SPARE 0	48	Do Not Connect	Spare I/O Pins to Expander Board.
SPARE 1	14	Do Not Connect	Spare I/O Pins to Expander Board.

Table 3 VHD-68 Front Panel I/O Pinout

4 XMC-A825-16 ON Board I/O Connectors

4.1 On Board µSDXC card connector

XMC-A825-16 offers a µSD card slot that supports FAT-32 formatted µSD and µSDHC cards. All standard µSD and µSDHC cards with sizes ranging from 1 GB to 128GB may be used with XMC-A825-16 if they support the Serial Peripheral Interface (SPI) mode and are capable of operating under a sustained SPI data rate of 25 Mhz.

The standard pinout of µSDHC cards is shown in Figure Illustration, the corresponding signal assignment in Table Table. Figure Illustration shows a standard µSDHC card as an example. All standard µSD, µSDHC, and µSDXC cards with sizes ranging from 1GB to 128 GB maybe used with XMC-A825-16. The µSD card interface is fully compatible with the µSD/HC specification of the SD Card Organization (www.sdcard.org).

The MicroSD card interface is used to configure the XMC-A825-16 during startup using a dedicated configuration file (see section 8) and to perform firmware upgrades (see section 9). Options for CAN data recording on µSDHC card are available on request.

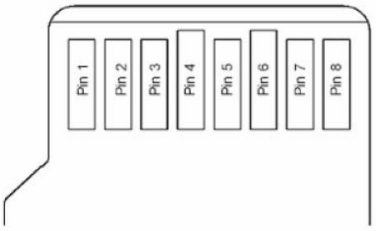


Figure 4-2 µSDHC Pinout

Figure 4-1 Typical µSDHC Card

Pin	Signal Name	Signal Description
1	RSD	Reserved
2	CS*	Chip Select (active low)
3	DI	Data In
4	VDD	Supply Voltage
5	SCK	System Clock
6	VSS	Supply Voltage Ground
7	DO	Data Out
8	RSD	Reserved

Table 4: µSDHC Connector Signal Definition

4.2 On Board CAN I/O P1

Name	P1 PIN	Description	Notes
CAN HI CH0	1	CAN 0 High	No on board Terminations
CAN LO CH0	3	CAN 0 Low	No on board Terminations
CAN HI CH1	5	CAN 1 High	No on board Terminations
CAN LO CH1	7	CAN 1 Low	No on board Terminations
CAN HI CH2	9	CAN 2 High	No on board Terminations
CAN LO CH2	11	CAN 2 Low	No on board Terminations
CAN HI CH3	13	CAN 3 High	No on board Terminations
CAN LO CH3	15	CAN 3 Low	No on board Terminations
CAN HI CH4	17	CAN 4 High	No on board Terminations
CAN LO CH4	19	CAN 4 Low	No on board Terminations
CAN HI CH5	21	CAN 5 High	No on board Terminations
CAN LO CH5	23	CAN 5 Low	No on board Terminations
CAN HI CH6	25	CAN 6 High	No on board Terminations
CAN LO CH6	27	CAN 6 Low	No on board Terminations
CAN HI CH7	29	CAN 7 High	No on board Terminations
CAN LO CH7	31	CAN 7 Low	No on board Terminations
CAN HI CH8	33	CAN 8 High	No on board Terminations
CAN LO CH8	35	CAN 8 Low	No on board Terminations
CAN HI CH9	4	CAN 9 High	No on board Terminations
CAN LO CH9	6	CAN 9 Low	No on board Terminations
CAN HI CH10	8	CAN 10 High	No on board Terminations
CAN LO CH10	10	CAN 10 Low	No on board Terminations
CAN HI CH11	12	CAN 11 High	No on board Terminations
CAN LO CH11	14	CAN 11 Low	No on board Terminations
CAN HI CH12	16	CAN 12 High	No on board Terminations
CAN LO CH12	18	CAN 12 Low	No on board Terminations
CAN HI CH13	20	CAN 13 High	No on board Terminations
CAN LO CH13	22	CAN 13 Low	No on board Terminations
CAN HI CH14	24	CAN 12 High	No on board Terminations
CAN LO CH14	26	CAN 12 Low	No on board Terminations
CAN HI CH15	28	CAN 13 High	No on board Terminations
CAN LO CH15	30	CAN 13 Low	No on board Terminations
IRIG IN +	34	IRIG Time Code Input +	IRIG B Timecode Input +
IRIG IN -	36	IRIG Time Code Input -	IRIG B Timecode Input -
N.C.	2	Not Connected	No Connection
N.C.	32	Not Connected	No Connection
N.C.	37	Not Connected	No Connection
N.C.	37	Not Connected	No Connection
N.C.	37	Not Connected	No Connection
N.C.	40	Not Connected	No Connection

4.3 On Board CAN I/O P2

Name	P2 IN	Description	Notes
CAN0 GND	3	CAN Bus 0 GND	ISO CAN GND
CAN1 GND	7	CAN Bus 1 GND	ISO CAN GND
CAN2 GND	11	CAN Bus 2 GND	ISO CAN GND
CAN3 GND	15	CAN Bus 3 GND	ISO CAN GND
CAN4 GND	19	CAN Bus 4 GND	ISO CAN GND
CAN5 GND	23	CAN Bus 5 GND	ISO CAN GND
CAN6 GND	27	CAN Bus 6 GND	ISO CAN GND
CAN7 GND	31	CAN Bus 7 GND	ISO CAN GND
CAN8 GND	35	CAN Bus 8 GND	ISO CAN GND
CAN9 GND	4	CAN Bus 9 GND	ISO CAN GND
CAN10 GND	8	CAN Bus 10 GND	ISO CAN GND
CAN11 GND	12	CAN Bus 11 GND	ISO CAN GND
CAN12 GND	16	CAN Bus12 GND	ISO CAN GND
CAN13 GND	20	CAN Bus 13 GND	ISO CAN GND
CAN14 GND	24	CAN Bus 14 GND	ISO CAN GND
CAN15 GND	28	CAN Bus 15 GND	ISO CAN GND
RS232-TX	34	TX Data Output	RS232 Debug Port
RS232-RX	36	RX DATA Input	RS232 Debug Port
GND	38	GROUND	RS232 Debug Port
N.C.	1	Not Connected	No Connection
N.C.	5	Not Connected	No Connection
N.C.	9	Not Connected	No Connection
N.C.	13	Not Connected	No Connection
N.C.	17	Not Connected	No Connection
N.C.	21	Not Connected	No Connection
N.C.	25	Not Connected	No Connection
N.C.	29	Not Connected	No Connection
N.C.	33	Not Connected	No Connection
N.C.	37	Not Connected	No Connection
N.C.	39	Not Connected	No Connection
N.C.	2	Not Connected	No Connection
N.C.	6	Not Connected	No Connection
N.C.	10	Not Connected	No Connection
N.C.	14	Not Connected	No Connection
N.C.	18	Not Connected	No Connection
N.C.	22	Not Connected	No Connection
N.C.	26	Not Connected	No Connection
N.C.	30	Not Connected	No Connection
N.C.	32	Not Connected	No Connection
N.C.	40	Not Connected	No Connection

4.4 On Board Pn5 XMC Connector

	A	B	C	D	E	F
01	DP00+	DP00-	3.3V	DP01+	DP01-	VPWR
02	GND	GND	TRST#	GND	GND	MRSTI#
03	DP02+	DP02-	3.3V	DP03+	DP03-	VPWR
04	GND	GND	TCK	GND	GND	MRSTO#
05	DP04+	DP04-	3.3V	DP05+	DP05-	VPWR
06	GND	GND	TMS	GND	GND	+12V
07	DP06+	DP06-	3.3V	DP07+	DP07-	VPWR
08	GND	GND	TDI	GND	GND	-12V
09	DP08+	DP08-	RPS	DP09+	DP09-	VPWR
10	GND	GND	TDO	GND	GND	GA0
11	DP10+	DP10-	MBIST#	DP11+	DP11-	VPWR
12	GND	GND	GA1	GND	GND	MPRESENT#
13	DP12+	DP12-	3.3V AUX	DP13+	DP13-	VPWR
14	GND	GND	GA2	GND	GND	MSDA
15	DP14+	DP14-	RPS	DP15+	DP15-	VPWR
16	GND	GND	MVMRO	GND	GND	MSCL
17	DP16+	DP16-	RFU	DP17+	DP17-	RFU
18	GND	GND	RPS	GND	GND	RPS
19	DP18+	DP18-	RPS	DP19+	DP19-	RPS

5 CANFLIGHT-16 Stand Alone 16-Channel Interface

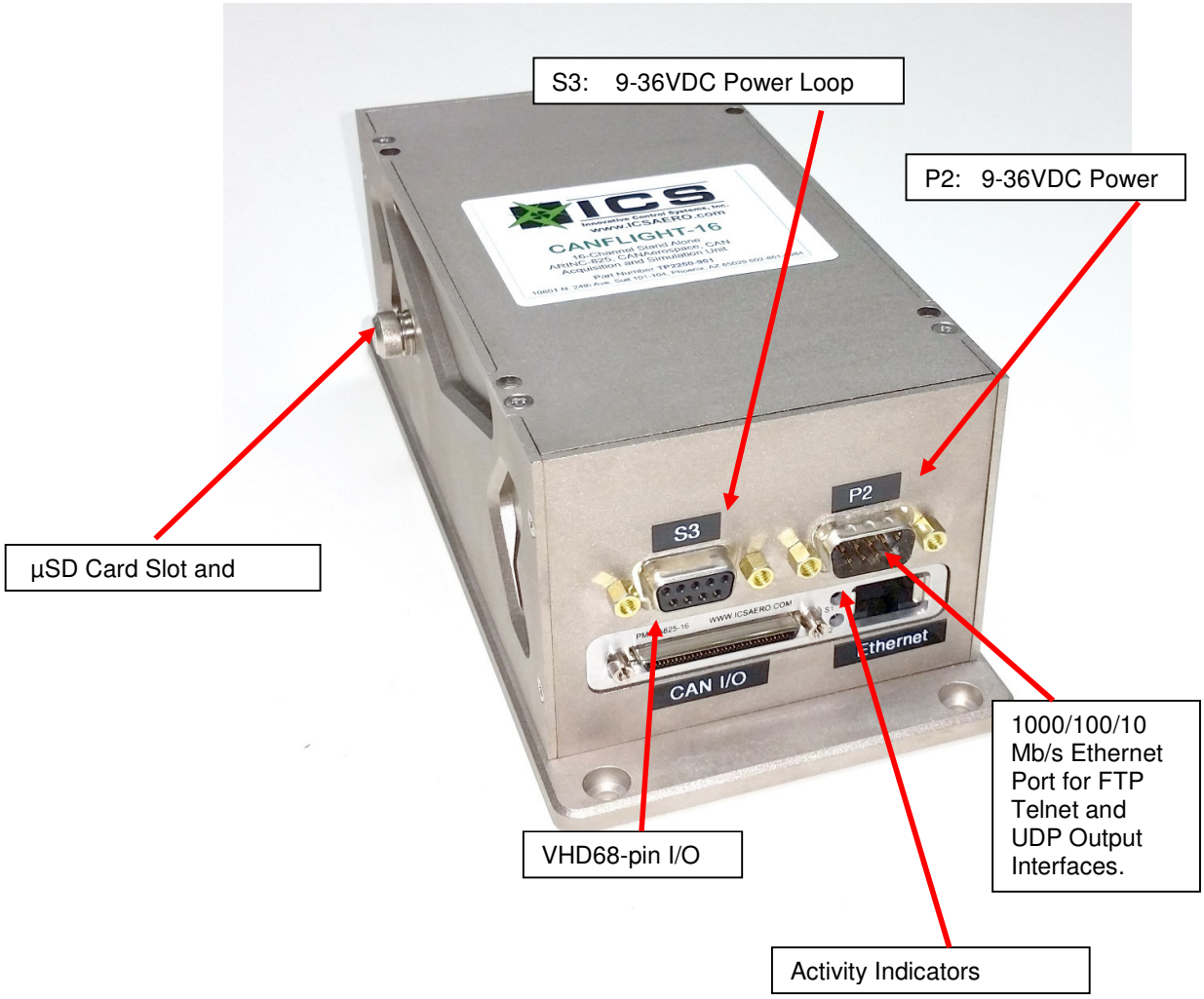


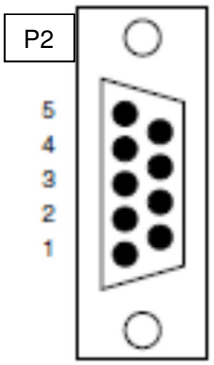
Figure 5-1 CANFLIGHT-16 configuration of XMC-A825-16

5.1 CANFlight-16 9-36VDC Power Connectors

Power is supplied through 9-PIN D sub connectors located above the main XMC-A825-16 processing board as shown in figure Figure 4 I/O Connectors above.

P2

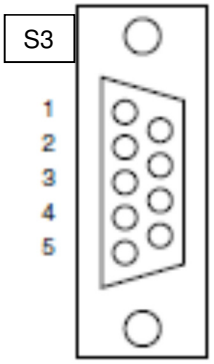
5	Power Ground (-)
4	No Connected, Future Use
3	No Connected, Future Use
2	No Connected, Future Use
1	Power +9-36VDC < 2 AMPS



9	No Connected, Future Use
8	No Connected, Future Use
6	No Connected, Future Use
7	No Connected, Future Use

S3

1	Power +9-36VDC < 2 AMPS
2	No Connected, Future Use
3	No Connected, Future Use
4	No Connected, Future Use
5	Power Ground (-)



6	No Connected, Future Use
7	No Connected, Future Use
8	No Connected, Future Use
9	No Connected, Future Use

Figure 5-2 CANFlight-16 Power Connectors

6 TP9060 Optional CAN-FD IO PANEL Connectors

The CAN-FD IO Panel is shown in Figure Illustration. These panel connectors provide easy access to the 16 CAN interfaces as shown in Figure Illustration, the usage and pin assignment of the connectors is described in Table Table.

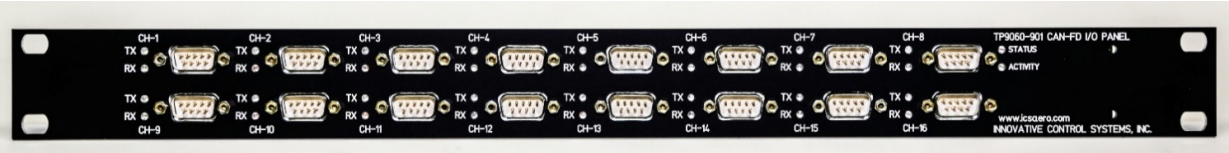


Figure 6-1 CAN-FD IO PANEL

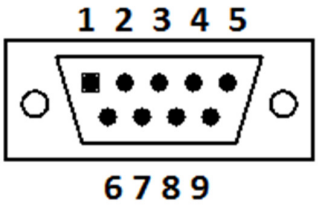


Figure 6-2 Connectors P1 to P16 CAN D-Sub Male Pinout

Pin	Signal Name	Signal Description
1	Unused	Do not connect
2	**CAN-L**	**CAN Low**
3	*OPT TRM-L	*Connect to pin2 for 120ohm termination at his conn.
4	Unused	Do not connect
5	Unused	Do not connect
6	Unused	Do not connect
7	**CAN-H**	**CAN High**
8	*OPT TRM-H	*Connect to pin7 for 120ohm termination at his conn.
9	GND_CH n	Ground Ref for this isolated channel

Table 5: P1 to P16 CAN D-Sub Male Connector Pinout

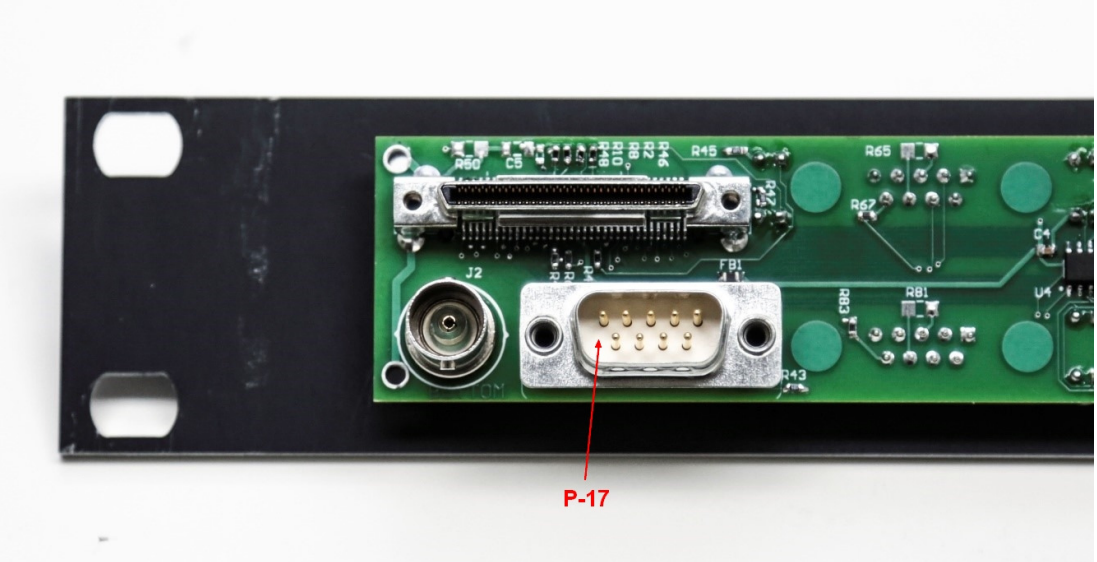
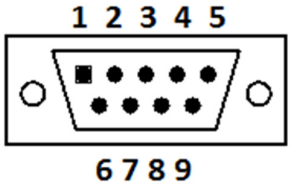


Figure 6-3 Debug P17 D-Sub Male Connector Pinout



Pin	Signal Name	Signal Description
1	Unused	Do not connect
2	RS232_TX	Debug TX
3	RS232_RX	Debug RX
4	Unused	Do not connect
5	RS232_GND	Ground Ref for this connector
6	Unused	Do not connect
7	Unused	Do not connect
8	Unused	Do not connect
9	Unused	Do not connect

Table 6: P17 D-Sub Male Connector Signal Definition

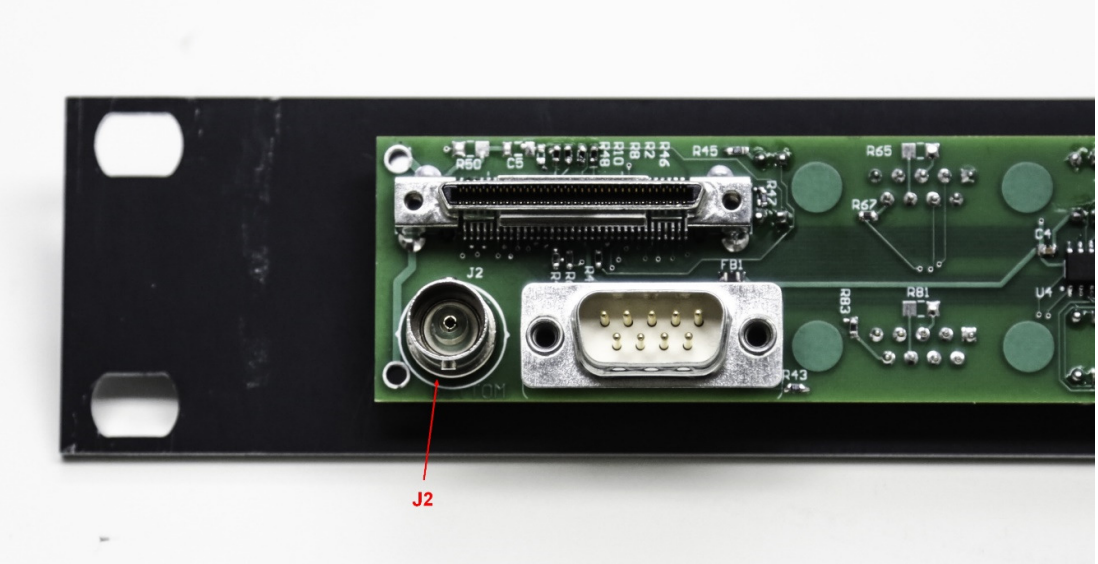


Figure 6-4 J2 IRIG-B BNC Jack

The J2 IRIG-B BNC connector accepts a signal according to IRIG Standard 200-04 on the center contact with the reference ground on the outer shield

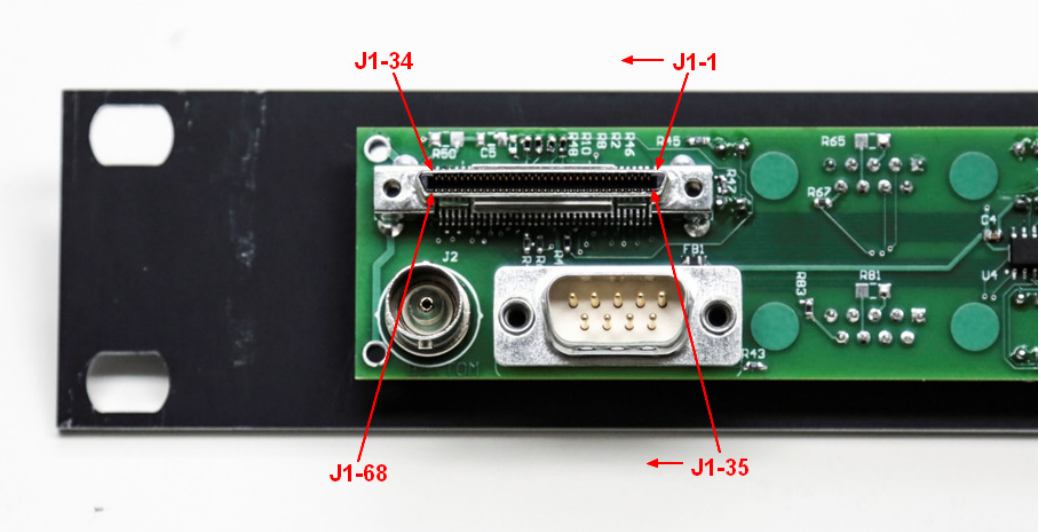


Figure 6-5 J1 VHD-68 Connector

The Interface Connector: J1 a VHD-68 is connected by the supplied cable to the TP2216-901 ARINC-825 16 Channel PMC front plate VHD-68 connector.

7 Ethernet Interface

XMC-A825-16 uses its Ethernet interface for the communication between a (theoretically) unlimited number of other XMC-A825-16 systems and/or host computers as shown in Figure ***Figure 3-2*** The Ethernet interface auto-negotiates its data rate with switches, routers or other network nodes between 10 and 1000 Mbit/s. To support smooth integration into standard Ethernet networks, XMC-A825-16 responds to Internet Control Message Protocol (ICMP) echo requests ("ping") as well as to Address Resolution Protocol (ARP) requests. The XMC-A825-16 IP address assignment may be either static or dynamic. XMC-A825-16 contains a DHCP client to support dynamic IP address assignment.

XMC-A825-16 employs the User Datagram Protocol (UDP) for the communication with host computers. IP addresses and UDP port numbers used by XMC-A825-16 may be assigned for each CAN channel individually. This maximizes flexibility for the integration of XMC-A825-16 systems into already configured networks. On power up, XMC-A825-16 obtains required Media Access Control (MAC) addresses from remote hosts using ARP request messages.

Note that delays introduced through switches/routers or through network traffic generated by other network nodes may adversely affect the XMC-A825-16 communication speed. If response times are vital, point-to-point Ethernet connections between XMC-A825-16 and host computers should be preferred.

XMC-A825-16 comes with an Application Programmer Interface (API) for the XMC-A825-16 Ethernet link supporting various operating systems (Linux, Solaris, MacOS, VxWorks, Windows). For a detailed description of the API refer to section 10 .

8 XMC-A825-16 Configuration

The XMC-A825-16 configuration is accomplished through a human readable ASCII configuration file that is stored on a µSD card which is inserted in the XMC-A825-16 µSD card slot. The configuration file has to have the case-sensitive name "XMC_A825.CFG" to be recognized by the firmware. This file is read by the XMC-A825-16 firmware from the µSD card slot each time power is applied. The content is used to configure the CAN baud rate, the local Ethernet interface and the board "name" used by XCT for additional reference and display of the board it is connected to.

The settings made through the configuration file are used on power-up but may be changed at any time using the corresponding Application Programmer Interface (API) system calls.

Additionally, the configuration file allows to specify default settings for the IP addresses, MAC addresses and port numbers for the communication with XCT (or the Ethernet API) on a per-CAN-channel-basis. The configuration file format uses the case-sensitive tags shown in Table 7: XMC-A825-16 Configuration File Tags. No spaces are allowed between the tag, the "=" and the following letters. All data in the configuration file that does not begin with a recognized tag will be ignored. This allows user comments in the file if caution is taken that no tag letters are used.

Note that all numbers for the tags "LMA=", "LIP=", "Urx=", "MAx=", "IPx=", "LPx=" and "RPx=" must use the corresponding fixed length format (i.e. "IP0=192.009.200.003", not "IP0=192.9.200.3")

Tag	Meaning	Format Example	Description
NME=	XMC-A825-16 Name	NME=XMC-A825-16 # 1	A string of up to 32 bytes consisting of a human readable ASCII text that gives the XMC-A825-16 unit a name. This name can be read from the module via the GET_MODULE_INFO system call at any time. All characters exceeding 32 bytes will be ignored.
LMA=	Local MAC Address	LMA=00:01:02:03:04:05	MAC address of the XMC-A825-16 unit, consisting of six two-digit hexadecimal numbers from 00 to FF, separated by colons. All Letters (A-F) have to be capital. **Note that the first byte has to be "00", otherwise the Ethernet interface will not work!**
LIP=	Local IP Address	LIP=192.009.200.003 LIP=DHCP	IP address of the XMC-A825-16 unit, consisting of four three-digit decimal numbers in the range of 000-255, separated by dots. Alternatively, XMC-A825-16 may be directed to obtain its IP address from a DHCP server by specifying the IP address as the four letter acronym "DHCP" in capital letters.
PCI=	0,1	PCI=0 or PCI=1	PCI=0 Use RJ-45 connector for communications with XMC 16 channel card PCI=1 Use On Board Intel Gigabit Ethernet Controller for communications
CB0= CB1=	CAN Baud Rate	CB1=250	Baud rate of the specified CAN channel. The following settings are valid and specify the baud rate in kbit/s: CBx=83 CBx=125 CBx=250 CBx=500 CBx=1000
LS0= LS1=	CAN Channel Ethernet Link Switch	LS0=1	Activates/deactivates the Ethernet link of the CAN specified channel. The following settings are valid and specify the state of the link (0 = inactive, 1 = active): LSx=0 LSx=1 Note that disabled Ethernet links will continue to transmit CAN status packets once per second and respond to IMCP packets.

Tag	Meaning	Format Example	Description
UR0= UR1=	CAN Channel Ethernet Update Rate	UR1=0100	Sets the gap between subsequent emissions of UDP/IP packets containing received CAN messages for the specified CAN channel in milliseconds, as a 4-digit decimal number in the range of 0001-9999. Note that this setting does ***not*** affect the continuous transmission of CAN status packets (every 100ms and once per second as broadcast) and the ability of the XMC-A825-16 to respond to IMCP packets.
RB0= RB1=	CAN Channel Readback Switch	RB0=0	Activates/deactivates the readback of CAN messages on the same channel as they are transmitted. The following settings are valid and specify the state of the readback (0 = inactive, 1 = active): RBx=0 RBx=1
MA0= MA1=	Remote MAC Address	MA1=00:14:4F:C3:B9:A2	MAC address of the remote host for communication with the specified CAN channel, consisting of six two-digit hexadecimal numbers from 00 to FF, separated by colons. All Letters (A-F) have to be capital.
IP0= IP1=	Remote IP Address	IP0=192.009.200.051	IP address of the remote host for communication with the specified CAN channel, consisting of four three-digit decimal numbers in the range of 000-255, separated by dots.
LP0= LP1=	Local UDP Port Number	LP0=34567	UDP port number of the XMC-A825-16 unit used to receive Ethernet/UDP/IP packets from, as a 5-digit decimal number in the range of 03000-65535. Note that port numbers below 3000 will create problems in many networks.
RP0= RP1=	Remote UDP Port Number	RP0=34568	UDP port number of the XMC-A825-16 unit used to send Ethernet/UDP/IP packets to, as a 5-digit decimal number in the range of 03000-65535. Note that port numbers below 03000 will create problems in many networks.

Tag	Meaning	Format Example	Description
BR0= BR1=	CAN Channel Bridge Switch	BR1=0	Activates/deactivates the bridge function of the specified CAN channel. If both channels have the bridge function activated, all CAN messages are transferred between the channels in both directions. The following settings are valid and specify the state of the readback (0 = inactive, 1 = active): BRx=0 BRx=1 **Note that the bridge function is currently not implemented in the XMC-A825-16 firmware.**
FDR=	Flight Data Recording Mode	FDR=RUN	Activates the Flight Data Recording Mode. The following settings are valid: FDR=RUN FDR=CAN FDR=ETH If RUN is selected, the data recording starts immediately after power-on and stops on power-off. A new recording file is created each time when the seconds specified through the “RFT=” tag have elapsed. If CAN is selected, the recording process is controlled through the TOP_MARKER CANaerospace message, either on CAN channel 1 or 2. If ETH is selected, the recording process is controlled through the Ethernet interface. **Note that this mode is currently not implemented in the XMC-A825-16 firmware.**
DRC=	CAN Channel Recording Switch	DRC=3	Specifies which CAN channels are recorded when the Flight Data Recording Mode is active: DRC=1 (Channel 1 only) DRC=2 (Channel 2 only) DRC=3 (Channel 1 + 2)
RFT=		RFT=0060	Specified the time after which a new data recording file is generated when the Data Recording is active in the RUN mode, in seconds, as a 4-digit decimal number in the range of 0001-9999.

Table 7: XMC-A825-16 Configuration File Tags

A sample XMC-A825-16 configuration file is shown below:

```
# XMC-A825 Configuration File
# (C) Stock Flight Systems 2015
# Board name (max. 32 bytes).

NME=XMC-A825 #1

# This XMC-A825s MAC and IP address.
LMA=00:01:02:03:07:02
LIP=192.009.200.033
#
# Ethernet Interface usage
#
# PCI=0  Use RJ-45 connector for communications with XMC 16 channel card
# PCI=1  Use On Board Intel Gigabit Ethernet Controller for communications
#

PCI=0

# CAN interface baud rates (1000, 500, 250, 125 or 83).
CB0=1000
CB1=1000
CB2=1000
CB3=1000
CB4=1000
CB5=1000
CB6=1000
CB7=1000
CB8=1000
CB9=1000
CB10=1000
CB11=1000
CB12=1000
CB13=1000
CB14=1000
CB15=1000

# MAC/IP addresses and local/remote port numbers for CAN channel 0.
IP0=192.009.200.051
LP0=34567
RP0=34568
LS0=1
UR0=0010
RB0=1

# MAC/IP addresses and local/remote port numbers for CAN channel 1.
IP1=192.009.200.051
LP1=34569
RP1=34570
LS1=1
```

```
UR1=0010
RB1=1

# MAC/IP addresses and local/remote port numbers for CAN channel 2.
IP2=192.009.200.051
LP2=34571
RP2=34572
LS2=1
UR2=0010
RB2=1

# MAC/IP addresses and local/remote port numbers for CAN channel 3.
IP3=192.009.200.051
LP3=34573
RP3=34574
LS3=1
UR3=0010
RB3=1

# MAC/IP addresses and local/remote port numbers for CAN channel 4.
IP4=192.009.200.051
LP4=34575
RP4=34576
LS4=1
UR4=0010
RB4=1

# MAC/IP addresses and local/remote port numbers for CAN channel 5.
IP5=192.009.200.051
LP5=34577
RP5=34578
LS5=1
UR5=0010
RB5=1

# MAC/IP addresses and local/remote port numbers for CAN channel 6.
IP6=192.009.200.051
LP6=34579
RP6=34580
LS6=1
UR6=0010
RB6=1

# MAC/IP addresses and local/remote port numbers for CAN channel 7.
IP7=192.009.200.051
LP7=34581
RP7=34582
LS7=1
UR7=0010
RB7=1

# MAC/IP addresses and local/remote port numbers for CAN channel 8.
```

```
IP8=192.009.200.051
LP8=34583
RP8=34584
LS8=1
UR8=0010
RB8=1

# MAC/IP addresses and local/remote port numbers for CAN channel 9.
IP9=192.009.200.051
LP9=34585
RP9=34586
LS9=1
UR9=0010
RB9=1

# MAC/IP addresses and local/remote port numbers for CAN channel 10.
IP10=192.009.200.051
LP10=34587
RP10=34588
LS10=1
UR10=0010
RB10=1

# MAC/IP addresses and local/remote port numbers for CAN channel 11.
IP11=192.009.200.051
LP11=34589
RP11=34590
LS11=1
UR11=0010
RB11=1

# MAC/IP addresses and local/remote port numbers for CAN channel 12.
IP12=192.009.200.051
LP12=34591
RP12=34592
LS12=1
UR12=0010
RB12=1

# MAC/IP addresses and local/remote port numbers for CAN channel 13.
IP13=192.009.200.051
LP13=34593
RP13=34594
LS13=1
UR13=0010
RB13=1

# MAC/IP addresses and local/remote port numbers for CAN channel 14.
IP14=192.009.200.051
LP14=34595
RP14=34596
LS14=1
```

```
UR14=0010
RB14=1

# MAC/IP addresses and local/remote port numbers for CAN channel 15.
IP15=192.009.200.051
LP15=34597
RP15=34598
LS15=1
UR15=0010
RB15=1

# End of XMC-A825 Configuration File
```

For CAN channels which are not accessed via the Ethernet/UDP/IP interface, the corresponding entries in the configuration file for MAC/IP addresses and port numbers may be omitted. All CAN channels that have no baud rate tag in the configuration file will be initialized to be “bus-off” by default. Specifying “255” as the last digit of an “IPx=” tag sets up the corresponding interface for broadcast communication. In this case, multiple instances of XCT may connect to the same CAN channel of a XMC-A825-16 unit (port numbers are still relevant in this case). The MAC address for broadcast channels is automatically set to FF:FF:FF:FF:FF:FF by the XMC-A825-16 firmware.

If the “LIP=”-tag is set to “DHCP”, XMC-A825-16 attempts to contact a DHCP server in the network and tries to obtain a valid IP address from it. Once this process has been completed successfully, the XMC-A825-16 unit will transmit a status message for CAN channel 0 once per second as a broadcast UDP packet. Reception of this packet allows host computers in the network to determine the IP address of a XMC-A825-16 based on its module name.

If the MAC address of a remote host is not known, the corresponding “MAx” tag may be omitted. In this case, the XMC-A825-16 will use ARP requests ten times per second to determine the remote hosts MAC address in order to establish communication with this host. Specifying the remote hosts MAC address in the configuration file avoids this process and speeds up the link initialization.

The continuous transmission of Ethernet/UDP/IP packets with received CAN messages can be enabled or disabled using the “LSx=” tag for each CAN channel individually. Likewise, the time between subsequent packets can be adjusted through the “URx=” tag. The minimum time gap between CAN receive packets is 1ms, the maximum time gap is 9999ms. Be aware that specifying large gap times may lead to data loss depending on the traffic on the corresponding CAN channel.

The “LSx=” and “URx=” tag settings do not affect the transmission of CAN status packets (every 100ms and once per second as broadcast) and the ability of the XMC-A825-16 to respond to Internet Control Message Protocol (ICMP) echo requests (“ping”) as well as to Address Resolution Protocol (ARP) requests.
For communication with XCT, the local and remote port numbers specified in the XCT setup have to match the XMC-A825-16 configuration. The default values are shown in Table 8 These values have to be specified correctly to enable communication between XMC-A825-16 and XCT.

Depicted CAN Channel	Logical CAN Channel	Local (XMC-A825-16) Port Number (“LPx=”-Tag)	Remote (Host) Port Number (“RPx=”-Tag)
CH1/CAN1	0	34567	34568

Depicted CAN Channel	Logical CAN Channel	Local (XMC-A825-16) Port Number (“LPx=”-Tag)	Remote (Host) Port Number (“RPx=”-Tag)
CH2/CAN2	1	34569	34570
CH3/CAN3	2	34571	34572
CH4/CAN4	3	34573	34574
CH5/CAN5	4	34575	34576
CH6/CAN6	5	34577	34578
CH7/CAN7	6	34579	34580
CH8/CAN8	7	34581	34582
CH9/CAN9	8	34583	34584
CH10/CAN10	9	34585	34586
CH11/CAN11	10	34587	34588
CH12/CAN12	11	34589	34590
CH13/CAN13	12	34591	34592
CH14/CAN14	13	34593	34594
CH15/CAN15	14	34595	34596
CH16/CAN16	15	34597	34598

Table 8: Local and Remote Port Numbers for Communication with XCT (Default Settings)

9 XMC-A825-16 Firmware Upgrades

XMC-A825-16 allows firmware upgrades to be made through the µSD card interface. The XMC-A825-16 binary firmware upgrade files have to be stored on a µSD card which is inserted in the XMC-A825-16 µSD card slot. The files must have the case-sensitive name “mb0.srd” through “mb4.srd” to be recognized by the XMC-A825-16 firmware which checks for these files each time power is applied. When these files are detected, the content is automatically programmed into FLASH memory and the new firmware is started.

On success, the XMC-A825-16 firmware deletes the upgrade files from the µSD card and writes a log file (“INSTALL.LOG”) to the card which provides information about the upgrade. The content of this log file will typically look as follows:

```
MB0 Software Update File 'mb0.srd' found:
    MB0 Software Update Programming Successful.
    MB0 Software Update Verification Successful.
    MB0 Software Update File Deleted.
MB1 Software Update File 'mb1.srd' found:
    MB1 Software Update Programming Successful.
    MB1 Software Update Verification Successful.
    MB1 Software Update File Deleted.
MB2 Software Update File 'mb2.srd' found:
    MB2 Software Update Programming Successful.
    MB2 Software Update Verification Successful.
    MB2 Software Update File Deleted.
MB3 Software Update File 'mb3.srd' found:
    MB3 Software Update Programming Successful.
    MB3 Software Update Verification Successful.
    MB3 Software Update File Deleted.
MB4 Software Update File 'mb4.srd' found:
    MB4 Software Update Programming Successful.
    MB4 Software Update Verification Successful.
    MB4 Software Update File Deleted.
```

Firmware upgrades using the µSD card are possible for Microblaze0, Mictoblaze1 or both XMC-A825-16 processors at the same time. The current XMC-A825-16 firmware is available online from:

www.arinc825.com

10 XMC-A825-16 Application Programmer's Interface

The XMC-A825-16 Application Programmer's Interface is a Socket Interface Library consisting of a set of functions which provide the interface between applications written in "C" and the XMC-A825-16 resources using 4.3 BSD datagram sockets. It is provided in source code and allows to be compiled and linked for various operating systems. The Interface Library has successfully been tested with SUN/Solaris, SuSE/RedHat Linux, MacOS, VxWorks and MS Windows. Compilation for other Unix derivatives including realtime operating systems like LynxOS, QNX or Integrity 178 should require only minor code changes, if any at all.

Example programs coming with the library show how applications make use of the library calls to attach to XMC-A825-16 channels, read and write CAN messages, perform control functions and detach from a XMC-A825-16 channel. All relevant interface structures and definitions are contained in just three include files (pmc825.h, canas.h, arinc825.h) to minimize integration effort. The current version of the library may be downloaded anytime from www.arinc825.com. Compatibility of new versions to previous versions is ensured so that upgrading to a new library version does not require changes in already existing applications.

The functions provided by the XMC-A825-16 Socket Interface Library are listed below. The calls are not multithread safe. Any application accessing the XMC-A825-16 has to call Pmc825StartInterface() once before using any other Socket Interface Library call. Good practice is to call Pmc825StopInterface() once the XMC-A825-16 resource is not needed by the application anymore. Failing to do so will prevent the XMC-A825-16 from freeing resources by shutting down the open UDP/IP socket.

The PMC825 Socket Interface Library provides the following user-callable routines:

- *Pmc825StartInterface()* - Establish a communication path to a PMC825 CAN channel
- *Pmc825StopInterface()* - Release a communication path to a PMC825 CAN channel
- *Pmc825RawCanRead()* - Read unformatted CAN messages
- *Pmc825RawCanWrite()* - Write unformatted CAN messages
- *Pmc825CanAerospaceRead()* - Read CANaerospace formatted CAN messages
- *Pmc825CanAerospaceWrite()* - Write CANaerospace formatted CAN messages
- *Pmc825Arinc825Read()* - Read ARINC 825 formatted CAN messages
- *Pmc825Arinc825Write()* - Write ARINC 825 formatted CAN messages
- *Pmc825CtrlRead()* - Read a PMC825 control response packet
- *Pmc825CtrlWrite()* - Write a PMC825 control command packet

10.1 Pmc825StartInterface()

Synopsis:

#include "pmc825.h"

*int Pmc825StartInterface(PMC825_IF *intf, unsigned int pm825_ip, unsigned int host_ip, int rx_port, int tx_port, int channel)*

Description:

The Pmc825StartInterface() function establishes the connection between the specified channel of a XMC-A825-16 unit and the host by the means of UDP/IP sockets. It initializes a PMC825_IF interface structure that refers to the CAN channel. The interface structure is used by other I/O functions to refer to that CAN channel.

Return Values:

Upon successful completion, *Pmc825StartInterface()* returns PMC825_OK. Otherwise, one of the following codes is returned:

PMC825_MEM_ALLOC_ERR: The memory required to establish communication could not be obtained.

PMC825_SOCKET_ERR: At least one of the communication sockets could not be successfully initialized.

10.2 Pmc825StopInterface()

Synopsis:

#include "pmc825.h"

*void Pmc825StopInterface(PMC825_IF *intf)*

Description:

The *Pmc825StopInterface()* function releases the connection between a XMC-A825-16 unit and the host by closing the corresponding UDP/IP sockets and releasing the allocated memory.

10.3 Pmc825RawCanRead()

Synopsis:

#include "pmc825.h"

*int Pmc825RawCanRead(PMC825_IF *intf, CAN_MSG *msg)*

Description:

The *Pmc825RawCanRead()* function tries to return one unformatted CAN message from the buffer associated with the CAN channel of the XMC-A825-16 unit specified through the PMC825_IF interface structure.

Return Values:

Upon successful completion, *Pmc825RawCanRead()* returns PMC825_OK. Otherwise, one of the following codes is returned:

PMC825_NO_MSG: No message could be read from the module.

10.4 Pmc825RawCanWrite()

Synopsis:

#include "pmc825.h"

*int Pmc825RawCanWrite(PMC825_IF *intf, CAN_MSG *msg, int msg_count)*

Description:

The *Pmc825RawCanWrite()* function tries to write one or more unformatted CAN messages to the buffer associated with the CAN channel of the XMC-A825-16 unit specified through the PMC825_IF interface structure for transmission. The number of messages to be transmitted is specified by "msg_count".

Return Values:

Upon successful completion, *Pmc825RawCanWrite()* returns PMC825_OK. Otherwise, one of the following codes is returned:

PMC825_BUF_OVERFLOW: The number of messages specified by "msg_count" exceeded the maximum number of MAX_CAN_MSG_COUNT.

10.5 Pmc825CanAerospaceRead()

Synopsis:

#include "pmc825.h"

#include "can_as.h"

*int Pmc825CanAerospaceRead(PMC825_IF *intf, CAN_AS_MSG *msg)*

Description:

The *Pmc825CanAerospaceRead()* function tries to return one CANaerospace formatted CAN message from the buffer associated with the CAN channel of the XMC-A825-16 unit specified through the PMC825_IF interface structure.

Return Values:

Upon successful completion, *Pmc825CanAerospaceRead()* returns PMC825_OK. Otherwise, one of the following codes is returned:

PMC825_NO_MSG: No message could be read from the module.

10.6 Pmc825CanAerospaceWrite()

Synopsis:

#include "pmc825.h"

#include "can_as.h"

*int Pmc825CanAerospaceWrite(PMC825_IF *intf, CAN_AS_MSG *msg, int msg_count)*

Description:

The *Pmc825CanAerospaceWrite()* function tries to write one or more CANaerospace formatted CAN messages to the buffer associated with the CAN channel of the XMC-A825-16 unit specified through the PMC825_IF

interface structure for transmission. The number of messages to be transmitted is specified by "msg_count".

Return Values:

Upon successful completion, *Pmc825CanAerospaceWrite()* returns PMC825_OK. Otherwise, one of the following codes is returned:

PMC825_BUF_OVERFLOW: The number of messages specified by "msg_count" exceeded the maximum number of MAX_CAN_MSG_COUNT.

10.7 Pmc825Arinc825Read()

Synopsis:

#include "pmc825.h"

#include "arinc825.h"

*int Pmc825Arinc825Read(PMC825_IF *intf, ARINC825_MSG *msg)*

Description:

The *Pmc825Arinc825Read()* function tries to return one ARINC 825 formatted CAN message from the buffer associated with the CAN channel of the XMC-A825-16 unit specified through the PMC825_IF interface structure.

Return Values:

Upon successful completion, *Pmc825Arinc825Read()* returns PMC825_OK. Otherwise, one of the following codes is returned:

PMC825_NO_MSG: No message could be read from the module.

10.8 Pmc825Arinc825Write()

Synopsis:

#include "pmc825.h"

*int Pmc825Arinc825Write(PMC825_IF *intf, ARINC825_MSG *msg, int msg_count)*

Description:

The *Pmc825Arinc825Write()* function tries to write one or more ARINC 825 formatted CAN messages to the buffer associated with the CAN channel of the XMC-A825-16 unit specified through the PMC825_IF interface structure for transmission. The number of messages to be transmitted is specified by "msg_count".

Return Values:

Upon successful completion, *Pmc825Arinc825Write()* returns PMC825_OK. Otherwise, one of the following codes is returned:

PMC825_BUF_OVERFLOW: The number of messages specified by "msg_count" exceeded the maximum number of MAX_CAN_MSG_COUNT.

10.9 Pmc825CtrlRead()

Synopsis:

#include "pmc825.h"

*int Pmc825CtrlRead(PMC825_IF *intf, CTRL_MSG *ctrl_msg)*

Description:

The *Pmc825CtrlRead()* function tries to return one PMC825 control message from the buffer associated with the CAN channel of the XMC-A825-16 unit specified through the PMC825_IF interface structure.

Return Values:

Upon successful completion, *Pmc825CtrlRead()* returns PMC825_OK. Otherwise, one of the following codes is returned:

PMC825_NO_MSG: No message could be read from the module.

10.10 Pmc825CtrlWrite()

Synopsis:

#include "pmc825.h"

*int Pmc825CtrlWrite(PMC825_IF *intf, CTRL_MSG *msg)*

Description:

The *Pmc825CtrlWrite()* function tries to write a PMC825 control messages to the buffer associated with the CAN channel of the XMC-A825-16 unit specified through the PMC825_IF interface structure for transmission.

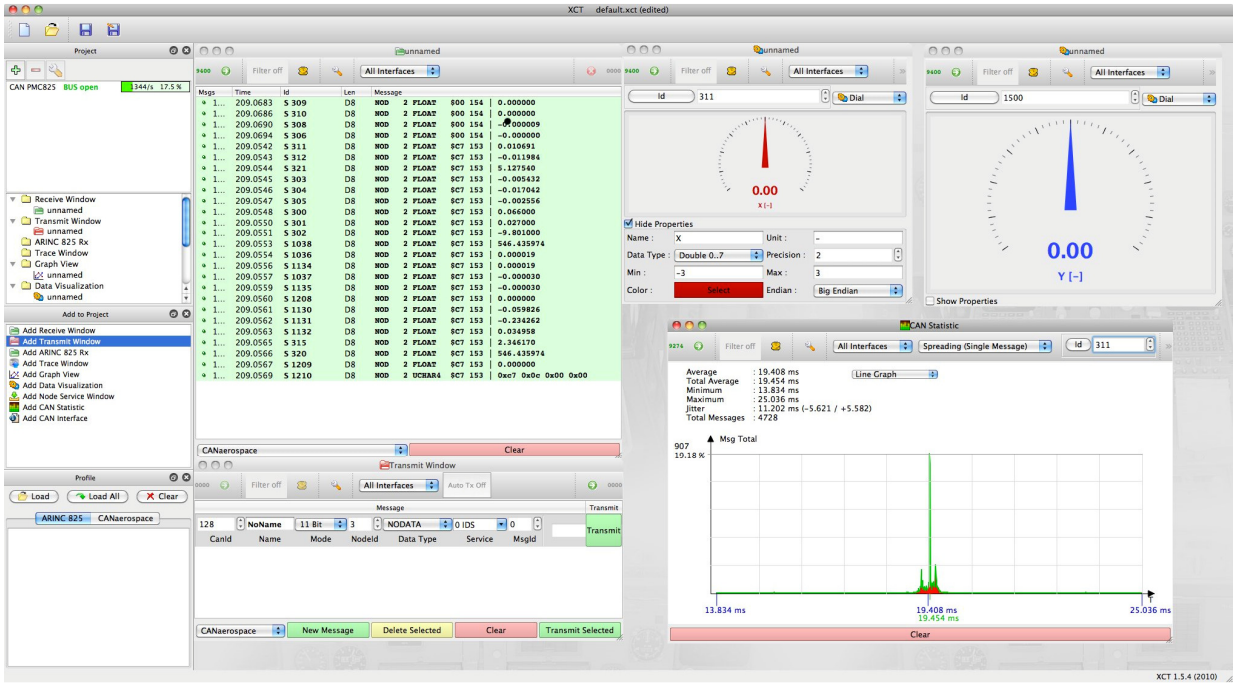
Return Values:

Upon successful completion, *Pmc825Arinc825Write()* returns PMC825_OK. Otherwise, one of the following codes is returned:

PMC825_BUF_OVERFLOW: The number of messages specified by "msg_count" exceeded the maximum number of MAX_CAN_MSG_COUNT.

11 The XCT Toolbox

XMC-A825-16 is delivered with the e**X**tended **C**AN **T**ool (XCT) software, a powerful CAN, ARINC825 and CANaerospace network toolbox for Linux, MacOS and Windows XP/7. Figure Illustration shows the main window of XCT.



Among other features, XCT contains an ARINC825 Communication Profile reader and editor, realtime data visualization in raw, ARINC825 and CANaerospace formats, network traffic/error statistics and an interface for CANaerospace/ARINC825 Periodic Health Status Messages and Node Services. XCT may be used also for interfacing to end systems corresponding to the ARINC specifications 812 and 826 which are both based on ARINC825. XCT contains all necessary functions for CAN, CANaerospace and ARINC825 network compatibility verification, end system testing, CAN network timing analysis and ARINC825 communication profile generation and analysis. The User's Manual for XCT is contained in the toolbox itself. The current version for all supported operating systems may be downloaded from:

www.wetzel-technology.com/files/XCT

XCT has a window-oriented interface that communicates with XMC-A825-16 using an Ethernet/UDP/IP connection. Multiple instances of XCT may connect to any XMC-A825-16 channel and control transmission and reception of CAN messages. XCT configuration files containing application specific settings ("Project") can be generated and reloaded. XCT project configuration files allow to save and reload XCT configurations and exchange them with other XCT users.

12 XMC-A825-16 Supplier List

Supplier	Contact	
Innovative Control Systems, LP.	10801 N 24th Ave. Suite 103 Phoenix, AZ 85029 USA phone: +1-602-861-6984 fax: +1-602-794-6453 e-mail: support@icsaero.com website: www.icsaero.com	
Stock Flight Systems	Schützenweg 8a 82335 Berg/Farchach Germany phone: +49-8151-9607-0 fax: +49-8151-9607-30 e-mail: info@stockflightsystems.com website: www.stockflightsystems.com	
Wetzel Technology GmbH	Hermann-Oberth-Straße 11 85640 Putzbrunn Germany phone: +49-89-460892-62 fax: +49-89-460892-63 e-mail: info@wetzel-technology.com website: www.wetzel-technology.com	
Reiser Simulation and Training GmbH	Oberer Luessbach 29-31 85335 Berg/Höhenrain Germany phone: +49-8178-8681-0 fax: +49-8178-8681-30 e-mail: info@resier-st.com web: www.reiser-st.com	

13 ARINC825/CANaerospace Websites

www.arinc.com
www.arinc825.com
www.canaerospace.net