Tristan Technologies

iMAG-400 4-Channel SQUID Electronics

By:

TRISTAN TECHNOLOGIES, Inc

San Diego, California
USA

Tristan Technologies, Inc. Part No. MAN-2064-004

Revision Record		
Date	**Revision**	**Description**
May 9, 2002	version 1.1	Customer Release
Jan 1, 2006	version 1.3	Firmware Upgrade
November 1, 2006	version 1.4	iPS-400 added
Sept 15, 2011	Version 1.5	update

Any questions or comments in regard to this product and other products from Tristan Technologies, please contact or see www.tristantech.com:

Tristan Technologies, Inc.

6191 Cornerstone Court East, Suite 107
San Diego, CA 92121 U. S. A.

Technical Support: (858) 550-2700
Fax: (858) 550-2799

e-mail: info1@tristantech.com

WARRANTY

TRISTAN TECHNOLOGIES Limited Warranty

TRISTAN TECHNOLOGIES Inc. warrants this product for a period of twelve (12) months from date of original shipment to the customer. Any part found to be defective in material or workmanship during the warranty period will be repaired or replaced without charge to the owner. Prior to returning the instrument for repair, authorization must be obtained from TRISTAN TECHNOLOGIES Inc. or an authorized TRISTAN TECHNOLOGIES service agent. All repairs will be warranted for only the unexpired portion of the original warranty, plus the time between receipt of the instrument at TRISTAN TECHNOLOGIES and its return to the owner.

This warranty is limited to TRISTAN TECHNOLOGIES' products that are purchased directly from TRISTAN TECHNOLOGIES, its OEM suppliers, or its authorized sales representatives. It does not apply to damage caused by accident, misuse, fire, flood or acts of God, or from failure to properly install, operate, or maintain the product in accordance with the printed instructions provided.

This Warranty Is In Lieu Of Any Other Warranties, Expressed Or Implied, Including Merchantability Or Fitness For Purpose, Which Are Expressly Excluded. The Owner Agrees That TRISTAN TECHNOLOGIES' Liability With Respect To This Product Shall Be As Set Forth In This Warranty, And Incidental Or Consequential Damages Are Expressly Excluded.

Table of Contents

Table of Figures

System Design

Tristan's iMAG400 4-channel SQUID electronics consists of a flux-locked loop (FLL) box (model iMC-404) and a separate power supply (model iPS-400). The iPS-400 can power one or two separate iMC-404s for a maximum of 8 channels. Each iMC-404 can operate 1-4 channels simultaneously.

Tristan Technologies' iMAG400 SQUID electronics consist of one or more iMC-404 Flux-Locked Loop control electronics and either iPS-400 Power Supply/Interface(s) or 19" rack mount card cage with an associated power supply.

iMC-404 Flux-Locked Loop

The iMC-404 flux-locked loop control box provides all necessary control signals (bias, modulation, and feedback, heater drive) necessary to operate up to four SQUID sensors.

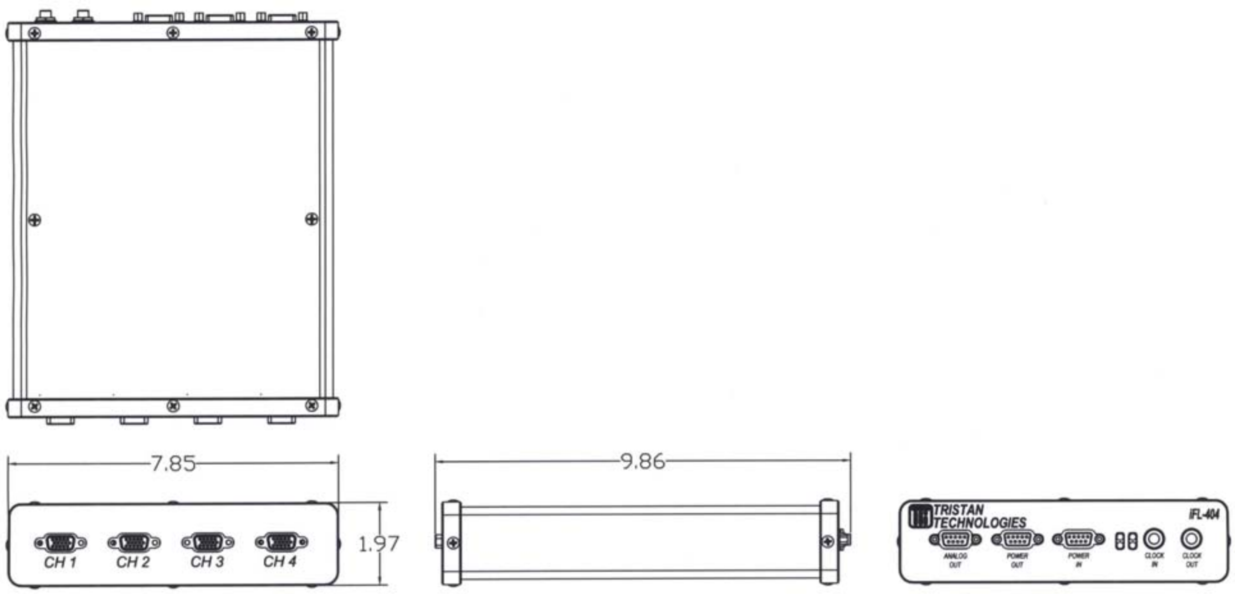

Figure 1: iMC-404 flux-locked loop control box dimensions (in inches)

Two iMC-404 flux-locked loop control boxes can be used with a single iPS-400 Power Supply/interface box for a maximum of 8 channels/iPS-400. For higher channel count requirements, the iPS-400 is replaced by a 19" rack mount card cage (Figure 2) and the iPS-400 with a power distribution hub. This can allow up to 84 channels/card cage to be accommodated.

Sensitivity: < 5 $\mu\Phi_o/\sqrt{Hz}$ (white noise equivalent) when used with Tristan Technologies model LSQ/20 LTS SQUID sensors or equivalent and SP-1.5 cryocable. Cable lengths significantly different from 1.5 m may affect system noise.

Bandwidth & Gain: dc – 50 kHz. Selectable gains of (1, 10, 100) corresponding to full-scale outputs ranging from approximately $\pm 300\ \Phi_o$ to $\pm 3\ \Phi_o$.

Remote Interfaces: RS232 or RS485 remote control interfaces are configurable. All control settings may be via these interfaces. The system is settable to 1200 to 57,600 baud for RS485 or 9600 baud for RS-232

Autotune: Autotuning of SQUID parameters is performed by a single command for each active SQUID channel. All adjustments may also be made manually via the remote interfaces.

FLL Reset: Any channel may be reset manually or automatically when the output saturates. Additionally, all channels may be simultaneously rest by a hardware active low signal input to the iPS-400.

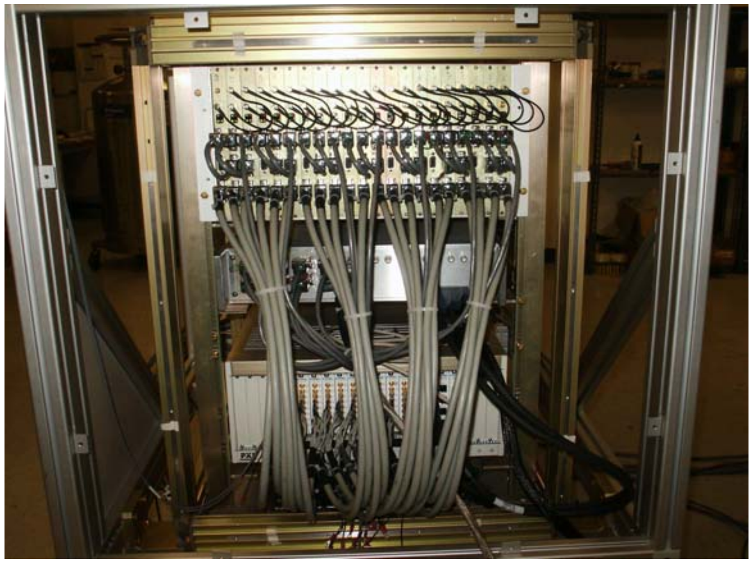

Figure 2: 84 channels of iMAG400 electronics in a standard rack mount card cage. The power distribution hub is located in the middle of the rack. The lower electronics box contains 84 channels of 24 bit data acquisition cards.

iPS-400 Power Supply/Interface

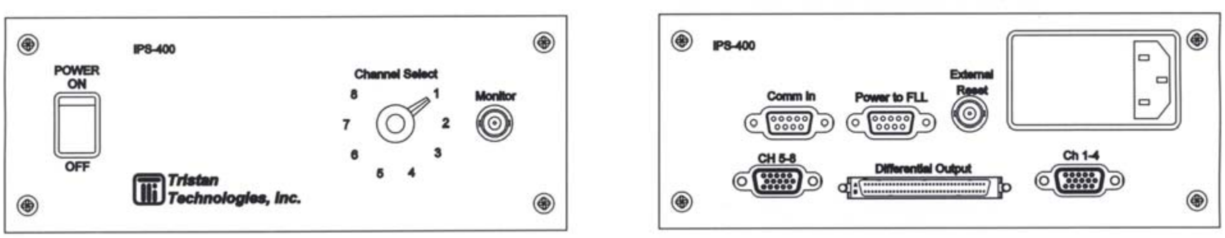

Figure 3: iPS-400 front and rear panel

The iPS400 supplies conditioned power for up to two iMC-404 flux-locked loop control boxes. It also provides the user with front panel selectable analog output (nominal 600 Ω) that can monitor the output of any of up to 8 SQUID channels. Rear panel connectors include cabling to the iMC-404 flux-locked loop control box(es) and a National Instruments compatible 68-pin connector provides differential analog outputs for 1-8 channels.

Dimensions: 215 mm wide, 87 mm high, 336 mm deep (8½” wide, 3½” high, 13¼” deep).

Weights: iMC-404: 1 kg (2.2 lbs)**; iPS-400:** 3.4 kg (7.5 lbs)

Power Requirements: 115 or 230 Volts AC, 50 or 60 Hz. Power consumption ~ 12 Watts.

Operating Voltage should be specified at time of order.

The iPS-400 can be configured for RS-485 or RS-232 communications by moving internal jumpers on the PCB as shown in the figures below.

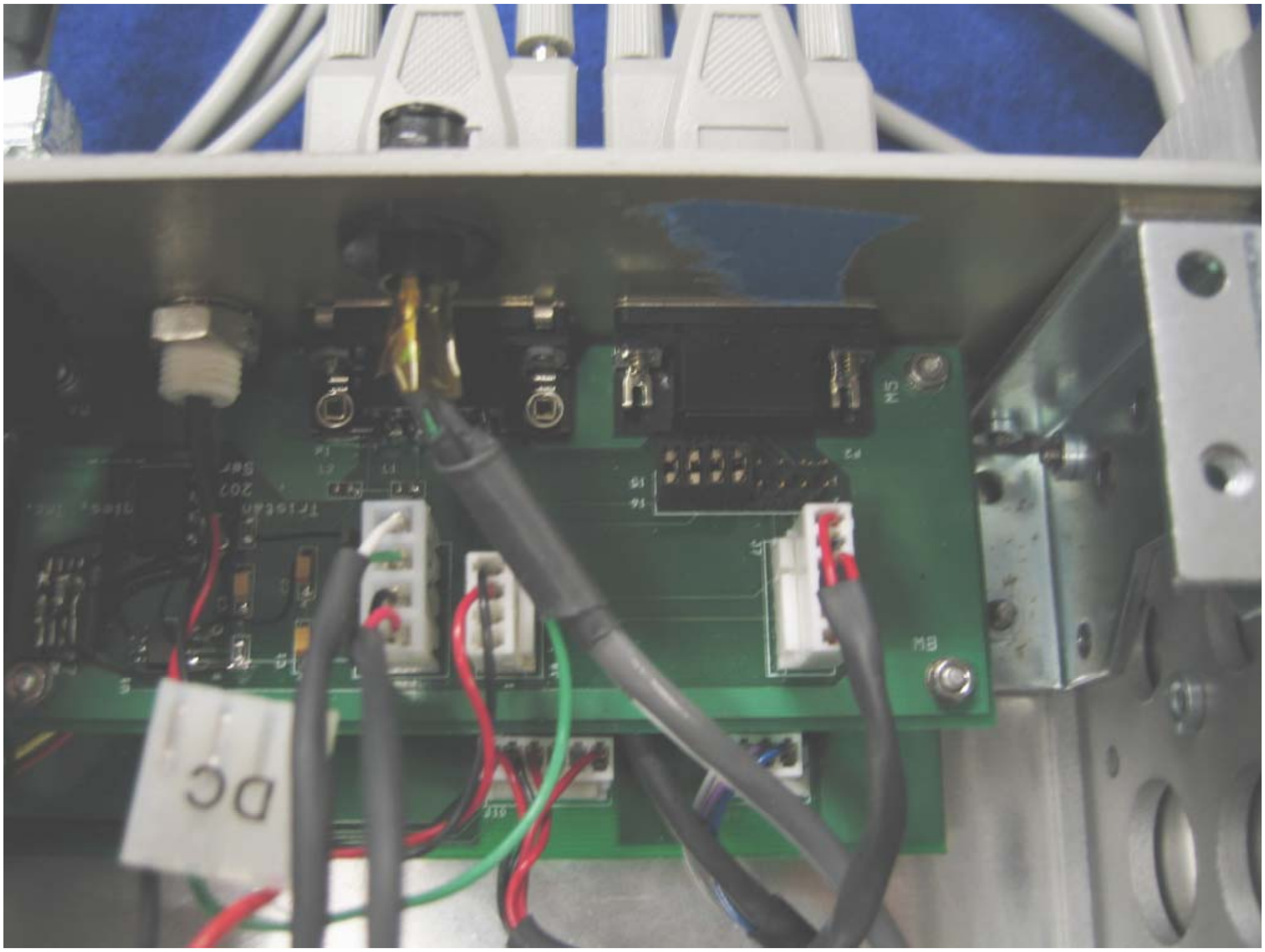

Figure 4: iPS-400 jumpers configured for RS-485 interface

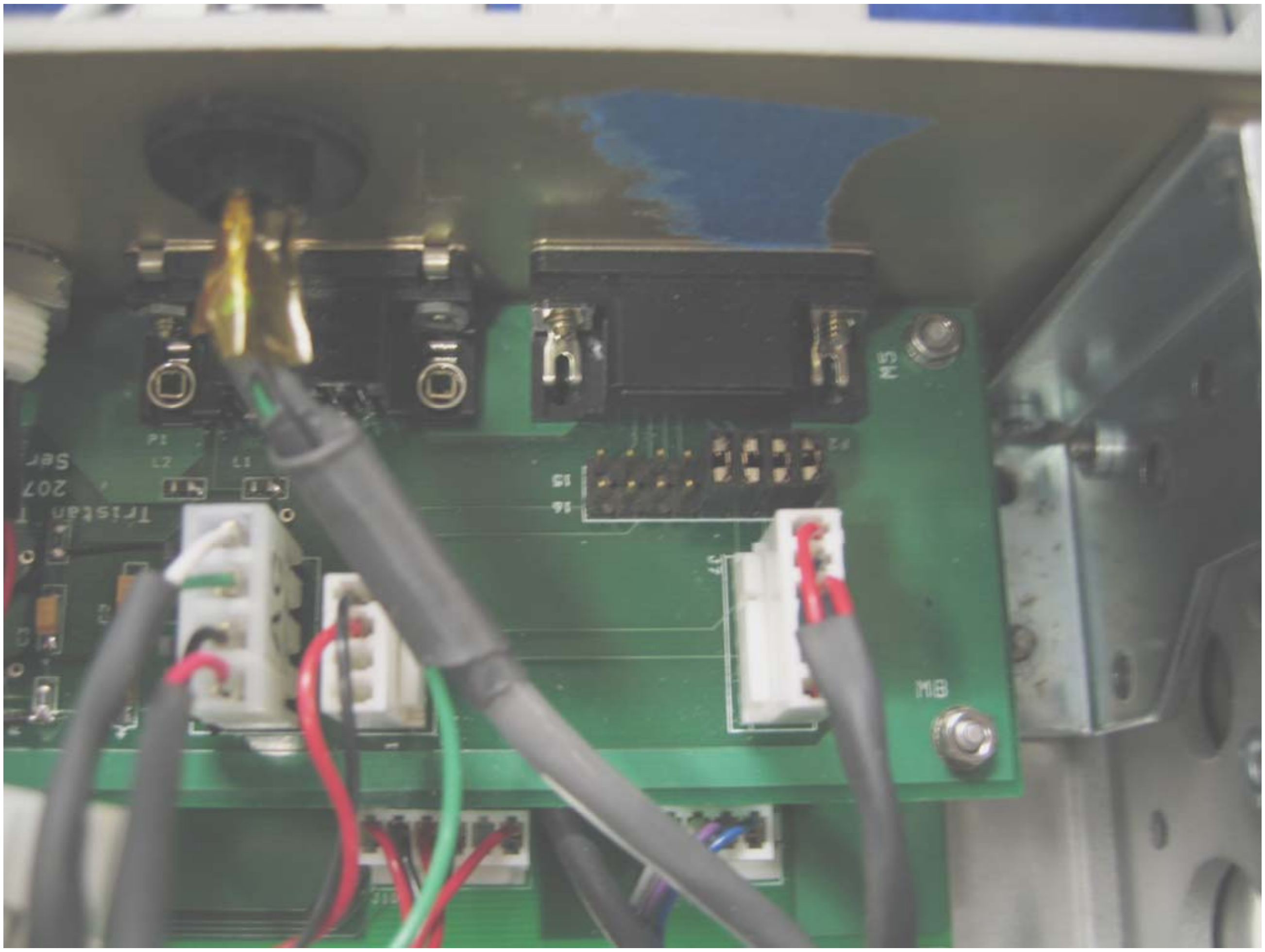

Figure 5: iPS-400 jumpers configured for RS-232 interface

The iPS-400 can also be configured to operate off of ±6 V dc power. Current required is ~0.65 amp/iMC-404 module and must have < 1 mV ripple. Please contact Tristan for details. If the optional external DC power capability is ordered with the system, the iPS-400 will be able to operate on either external DC or AC power. To switch between the two power sources, connections must be changed inside the enclosure as shown in the figures below.

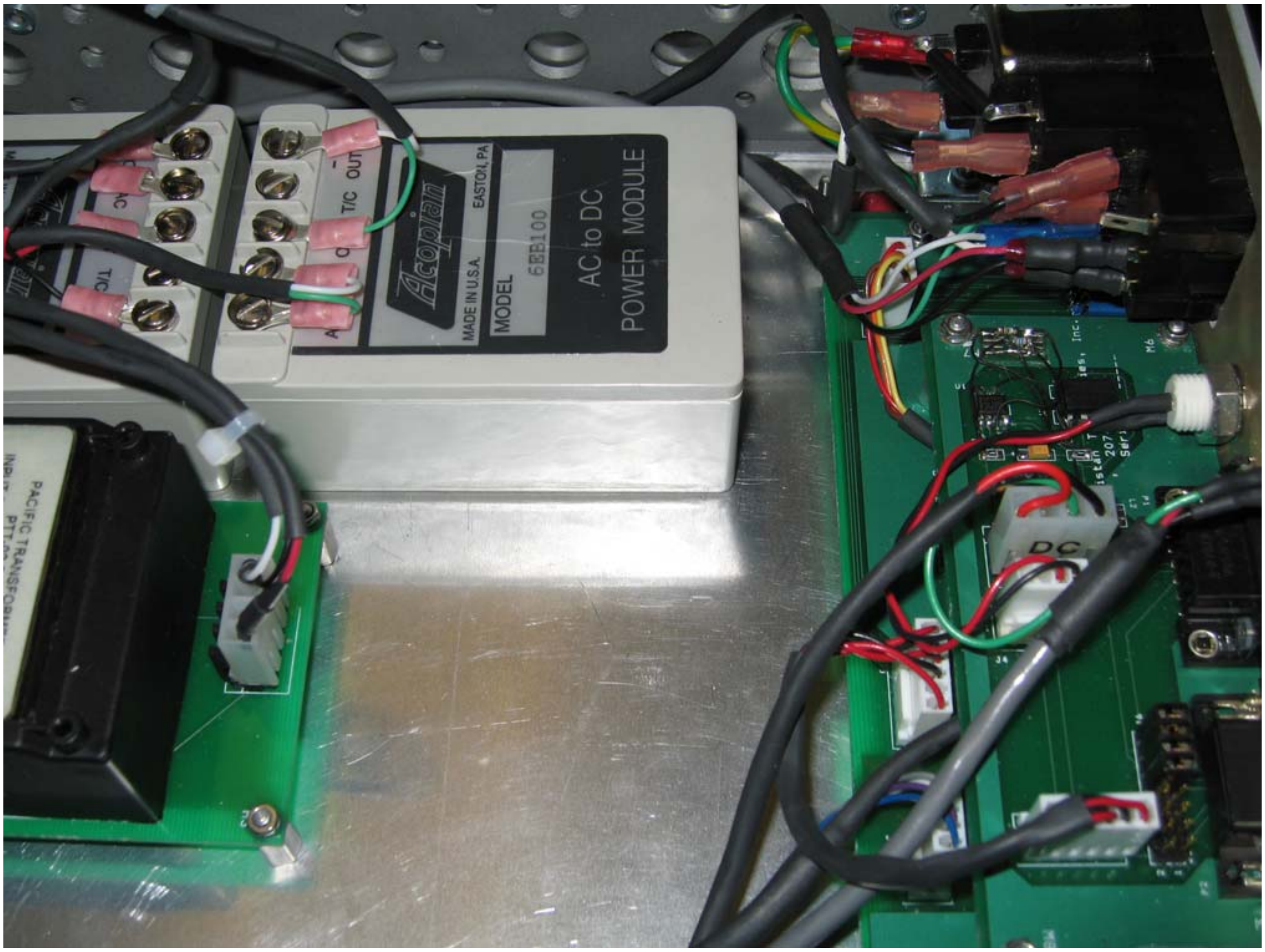

Figure 6: iPS-400 connected for external DC power

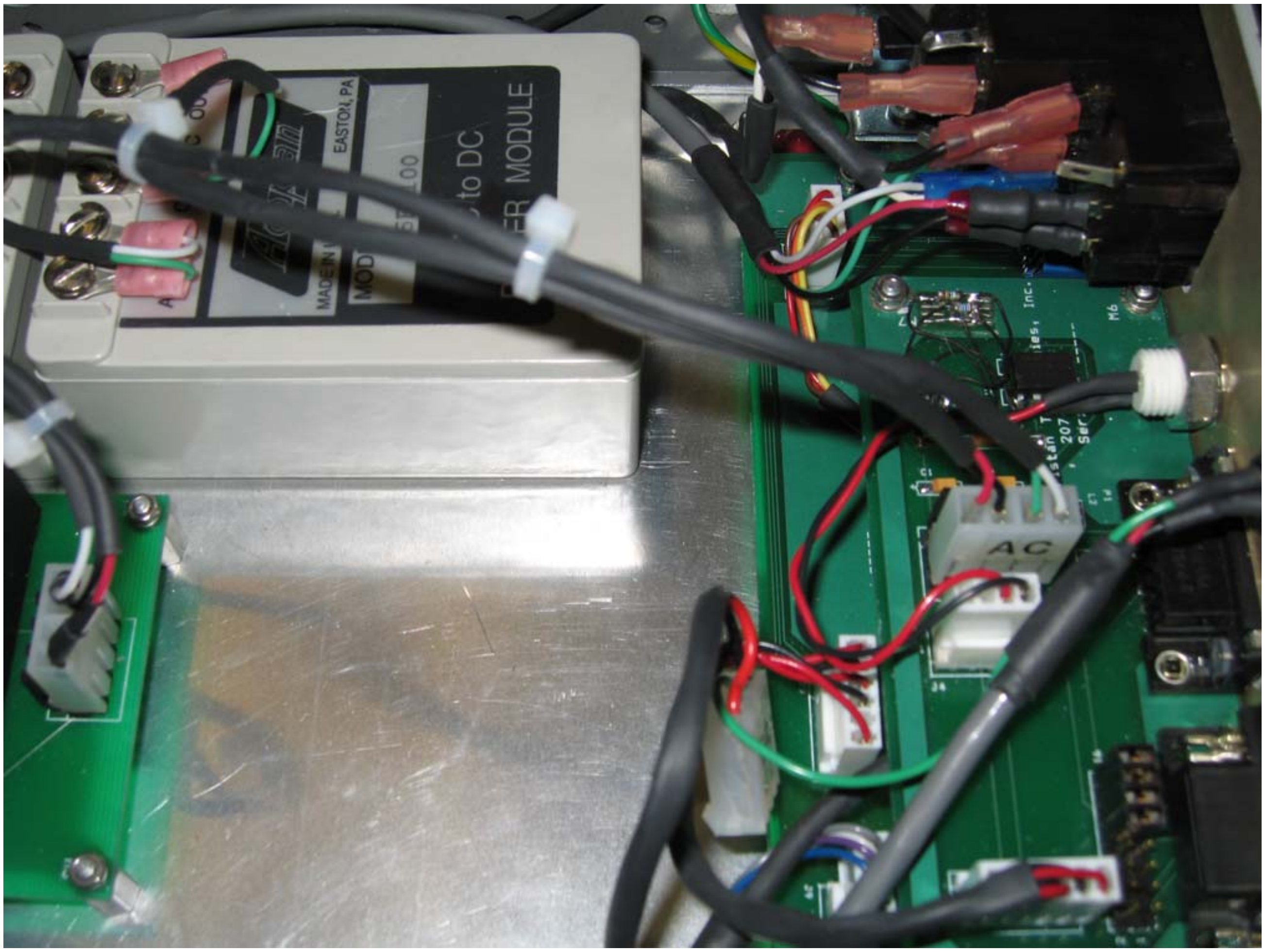

Figure 7: iPS-400 connected for AC mains input power

System Architecture

The SQUID control logic is shown in Figure 8. There are four units to a control section (Figure 8). The bandwidth is designed to be 50 kHz. The system can be addressed via a RS-485 2-wire network or locally via an RS-232 backdoor. The upgraded firmware supports RS-485 baud rates as high as 57,600, and has a hard-wired RS-232 baud rate of 9600. There is provision for both a low pass and high pass hardware filter in addition to 3 gain ranges, offset, and slew ranges. The system output is a differential output. The clocks are all configured as one master rest slave between boards. The addressing on 485 is via a node address and command. The RS-485 is 2-wire half-duplex mode. The microcontroller has provision for several autotune modes. The nodes are factory set.

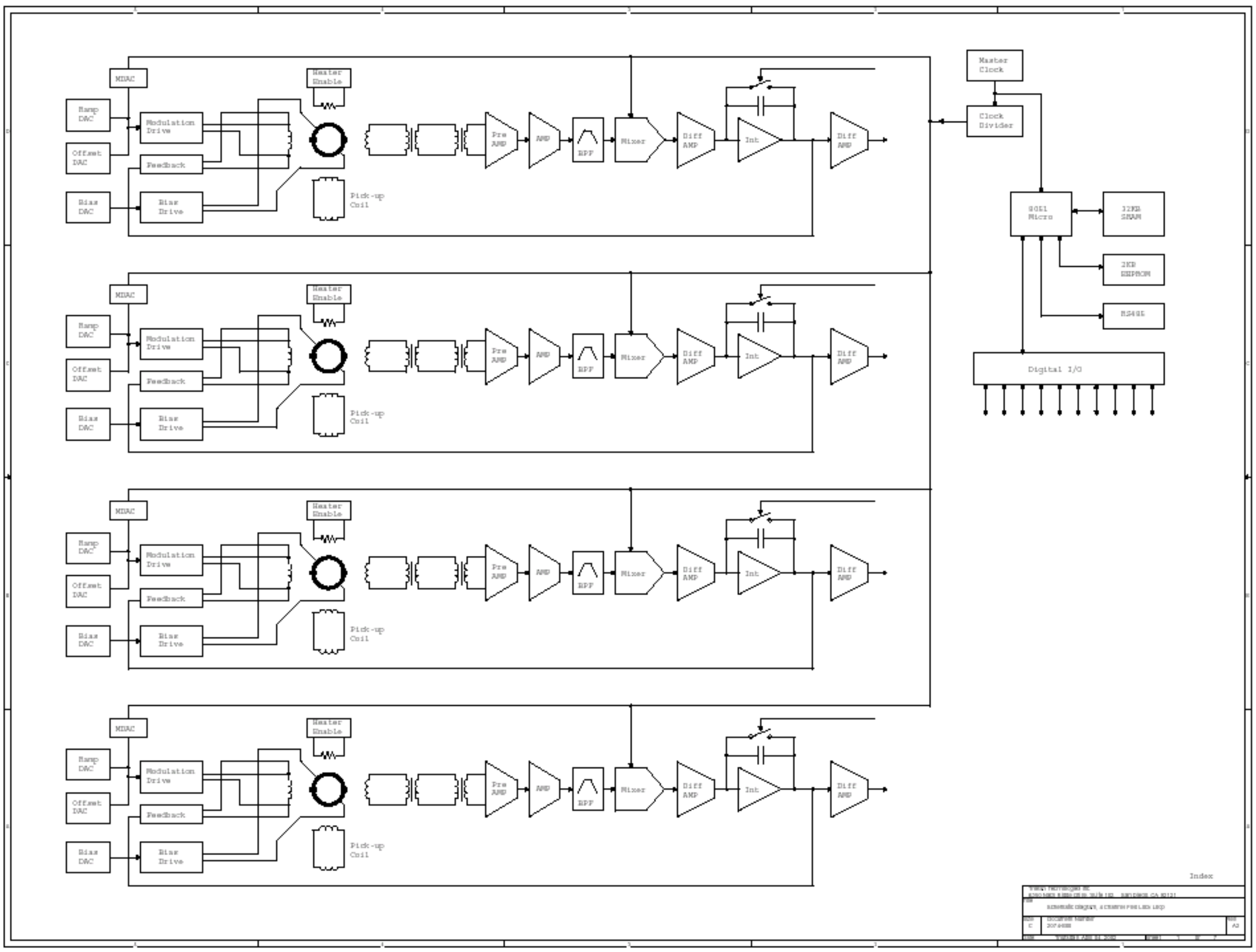

Figure 8: System Hardware Design

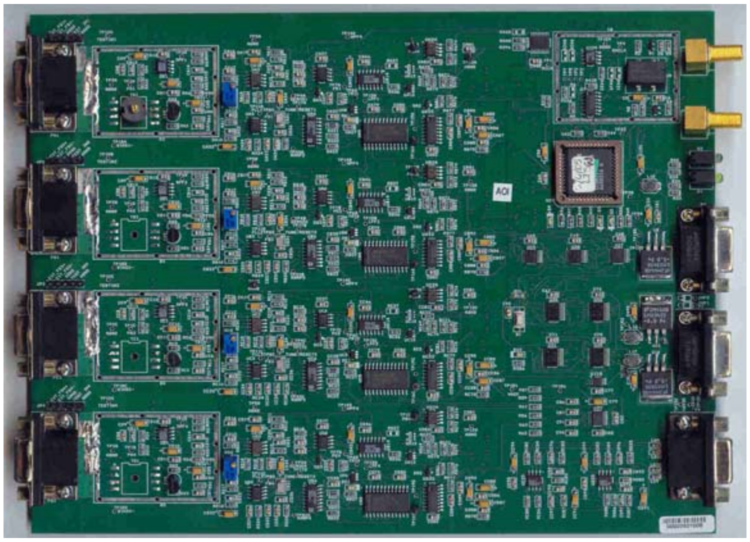

Figure 9: iMC-404 circuit board (7½ " x 9¼ ")

System Cabling

SQUID Cables, CH1 to CH4, Header Connectors on PCB

P1 = SQUID CH1	
Pin	Function
1	SQUID_SigIn-_CH1
2	SQUID_SigIn+_CH1
3	ModSig-_CH1
4	ModSig+_CH1
5	BiasSig-_CH1
6	BiasSig+_CH1
7	HEATER-_CH1
8	HEATER+_CH1
9	Ext_Feedback-_CH1
10	Ext_Feedback+_CH1

P2 = SQUID CH2	
Pin	Function
1	SQUID_SigIn-_CH2
2	SQUID_SigIn+_CH2
3	ModSig-_CH2
4	ModSig+_CH2
5	BiasSig-_CH2
6	BiasSig+_CH2
7	HEATER-_CH2
8	HEATER+_CH2
9	Ext_Feedback-_CH2
10	Ext_Feedback+_CH2

P3 = SQUID CH3	
Pin	Function
1	SQUID_SigIn-_CH3
2	SQUID_SigIn+_CH3
3	ModSig-_CH3
4	ModSig+_CH3
5	BiasSig-_CH3
6	BiasSig+_CH3
7	HEATER-_CH3
8	HEATER+_CH3
9	Ext_Feedback-_CH3
10	Ext_Feedback+_CH3

P5 = SQUID CH4	
Pin	Function
1	SQUID_SigIn-_CH4
2	SQUID_SigIn+_CH4
3	ModSig-_CH4
4	ModSig+_CH4
5	BiasSig-_CH4
6	BiasSig+_CH4
7	HEATER-_CH4
8	HEATER+_CH4
9	Ext_Feedback-_CH4
10	Ext_Feedback+_CH4

Clock Cables, Clock-IN and Clock-OUT, SMA Connectors on PCB

J1 = Clock INPUT	
Pin	Function
CTR	Clock Signal Input
Shield	GND_Analog

J2 = Clock OUTPUT	
Pin	Function
CTR	Clock Signal Output
Shield	GND_Analog

Power / Communication Cable, D-sub 9 Pin Female Connector on PCB

P6 and P7 = POW/COMM	
Pin	Function
1	SysPowIn_+6V
2	GND_Digital
3	RS485_-
4	_Control_Digital-Off
5	_Control_Sys-Reset
6	SysPowIn_-6V
7	RS485_+
8	GND_Digital
9	_Control_Int-Reset

Analog Output Cable, D-sub 15 Pin HD Female Connector on PCB

P4 = Analog Output	
Pin	Function
1	PC_RS232_TxD
2	LoopOut-_CH1
3	LoopOut-_CH2
4	LoopOut-_CH3
5	LoopOut-_CH4
6	PC_RS232_RxD
7	LoopOut+_CH1
8	LoopOut+_CH2
9	LoopOut+_CH3
10	LoopOut+_CH4
11	GND_Digital_RS232
12	GND_Analog_CH1
13	GND_Analog_CH2
14	GND_Analog_CH3
15	GND_Analog_CH4

Figure 10: System Cables

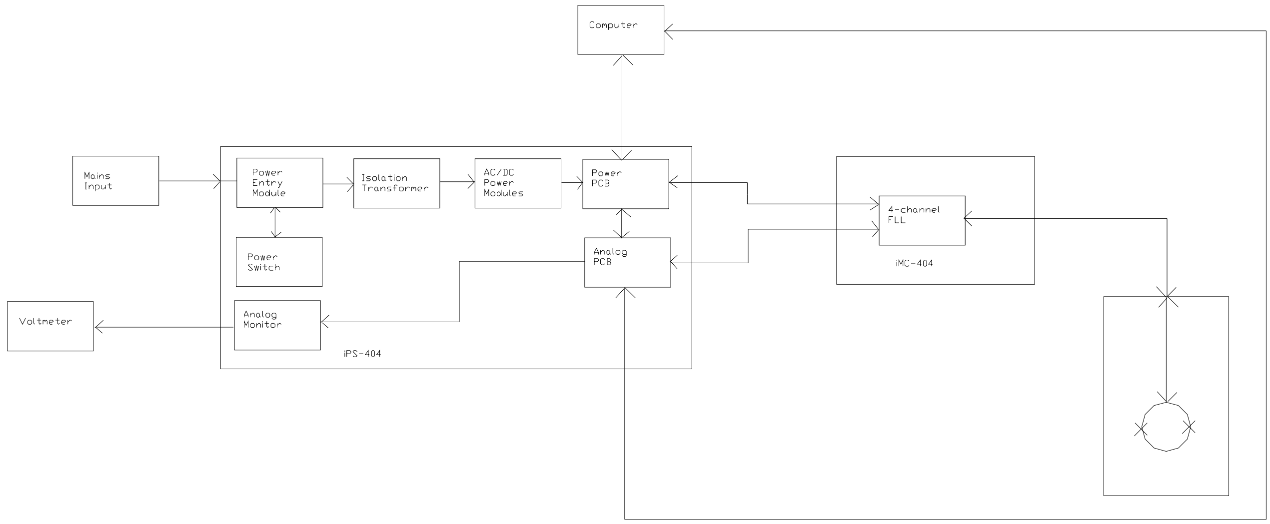

Figure 11: System Cables Connections

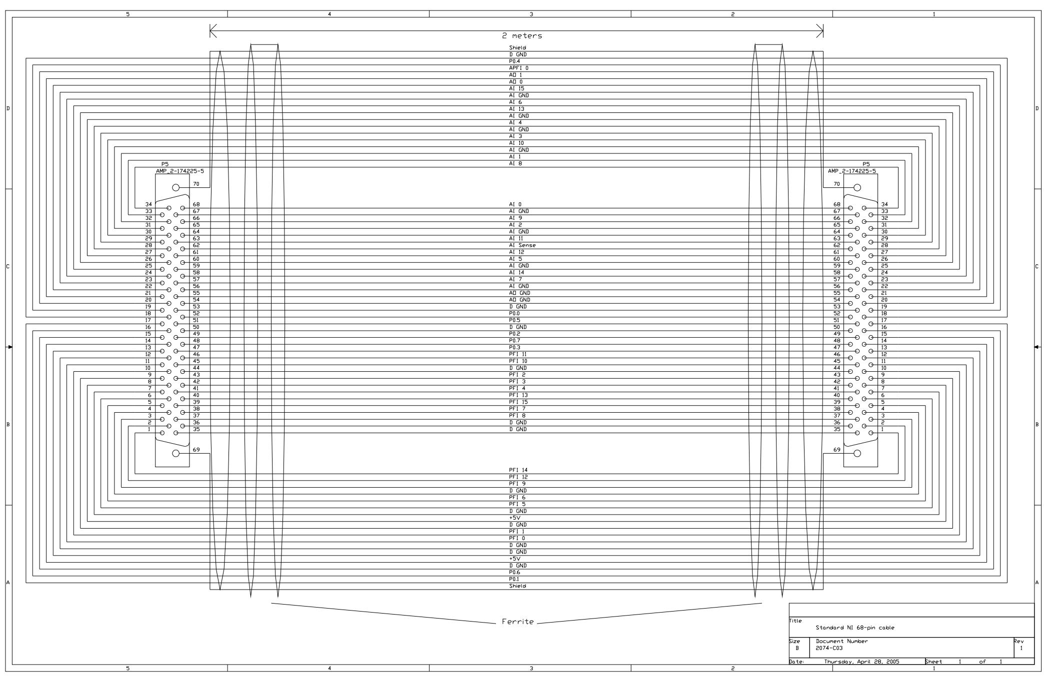

Figure 12: System 68 pin Cable Connections (supplied with optional data acquisition system)

Data Acquisition Hardware

Normally, data acquisition utilizes either 16-bit or 24-bit LabView compatible data acquisition hardware. A separate users manual is supplied to describe the specific computer and data acquisition hardware supplied (if any) with your system. Examples of specific software commands are shown in the next section.

Software Console Commands

```
eSim     , "SIMULATE" , "Launch Host Simulate mode [SIM:]" }
eProgram , "PROGRAM"  , "Launch Factor Program mode [PRG:]" }
eList    , "@LIST"    , "List Comsole Commands" }
```

Figure 13: List of Software Console Commands

Introduction

This program has built in tools for performing low-level functions or simulations for the user. This collection of tools is called the "**Console**" and is evoked by the external UART. The "Console" is, fundamentally, a command line interpreter using free form input of words and values separated by spaces or commas. The "Console" is not a diagnostic although it can lend itself to troubleshooting or evaluation.

"**Console**" supports three basic features, binary operations, low-level functions and remote command simulation. Binary operations are primitives because they operated at the lowest level of accessing or operating a device. The low-level functions evoke an operational feature. The remote command simulation allows the user to perform the same commands a host computer in a local mode.

The "**Console**" provides the following set of commands...

- SIMULATE local mode
- PROGRAM local mode

If the user enters any other text that is not one of the above commands, "Console" will present the error indication "**???**".

Each command has a format, which will be presented later. The format is generally the same for each command. The general difference is the number of parameters. The number of parameters is dependent on the command and the target device.

Each command can be entered with the minimum set of characters what will distinguish the command. The following presents the minimum text to enter for each command…

- Program … "p"
- Simulate … "s"

Software Commands RS485 mode

Introduction

This document provides more details on the interface format for the RS-485 transport layer. RS-485 only defines the hardware or electrical layer. RS-485 does not specify a protocol or data format.

This format and protocol is a convention designed by or for Tristan Technologies. It packages the interface dialog information for FLL to host computer communications.

This document represents the design or design goals for the software on the Tristan Technologies FLL unit and more specifically to the communications interface. It is subject to changes additions or deletions as required or agreed upon by relevant parties.

Protocol

By convention the FLL will act as a slave unit to a host system. The PC will be considered the common host system and will serve as the network controller. Other computers or intelligent devices may serve as network controller.

The host will initiate all communications. The communications will be in binary. The host will communicate using packets. The packets are a simple structure described in the “FORMAT” section.

Each FLL is a device that contains, at a minimum, one FLL channel but may contain any number of FLL channels. This design initially targets a 4-channel FLL device.

Each FLL device has a unique serial number. The network controller may use the broadcast command along with the serial number to program the device with node ID’s. For Tristan Technologies systems, the Node ID’s are pre-programmed prior to shipment, and do not necessarily correspond to the FLL channel number. This should be mapped.

The communications will be in the form of a command or request. The command is any communication that directs the FLL to perform an operation or function. A request is any communication that directs the FLL to return any status or data.

There are three types of commands, Broadcast, Group and Node. There are only two types of request, Broadcast and Node. The response of the FLL will depend on the type of command or request as follows...

Broadcast Command:	There are no replies to the broadcast command. If all nodes responded, there would be collision on the network.
Group Command:	There are no replies to group commands.
Node Command:	There is always a one-byte reply to the node command. That reply is the echo of the Node ID and the error status bit.
Broadcast Request:	The FLL device can respond to a broadcast request that targets a device. Each device has a serial number and can reply like a normal node Request.
Group Request:	Not a capability!
Node Request:	The reply to a request will be any number of bytes starting with the echo of

the Node ID and the error status bit followed by the status or data stream.

The FLL will not respond to a command or request under certain error conditions. All errors are posted internally in the FLL and the error status bit is always until the errors are cleared.

A maximum of 10 errors are recorded. The last error code will indicate if the error stack was overrun. The errors are cleared by a request for error status.

Format

The format for the Command and the Request is the same. The host will send a packet that contains a node number, a data count, the body and a checksum. The body will contain the command or request and supporting parameters. The size of the body depends entirely on the command or request.

The format of the Reply will be similar to format for the Command or Request. The only exception will be the error status bit in the node byte. The size of the body depends entirely on the reply information, status or data.

The following pages will outline and expand upon the command or request formats.

Legend:

Node ID: Any number from 10 to 120. (9 < Node < 121)

Count: Any 8 bit value (0 <= Count <= 255). For our application, should not exceed 5.

Cmd: And valid code listed in command or request table.

Code: Any 8 bit value (0 <= Code <= 255). Meaning of “Code” depends on “Cmd”.

Value: Any 16 bit value (0 <= Value <= 65535). Meaning of “Value” depends on “Cmd”.

Reply: A combination of the “Node” number & the error status in the most significant bit (bit 7).

Error Codes:

0: No errors.

10: Bad packet checksum.

11: Protocol error. (packet size is too large)

12: Invalid command or request code.

13: Invalid parameter.

14: Command Time-out.

15: Command failed.

16: Initial tune failed.

17: Auto-tune failed.

18: Retune failed.

255: Buffer over-run error.

General Format

1 byte	Node ID … number between 10 and 249. (the other numbers are reserved.
1 byte	Count … This is the size of the body. (typically this will be 1 to 5)
1 to 5 bytes	Body … This depends on how many parameters and parameter size.
2 bytes	Checksum … The arithmetic sum of the ASCII code for each character on the body.

Format – Command With No Parameters

1 byte	Node ID
1 byte	1 (count)
1 byte	Command.
2 bytes	Checksum

Format – Command With One 8 Bit Parameter

1 byte	Node ID
1 byte	2 (count)
1 byte	Command.
1 bytes	8 bit parameter value.
2 bytes	Checksum

Format – Command With Two 8 Bit Parameters

1 byte	Node ID
1 byte	3 (count)
1 byte	Command.
1 bytes	8 bit parameter value.
1 bytes	8 bit parameter value.
2 bytes	Checksum

Format – Command With One 8 & One 16 Bit Parameters

1 byte	Node ID
1 byte	4 (count)
1 byte	Command.
1 bytes	8 bit parameter value.
2 bytes	16 bit parameter value.
2 bytes	Checksum

Format – Command With One 16 Bit Parameter

1 byte	Node ID
1 byte	3 (count)
1 byte	Command.
2 bytes	16 bit parameter value.
2 bytes	Checksum

Format – Command With Two 16 Bit Parameters

1 byte	Node ID
1 byte	5 (count)
1 byte	Command.
2 bytes	16 bit parameter value.
2 bytes	16 bit parameter value.
2 bytes	Checksum

Another structure for presenting the above format.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8

Format – Command With No Parameters

Node ID	1	Cmd/Qry	Checksum

Format – Command With One 8 Bit Parameter

Node ID	2	Cmd/Qry	Code (8)	Checksum

Format – Command With Two 8 Bit Parameters

Node ID	3	Cmd	Code (8)	Code (8)	Checksum

Format – Command With One 8 & One 16 Bit Parameters

Node ID	4	Cmd	Code (8)	Value (16)	Checksum

Format – Command With One 16 Bit Parameter

Node ID	3	Cmd	Value (16)	Checksum

Format – Command With Two 16 Bit Parameters

Node ID	5	Cmd	Value (16)	Value (16)	Checksum

Format – Command Reply

1 byte	Reply ... Node + Error status.

Format – Request Reply

1 byte	Reply ... Node + Error status.
1 byte	count
1 to n byte	status or data.
2 bytes	Checksum

Another structure for presenting the above format.

Format – Command Reply

Node

Format – Request Reply

Node	Count	status	. . . .	Checksum

List of Commands & Codes

Command		Size	Code	# 8 bit Params	# 16 bit Params	Command Description
	ARstOff (10)	1	10			Disable Auto Reset
	ARstOn (11)	1	11			Enable Auto Reset
	Atune (12)	5	12		2	Auto Tune From Current, heat and cool in msec
	Bias (13)	2	13	1		Set Bias DAC in Counts (0-255)
	Bias+ (14)	2	14	1		Increment Bias DAC by desired amount (0-255)
	Default (15)	1	15			Load default settings
	Gain (16)	2	16	1		Set Gain amplification to 1, 10, or 100 (1,2, or 3)
	Heat (17)	3	17		1	Heat for desired milliseconds
	Hpass (18)	2	18	1		Turn High-pass filter on or off (1 or 0)
	%$Init (19)	1	19			Save Serial, Node, and Group numbers
	Itune (20)	5	20		2	Auto Tune from Initial Settings, heat & cool in msec
	Last (21)	1	21			Load last settings
	Lpass (22)	2	22	1		Turn Low-pass filter on or off (1 or 0)
	Mod (23)	2	23	1		Set Mod Pot in Counts (0-255)
	Mod+ (24)	2	24	1		Increment Mod Pot by desired amount (0-255)
	Mode (25)	2	25	1		Set FLL channel to run or tune (1 or 2)
	Offset (28)	2	28	1		Set Offset DAC in Counts (0-255)
	Offset+ (29)	2	29	1		Increment Offset DAC by desired amount (0-255)
	Reset (31)	3	31		1	Close feedback integrator for desired msec
	ResetHold (32)	1	32			Hold feedback integrator closed
	ResetStop (33)	1	33			Release feedback integrator
	Retune (34)	5	34		2	Check if a tune is needed, heat & cool in msec
	%$Save (35)	1	35			Save channel settings to default area of EEPROM
	Save (36)	1	36			Save channel settings to last area of EEPROM
(255)	Set (37)	5	37	2	1	Set Node (11-120) or Group (131-230) for FLL unit with matching serial number
	Baud (43)	2	43	1		Set 485 Baud rate (0-6) for (1.2k, 2.4k, 4.8k, 9.6k, 19.2k, 38.4k, or 57.6k)

Figure 14: List of RS-485 Commands and Codes

List of Requests & Codes

Request		Size	Code	# 8 bit Params	# 16 bit Params
Amp? (50)		1	50		
Bias? (51)		1	51		
Current? (52)		1	52		
Default? (53)		1	53		
FLLOut? (54)		1	54		
Hpass? (55)		1	55		
ID? (56)		1	56		
Last? (57)		1	57		
Lpass? (58)		1	58		
Mod? (59)		1	59		
Mode? (60)		1	60		
Offset? (61)		1	61		
(255) Poll? (62)		3	255		1
Reset? (63)		1	63		
Serial? (64)		1	64		
Baud? (67)		1	67		
Gain? (68)		1	68		
Error (69)		1	69		
Errors (70)		1	70		

Figure 15: List of RS-485 Requests and Codes

List of Replies & Codes

Request		Reply	Size	Code	Status or Data
Amp? (50)		Node ID	3	50	1 - 16 Bit: Tuning Amplitude
Bias? (51)		Node ID	2	51	1 - 8 Bit: Bias DAC Setting
Current? (52)		Node ID	14	52	6 - 8 Bit: Gain, Hpass, Lpass, Mode, Reset, Slew 7 - 8 Bit: Atune, ITune, Rtune, Bias, Mod, Offset, Skew
Default? (53)		Node ID	14	53	6 - 8 Bit: Gain, Hpass, Lpass, Mode, Reset, Slew 7 - 8 Bit: Atune, ITune, Rtune, Bias, Mod, Offset, Skew
FLLOut? (54)		Node ID	3	54	1 - 16 Bit: Output Amplitude
Hpass? (55)		Node ID	2	55	1 - 8 Bit: Hpass State
Last? (57)		Node ID	14	57	6 - 8 Bit: Gain, Hpass, Lpass, Mode, Reset, Slew 7 - 8 Bit: Atune, ITune, Rtune, Bias, Mod, Offset, Skew
Lpass? (58)		Node ID	2	58	1 - 8 Bit: Lpass State
Mod? (59)		Node ID	2	59	1 - 8 Bit: Mod Pot Setting
Mode? (60)		Node ID	2	60	1 - 8 Bit: Mode Setting (1=Run; 2=Tune)
Offset? (61)		Node ID	3	61	1 - 8 Bit: Offset DAC Setting
(255) Poll? (62)		Node ID	-	-	-
Reset? (63)		Node ID	2	63	1 - 8 Bit: Reset Latch State
Serial? (64)		Node ID	3	64	1 - 16 Bit: Device Serial # at Node
Baud? (67)		Node ID	2	67	1 - 8 Bit: Baud Rate Setting (0 to 6)
Gain? (68)		Node ID	2	68	1 - 8 Bit: Gain State (1, 2, or 3)
Error (69)		Node ID	3	69	2 - 8 Bit: error cmd, error code
Errors (70)		Node ID	n	70	n - 8 Bit: error cmd, error code; error cmd, error code; …

Figure 16: List of RS-485 Replies and Codes

Software Commands RS232 SIM mode

The 4 channel units can be addressed via a backdoor on the 15-pin connector. A list of these commands is shown below. Note that required input fields are space delimited.

"ARSTOFF" , "Disable Auto Reset" },

"ARSTON" , "Enable Auto Reset" },

"ATUNE" , "Perform AutoTune with DEFAULT settings after heating for 0<=N<=65536 millisec, and cooling for 0<=N<=65536 millisec" },

"BIAS" , "Set Bias DAC to 0<=N<=255 counts" },

"BIAS+" , "Increment Bias DAC by -128<=N<=127 counts" },

"BAUD" , "Select New Baud Rate (1-7)" },

"DEFAULT" , "Load EEPROM DEFAULT Settings to CURRENT RAM Settings" },

"GAIN" , "Set Gain to 1(1), 10(2) or 100(3)" },

"HEAT" , "Turns on heater for 1<=N<=65536 millisec " },

"HPASS" , "Turns High Pass Filter ON(1) or OFF(0)" },

"%$INIT" , "Stores Factory settings" },

"ITUNE" , "Perform AutoTune with initial settings: Bias=0, MOD=40 after heating for 0<=N<=65536 millisec, and cooling for 0<=N<=65536 millisec" },

"LAST" , "Load EEPROM LAST Settings to CURRENT RAM Settings" },

"LPASS" , "Turns Low Pass Filter ON(1) or OFF(0)" },

"MOD" , "Set Mod DAC to 0<=N<=255 counts" },

"MOD+" , "Increment Mod by -128<=N<=127 counts" },

"MODE" , "Put FLL in RUN(1) or TUNE(2) mode" },

"MICROOFF" , "Turns Power OFF to FLL Micro-Processor" },

"OFFSET" , "Set Offset to 0<=N<=255 counts" },

"OFFSET+" , "Increment Offset by -128<=N<=127 counts" },

"OFFSETAUTO" , "Autozeros the output (if possible)" },

"RESET" , "Holds Integrator Reset for 1<=N<=65536 millisec" },

"RESETHOLD" , "Holds Integrator Reset until 'ResetStop'" },

"RESETSTOP" , "Releases Integrator Reset" },

"RETUNE" , "Performs a Retune with Current Settings" },

"%$SAVE" , "Save CURRENT RAM Settings to EEPROM DEFAULT Settings" },

"SAVE" , "Save CURRENT RAM Settings to EEPROM LAST Settings" },

"SET" , "Set Unit with Serial Number S at Channel C to Node N" },

"SKEW" , "Set Skew to 0<=N<=255 counts" },

"SKEW+" , "Increment Skew by -128<=N<=127 counts" },

"SLEEP" , "Put FLL Micro-Processor into Sleep mode" },

"SLEW" , "Set slew to Slow(1), Normal(2) or Fast(3)" },

"WAKE" , "Wake up Micro-Processor" },

"AMP?" , "Return Peak Triangle Amplitude" },

"BIAS?" , "Return current Bias setting" },

"BAUD?" , "Return current BAUD Rate setting" },

"CURRENT?" , "Return CURRENT settings (from RAM)" },

,"DEFAULT?" , "Return DEFAULT settings (from EEPROM)" },

"ERROR?" , "Return next recorded error" },

, "ERRORS?" , "Return all recorded errors" },

"FLLOUT?" , "Return Analog Triangle Amplitude" },

"GAIN?" , "Return current Gain setting" },

"HPASS?" , "Return current State of High Pass Filter" },

"ID?" , "Return all FLL Unit values stored at factory" },

"LAST?" , "Return LAST settings (from EEPROM)" },

"LPASS?" , "Return State of Low Pass Filter" },

"MOD?" , "Return current Mod setting" },

"MODE?" , "Return 1 if in RUN mode and 2 if in TUNE mode" },

"OFFSET?" , "Return Offset DAC setting" },

"POLL?" , "Return Node/Channel Number of Unit with Serial#=N" },

"RESET?" , "Return State of Reset Latch and Clears It" },

"SERIAL?" , "Return Serial# from unit of Node (Channel)" },

"SKEW?" , "Return current Skew setting" },

Figure 17: List of RS-232 commands in SIM mode