advanced surgical systems, inc.

RF Generator

500-020

Users Guide

Advanced Surgical Systems, Inc.
5030 East Calle Barril
Tucson, Arizona 85718
www.Advanced-Surgical.com

Contents

Section

1

Safety Information

WARNING

The *RF Generator 500-020* is not intended for use on human subjects. *Advanced Surgical Systems, Inc.* does not recommend nor authorize the use of the *RF Generator* on human subjects. This device does not currently have the necessary regulatory approvals for patient use.

Built in Safety Measures

The network of distributed control within the generator allows for a high degree of safety resulting from tight control of system operation. As the microcontrollers are directly integrated into the feedback network of the amplifiers, forward and reflected power can be measured and compared to preset limits hundreds of times each second. If the measured power deviates beyond the preset limits, the microcontroller either automatically and immediately shuts down the amplifier channel or prompts you for intervention. As each channel is independently controlled, all channels do not need the same limits nor respond to over-limit conditions with the same action. The microcontrollers and the embedded controller include watchdog timers to prevent unwanted power delivery. Communication integrity is maintained by the power enable timer (see “Power Enable Timer” on page 5) which will shut the system down if another command is not received within 15 seconds.

Precautions

The precautions are listed below:

Power Source

Power requirements are listed in the specification sheet provided with this system.

Grounding the RF Generator

The *RF Generator* is grounded through the power cord. Use the power cord supplied with the system. Ensure that the power source has the proper grounding and is connected.

500-020 Breaker Switch

To avoid fire hazard, the system is equipped with a circuit breaker. Make sure that the door to the breaker switch remains closed during operation.

Covers

Covers should not be removed due to risk of electrical shock.

Section

2

Introduction

Guide Conventions

The following symbols in this guide call your attention to particular information.

IEC Symbols

Symbol	Meaning
	This symbol indicates hazardous voltages may be present
	If you see this symbol, refer to the specific instruction manual for specific warning or caution information to avoid personal injury or damage to the product.

Safety Terms

CAUTION: Identifies conditions or practices that could result in damage to equipment.

WARNING: Identifies conditions or practices that could result in injury or loss of life

Introduction to the RF Generator

The *RF Generator 500-020* features multiple channels that generate a RF signal necessary to drive ultrasound transducers in a custom frequency range. The *RF Generator* is a self-contained driving system that allows complete control of the amplifiers via a RS-232 computer interface. You can configure and operate the multiple RF channels independently and enable and disable channels through the software. You program the power and frequency for the selected channel, and each channel retains its settings when disabled. The *RF Generator* allows you to independently select a desired output power for each channel. All power levels are reported in watts.

The amplifier is designed to drive 50 Ω loads. While some mismatch is acceptable, it is possible to damage the system by attempting to drive widely mismatched loads.

Basic Operations

To set up a desired frequency and power output, follow these steps:

1. Turn on the AC power. The system will scan for all installed cards.
2. Program the desired frequency for the channel.
3. Program the desired power for the channel.
4. Send the Enable command to the RF Generator.

Theory of Operation

Amplifier Design

The amplifier used in the *RF Generator* is a highly efficient class D/E switching amplifier designed for driving ultrasonic transducers. The amplifier topology provides excellent efficiency but is limited to a narrow output bandwidth because of its resonant characteristics. Since the amplifier is very efficient, it can be compact, with minimal cooling requirements. Each channel delivers RF power to a 50 Ω resistive load with approximately 70% efficiency.

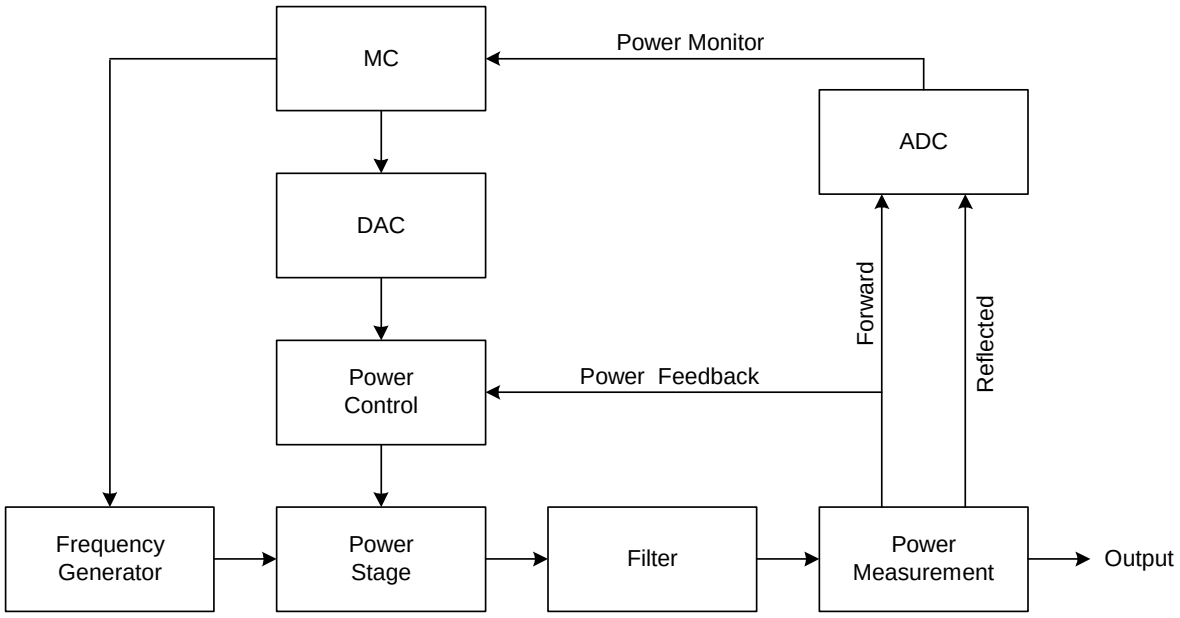


Figure 1 – Block Diagram of Amplifier

The amplifier has control of the output power and the operating frequency of each channel. A microcontroller interfaces with an external computer to monitor the amplifier's operation. The interface computer sends commands to the microcontroller to produce appropriate action. The microcontroller continuously monitors the forward and reflected power.

Advanced Surgical Systems, Inc. has developed a power-leveling scheme that senses the amplifier's forward power and automatically adjusts the output amplitude to compensate for changes in the load impedance. The forward power is the amount of power that the amplifier sends to the load. If there is a perfect match between the amplifier's output and the load (i.e., the load is real with a magnitude of 50), then all of the available power is delivered to the load. If there is any mismatch in impedance, a certain amount of power is reflected back to the amplifier. The amount of power dissipated in the load (forward power - reflected power) is referred to as the transmitted power. While both the forward power and reflected power are measured and monitored by the amplifier, only the forward power is used in the feedback loop. The *RF Generator* is designed to prevent problems that could result from fault conditions in the transducer or associated transmission line.

Power Enable Timer

The power enable timer will disable the amplifier output if the amplifier does not receive a valid command from the host computer within a set time interval. This prevents power output in the case that the host computer has a software failure or loses communication. After the timeout, the amplifier disables the power and displays error 1019. All settings, including set power and individual channel output states, remain unchanged. The timer begins after the Enable On command is sent and is reset each time the amplifier recieves a command from the host computer. The time interval is set using the PwrWatch variable. The value can be changed with **Set PwrWatch <number of ticks>**. The default value for PwrWatch is 273 which corresponds to 15 seconds. There are 18.2 ticks per second, therefore, using a value of 182 would set a time of 10 seconds.

Section

3

Getting Started

This section explains the procedures for unpacking and setting-up the *RF Generator* system.

Procedures for Unpacking and Set Up

1. Carefully unpack the amplifier from the shipping crate.
2. Be sure that the following items are included with the system:
   - 500-020 RF Generator system
   - Serial Null modem cable (Figure 7)
3. Inspect the system for any damage that may have occurred during shipping.
4. Plug the AC power cord to the rear of the *RF Generator* into an appropriate outlet.
5. Switch on the breaker switch, located in an enclosure below the amplifier card rack.
6. To start the system, pull the main power switch on the front panel.

7 Connect a load to the amplifier RF output, as shown below (Figure 2), the output connectors are Cannon DLM-156.

Note: *Some systems may have a different number of output connectors than shown here.*

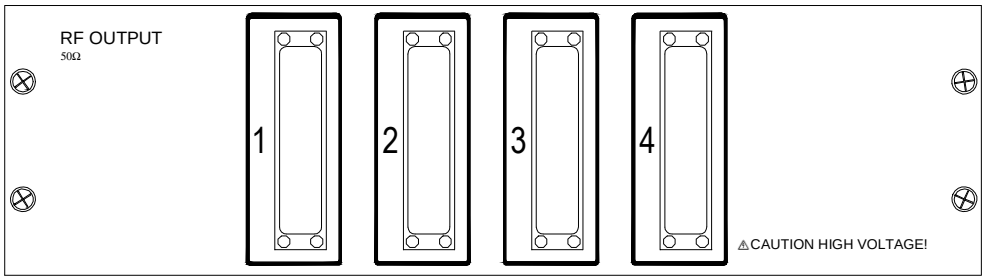


Figure 2 – RF Output

CAUTION: *Do not operate the amplifier without a load connected.*

RF Connector Pinout ITT/Cannon P/N DL1-156P

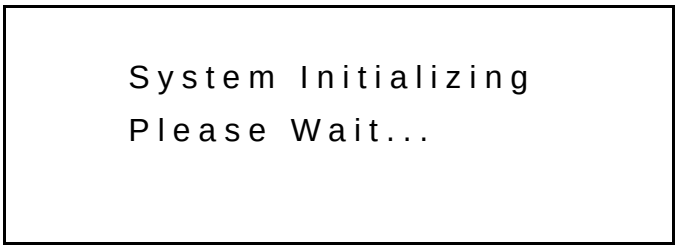
	1	2	3	4	5	6
A	P1	SH2	P3	SH4	P5	SH6
B	SH1	P2	SH3	P4	SH5	P6
C	P7	SH8	P9	SH10	P11	SH12
D	SH7	P8	SH9	P10	SH11	P12
E	P13	SH14	P15	SH16	P17	SH18
F	SH13	P14	SH15	P16	SH17	P18
G	P19	SH20	P21	SH22	P23	SH24
H	SH19	P20	SH21	P22	SH23	P24
J	P25	SH26	P27	SH28	P29	SH30
K	SH25	P26	SH27	P28	SH29	P30
L	P31	SH32				
M	SH31	P32				
N						
P						
R					P33	SH34
S					SH33	P34
T	P35	SH36	P37	SH38	P39	SH40
U	SH35	P36	SH37	P38	SH39	P40
V	P41	SH42	P43	SH44	P45	SH46
W	SH41	P42	SH43	P44	SH45	P46
X	P47	SH48	P49	SH50	P51	SH52
Y	SH47	P48	SH49	P50	SH51	P52
Z	P53	SH54	P55	SH56	P57	SH58
a	SH53	P54	SH55	P56	SH57	P58
b	P59	SH60	P61	SH62	P63	SH64
c	SH59	P60	SH61	P62	SH63	P64

P = Center Conductor
SH = Coax Shield

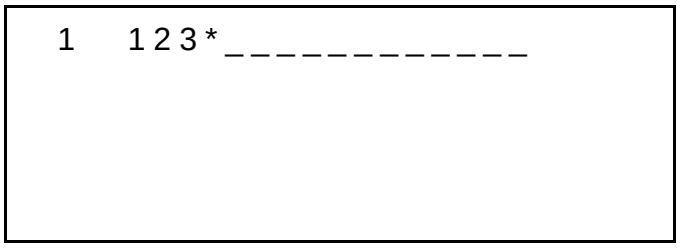
Top View

Figure 3 – RF Connector Pinout

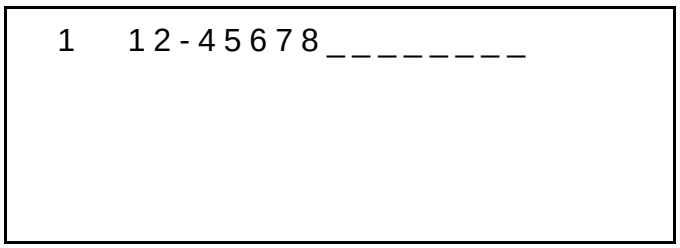
8 Turn on the *RF Generator* by pulling the POWER button outward. The system will go through a startup sequence as shown below.



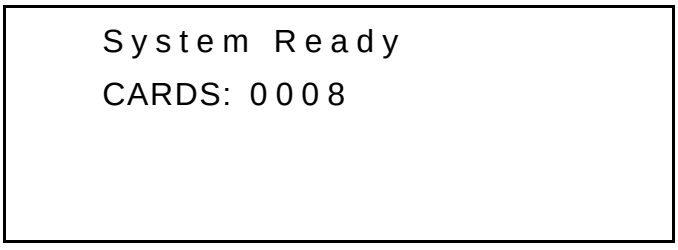
The system scans for cards as shown below.



If the a card is not found, a dash is displayed. In the example below, card3 was not found.



After the system locates the card, the following is displayed.



The System Ready screen (shown above) will display the number of cards installed. The system is now ready for operation.

Note: *For RS-232 operation, a null modem serial cable (Figure 7) must be connected to the RS-232 port on the front panel (Figure 4) just to the right of the power button.*

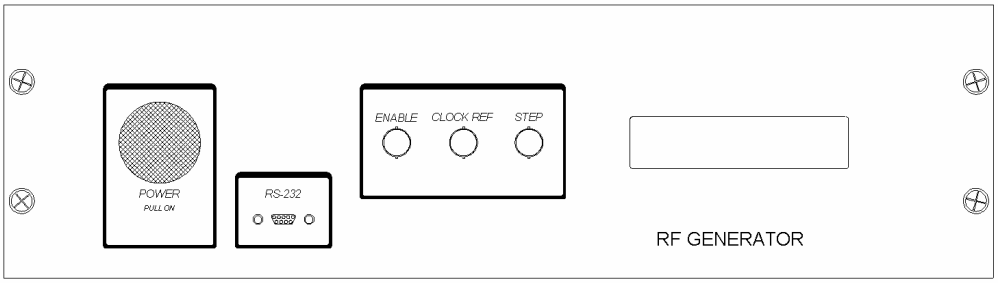


Figure 4 – Front Panel

Front Panel Controls

The Red mushroom switch controls the power to the amplifier system. To turn on, pull out on the switch. It takes approximatels 9 seconds for the system to boot up. Communication between the amplifier system and remote computer is through a Null Modem cable connected to the D-sub 9 connector. The Enable input is active low and can be controlled with a digital signal or using a switch to short the input. The Clock Reference input is not used on this system. The Step input is rising edge triggered.

Procedures for Sample Output

The following list of procedures demonstrates how to output 40 Watts on channel 2 at a frequency of 2.12 MHz. Enter the commands at the command prompt in HyperTerminal after the system has initialized. The example below assumes that a 50 Ω load with an appropriate power rating is connected to channel 2.

Table 1: Sample Commands for Sample Output

Commands	Result
Freq 2 2.12 MHz	Sets the frequency to 2.12 MHz
Power 2 40	Sets the power to 40 Watts
Enable on	Enables the outputs
Output 2 on	Turns on the output for channel 2

Sample Commands Entered from Table 1, “Sample Commands for Sample Output” .

```
Frequency 2 2.12 MHz
OK

Power 2 40
OK

Enable on
OK

Output 2 on
OK
```

Note: *Commands are not case sensitive. Commands are displayed in this manual with specific capitalization to aid in recognizing the compound elements of the command.*

Section

4

Using the Generator Via Computer Interface

Remote operation is accomplished via an RS-232 connection from an external computer. The amplifier will support baud rates of 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200. The RS-232 connection can be made with the provided NULL modem cable (Figure 7) or a custom cable can be made as shown in Figure 7 on page 39. The default baud rate is 115200.

If a command is entered without an argument, the amplifier returns the current value of that parameter. An example query command is provided in this section along with the standard operation example for each command that may act as a query.

Since the amplifier is an RS-232 device, communication can be conducted using terminal emulation software such as Microsoft Windows HyperTerminal.

Set up HyperTerminal to operate with the following settings:

- serial port (for example COM1, or COM2)
- 15200 baud rate
- 8 data bits
- no parity
- 1 stop bit
- set the flow control to None

Once the terminal settings are set properly, the amplifier can be controlled by using the simple English language commands. This mode of control is useful for debugging problems and testing programming issues.

When the amplifier is turned on while connected to the terminal program, the following screen is displayed. RESET indicates that the amplifier is in RESET mode. This occurs both at power on and when the amplifier is reset via software. The next few lines indicate the software title, version, and compile date and time. This information will vary depending on the actual version of the software installed in the amplifier. The time and date shown farther down the screen is the actual time and date as determined by the onboard computer.

The amplifier then runs diagnostic routines to determine that the amplifier is fully functional. If there is a hardware problem, it displays here. The "CARDS <Number of Cards>" line indicates that the amplifier is fully functional and ready to go.

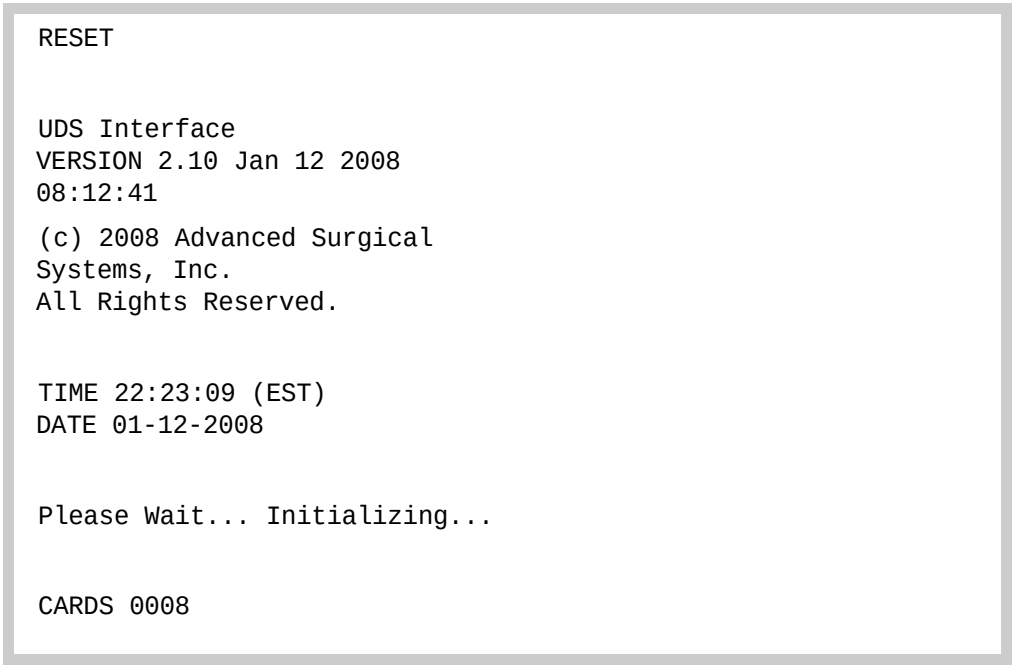


Figure 5 – System ready screen

Now that HyperTerminal is running, you can enter commands at the prompt to control the amplifier.

Note: *If nothing is displayed when typing commands, echo may not be on. To verify that echo is on, type* **set**.

```
Set
  Debug            0
  ACKDelay         5
  Echo             0
  Polling          1
  Retries          3
  Skips            1000
  Stack_int        18
  Calibrate        0
  Baud             115200
  SearchSlots      16
  Ssdelay          300
  PwrWatch         273
OK
```

Figure 6 – Set Commands List

The set command lists all of the system variables. These variables are explained later on in this section. Refer to the value of Echo. If Echo = 0, then type: "set echo 1".

```
Set echo 1
OK
```

Note: *Typed commands will not be visible when echo is set to zero.*

Using RS-232 Commands

This section describes the different commands you can use for different purposes.

Baud <rate>

This command allows the user to change the baud rate that the system uses to communicate via the serial line. Baud rates of up to 115.2 K baud are allowed. To change the baud rate, send the command with the new baud rate, for example, "Baud 115200". After "OK" is received, the system uses the new baud rate, and the user must change the baud rate of the interfacing program.

Default = 115200

Maximum = 115200

Minimum = 1200

Baud Rate	Result
Baud 9600	Sets the-RS-232 baud rate to 9600
Baud	Returns the currently specified baud rate.

Note: *If the baud rate is changed, be sure to remember to change the baud rate in HyperTerminal. It may be necessary to close and then re-open the session for the communication setting in HyperTerminal to be updated.*

Cards

When this command is used, a list of the card addresses in the system rack is returned.

```
Cards
0001
0002
0003
0004
0005
OK
```

CardVer <card|all>

This command returns the version of software running on the specified amplifier card.

```
cardver all
0001 0x21a2
0002 0x21a2
0003 0x21a2
0004 0x21a2
0005 0x21a2
OK
```

Channels <channel|all>

Returns the channel number and the card number for each channel.

ClearLimit <Channel> <Forward|Reflected> <High|Low>

This command clears the over-limit event flag for the specified channel.

Command	Result
`ClearLimit 4 Forward High`	Clears the forward high power limit flag on channel 4

```
ClearLimit 4 Forward High
OK
```

ClearStat <card>

This command clears the status register of the specified card, thus resetting any error codes the card may have generated or any over limit conditions that are currently in effect.

Command	Result
`ClearStat 3`	Clears the status flag for card 3

```
ClearStat 3
OK
```

Disable

This command disables the output of all channels. This command is the same as "Enable off".

```
Disable
OK
```

Enable <on|off>

This command enables or disables all outputs. Enable must be on for any channel to produce an output. When enable is turned off, all outputs are shut off. For a channel to produce power, enable must be on, and the output of the amplifier must also be turned on.

Command	Result
Enable on	Enables the outputs of all amplifier channels
Enable	Returns the current enable status (on or off)

```
Enable on
OK
```

ExtEnable <on|off>

This command activates the "Enable" input on the front panel, which is an active low signal. It is useful creating precisely controlled RF pulses using an external function generator or computer.

Note: *ENABLE must be "on" for EXTENABLE to function. ENABLE supersedes EXTENABLE, and the output will always be disabled unless Enable is on.*

ExtStep <on|off>

The external input “Step” on the front panel of the system allows external control for stepping through the modulation stack. Once the stack has been programmed and activated, the external input must be enabled with ExtEnable. Once enabled, the external input is a simple CMOS level rising edge triggered input.

Frequency <channel|all> <frequency Hz><MHz|kHz>

This command sets the operating frequency for the specified channel. If no frequency is specified, the current setting is returned. The entered frequency must be followed by kHz or MHz.

Command	Result
Freq 2 2.15 MHz	Sets the channel 2 to 2.15MHz
Freq 2	Returns the current frequency on channel 2

```
Freq 2 2.15 MHz
OK

Freq 2
2150000
OK
```

GetPower <channel|all>

This command retrieves the current power settings for the specified channel in watts.

Command	Result
GetPower 2	Requests the currently programmed power for Channel 2

```
GetPower 2
40.50
OK
```

Init

This command causes a quick re-initialization of the system.

LimitDisablePower <on|off>

When enabled the amplifier card will disable the power output and return an error if the power limits are exceeded. When disabled, the amplifier card will leave the power on and return an error. This command affects all cards in the system. The Default is on.

Command	Result
LimitDisablePower on	Enables the LimitDisablePower function

```
LimitDisablePower on
OK
```

LimitEnable <on|off>

This command turns the power limits on or off.

Command	Result
LimitEnable on	Turns the system power limits on

```
LimitEnable on
OK
```

Limits <chan|all> <fwd high|off> <fwd low|off> <refl high|off> <refl low|off>

Limits <chan|all> <FOR|REF> <HIGH|LOW> <pwr limit>

This command can be used to define all of the power limits for a channel or to change a single limit settings on the specified channel.

Command	Result
`Limits 2 35 5 9 off`	Sets the limits for channel 2 as follows: Forward High = 35W, forward low =5W, Reflected High = 9W and turns the Reflected Low power limit off.
`Limits 5 Ref High 10`	Sets the reflected high power limit to 10 watts on channel 5

```
Limits 2 35 5 9 off
OK

Limits 5 Ref High 10
OK
```

MaxFreq <channel|all>

This command returns the maximum frequency for the channel(s) in Hertz. This command allows a user interface to query the system for the proper maximum frequency. The maximum frequency is a value stored in non-volatile memory in the amplifier card.

Command	Result
MaxFreq 2	Requests the maximum frequency of the specified channel

```
MaxFreq 2
2200000.0
OK
```

MaxPower <channel\all>

This command returns the maximum output power per channel for the specified card. The maximum power is returned in watts. This command allows a user interface to query the cards for the proper maximum power. The maximum power is a value stored in non-volatile memory in the amplifier card.

Command	Result
MaxPower 2	Requests the maximum power for channel 2

```
MaxPower 2
60.00
OK
```

Measure Sum <forward|reflected|transmitted>

Measure <low channel> <high channel> <forward|reflected|transmitted>

Measure <channel> <forward|reflected|transmitted>

These commands measure the forward or reflected power output generated by the specified channel, or range of channels, in watts.

Note: *"Forward", "Reflected", and "Transmitted" may be abbreviated. The transmitted power measurement measures both the forward and reflected power and returns the difference.*

Command	Result
Measure Sum For	Returns the sum of all the forward powers
Measure 5 7 For	Returns the forward power for channels 5 through 7
Measure 3 Ref	Returns the reflected power on channel 3 in watts

```
Measure Sum f
34.7
OK
Measure 5 7 f
0005 6.3
0006 2.8
0007 9.3
OK
Measure 3 r
0.3
```

MinFreq <channel|all>

This command returns the minimum frequency for the channel in Hertz. It allows a user interface to query the cards for the proper minimum frequency. The minimum frequency is a value stored in non-volatile memory in the amplifier card.

Command	Result
`MinFreq 2`	Requests the minimum clock frequency of the specified channel

```
MinFreq 2
1600000
OK
```

ModEnable <Chan|all> <On|Off>

This command controls Modulation. Sending this command starts modulation and updates the system settings for ModHigh, ModLow, and ModRate. When ModEnable is turned off, channel resumes operation with frequency set prior to modulation.

Note: ModHigh, ModLow, and ModRate settings do not take affect until the ModEnable command is sent.

Command	Result
ModEnable 4 On	Starts Modulation for channel 4.
ModEnable All Off	Turns Modulation off for all channels.

ModHigh <channel|all> <frequency>

This command sets the upper frequency for modulation. This value must be within the range of the amplifier and specified in Hz. If no frequency is entered, returns the uppper modulation frequency for the channel in Hertz.

Command	Result
ModHigh 2 1600000	Sets the maximum modulation frequency for the specified channel
ModHigh 2	Requests the high end modulation frequency for the specified channel

```
ModHigh 2 1600000
OK
ModHigh 2
1600000
```

ModLow <channel|all> <frequency>

This command sets the lower frequency for modulation. This value must be within the range of the amplifier and specified in Hz. If no frequency is entered, returns the lower modulation frequency for the channel in Hertz.

Command	Result
ModLow 2 1300000	Sets the minimum frequency for modulation.
ModLow 2	Requests the lowermodulation frequency for the specified channel

```
ModLow 2
1300000
OK
```

ModRate <channel|all> <frequency>

This command sets the modulation rate. One cycle consists of moving from the low frequency to the high frequency and back again. The minimum rate is 0.1 Hz and the maximum is 9 Hz. Settings above 1 Hz will have a very small frequency resolution. The frequency is entered in Hertz.

Command	Result
ModRate 2 0.9	Sets the modulation rate to 0.9 Hz for channel 2
ModRate 2	Requests the modulation rate for the channel 2

```
ModRate 2
0.9
OK
```

Output <channel|all> <on|off>

This command enables or disables a specific channel.

Note: *To allow power output, the channel's power must be set, it must be enabled with this command, and the entire RF output must be enabled with the "Enable" command. A command entered with only the channel number will return the output state of the specified channel.*

Command	Result
Output 3 on	Enables output of channel 3
Output 3	Displays the state of channel 3
Output all on	Turns all channels on

```
Output 3 on
OK

Output 3
On
OK

Output all on
OK
```

Powerfeedback <card> <on|off>

This command enables or disables the power feedback feature. If the desired state is not specified (on or off), the current setting is returned.

Command	Result
Powerfeedback 2 on	Turns on PowerFeedback for card 2
Powerfeedback 2	Displays PowerFeedback for card 2

```
Powerfeedback 2 on
OK

Powerfeedback 2
ON
OK
```

Power <channel|all> <power>

This command programs the forward RF power for the specified channel in watts. If a power value is omitted, then the command returns the current power value.

Command	Result
Power 8 20	Programs channel 8 to 20W
Power 8	Returns current power value
Output all 5	Sets all channels to 5 Watts
Power all	Returns sum of programmed power

```
Power 8 20
OK

Power 8
20
OK

Power all 5
OK
```

Restart

The Restart command causes the UDS system software to reset and search for cards. Restart is generally the first command that is sent to the UDS system, as it returns the number of cards found.

Search

Searches for and initializes cards.

Set <system variable> <new value>

Set allows the system variables to be changed. A list of all system variables and their current values can be listed by simply sending the command set. These variables are factory settings and should not be modified, with the exception of Echo and Baud.

Note: *The system variables are stored in non-volatile memory, so turning the system off will not reset a poorly chosen value. The system variables should only be changed as a last resort and with full knowledge of the consequences.*

Note: *The command "Set Default" will return all system variables to their default factory state.*

```
Set
   Debug            0
   ACKDelay         5
   Echo             1
   Polling          1
   Retries          3
   Skips            1000
   Stack_int        18
   Calibrate        0
   Baud             115200
   SearchSlots      16
   Ssdelay          300
   PwrWatch         273
OK
```

SetCoeff <channel> <coefficient number> <coefficient>

Each channel is calibrated with its own coefficients that are used in the conversion from power to internal ADC values and from the internal DAC to output power. There are six coefficients. They are described below:

Forward Power = Ax^2+Bx+C where x is the DAC/ADC voltage from 0 to 5V, and A, B, and C are coefficients 1, 2, and 3, respectively.

Reflected Power = Ax^2+Bx+C where x is the DAC/ADC voltage from 0 to 5V, and A, B, and C are coefficients 4, 5, and 6, respectively.

Note: *If the coefficient is omitted, then the current coefficient is returned.*

This command is used primarily for debugging purposes. The coefficients are stored in non-volatile memory on each amplifier card. These values are calculated when the amplifier cards are calibrated at the factory. This command does not alter the calibration data stored in non-volatile memory.

Changing the value of the power coefficients can result in an unexpected high output power, if the values are incorrect.

TempC <channel|all>

This command returns the temperature for the specified channel in degrees C.

TempF <channel|all>

This command returns the temperature for the specified channel in degrees F.

Shutdown

This command places the amplifier in a shutdown mode, turns outputs off, sets the power to zero, disables the amplifier, and stops the clock on the microcontroller on the card. You must issue the restart command or toggle the system power to exit the shutdown mode (safe mode).

```
Shutdown
OK
```

SystemState

This command returns the state of the system, either Shutdown or Normal.

Command	Result
SystemState	Returns the current state of the system

```
SystemState
Normal
OK
```

WhoIs <card\all><card\all>

This command is used to determine which card is in which slot, based on the internal card ID that is stored on the card. This command is only used for debugging or to track the performance of a specific card.

```
WhoIs 3
273202
OK
```

Appendix

A

Serial Communications DLL

A serial communications DLL is provided for programmers who want to write their own amplifier control software in Microsoft Windows. This DLL contains the necessary function calls for serial communications with the amplifier. This header file lists all function calls in the DLL with a brief description of each function.

Note: *The INTERFACE.H file must be included in all programs that use the serial communications DLL.*

```
/* UDS Interface.h                                                          */
/*             Header file for UDS Interface Communication DLL              */
/*              (c) 1998-2002 Advanced Surgical Systems, Inc.               */
/*                          All Rights Reserved.                            */
/*             ---------------------------------------------                */
/*               To be included with all C/C++ programs that                */
/*                communicate with the UDS Amplifier System.                */
/*             ---------------------------------------------                */
/*                       To ensure compatibility                            */
/*                      !DO NOT MODIFY THIS FILE!                           */
/* ------------------------------------------------------------------------ */
/* DLLVERSION is defined as "0x00.0x00.0x00" + the 2-digit language code    */
/* Language Code: 0xBC -- Visual Basic 5.0 and C/C++                        */
#define DLLVERSION 0x010100BC
/*
/*Function descriptions and usage:                                          */
/* ------------------------------------------------------------------------ */
/* void GetDLLVersion(LPDWORD DLLVersion)                                   */
/*   Retrieves the current DLL version numbers. The format is:              */
/*   0x00.0x00.0x00 + 2 Hex Digit Language Code (Should be 0xBC)            */
/*   This should be checked in every program to insure compatibility.       */
/* ------------------------------------------------------------------------ */
/* BOOL GetDLLSpecifics(LPDWORD Index, LPBSTR DetailString)                 */
```

```
/*   Detailed information about DLL compilation date, time, etc.              */
/*     Index: 1. FILE, 2. TIMESTAMP, 3. DATE, 4. TIME                         */
/*     DetailString: On return contains requested information as a string.    */
/*   Return Value:  TRUE if Index is valid.                                   */
/* ------------------------------------------------------------------------   */
/* BOOL SetTimeouts(LPDWORD CommandTimeout,LPDWORD ResetTimeout)              */
/*   Sets the communications timeouts, all in milliseconds.                   */
/*     CommandTimeout: Time to wait between characters for 'OK' response.     */
/*     ResetTimeout: Time to wait for the amplifier system to reset.          */
/*   Send NULL arguments to retrieve current timeout settings.                */
/*   Return Value:  Always returns TRUE in DLL Version 1.0.2.                 */
/* ------------------------------------------------------------------------   */
/* BOOL SetErrCallback(BOOL (*ErrorFunction)(LPDWORD ErrorCode,               */
/*     LPDWORD ErrorSubCode,LPDWORD Arguments,LPBSTR ErrorString))            */
/*   NOTE: Should use "SetErrorCallback" function name.                       */
/*   Sets the error handling routine callback function.                       */
/*     ErrorCode: Specific error code                                         */
/*     ErrorSubCode: Major error code to categorize errors                    */
/*     ErrorString: Original error string from amplifier                      */
/*   Return Value:  TRUE if callback function is valid.                       */
/* ------------------------------------------------------------------------   */
/* BOOL OpenCommPort(int PortNumber,int BaudRate);                            */
/*   Opens the COM port and sets necessary parameters.                        */
/*     PortNumber: COM Port number (1,2,3,...)                                */
/*     BaudRate: Desired Baud Rate (19200 - 115200)                           */
/*   Return Value:  TRUE if successful.                                       */
/* ------------------------------------------------------------------------   */
/* BOOL ResetAmplifierSystem(int *CardsFound)                                 */
/*   Resets the amplifier system.                                             */
/*     CardsFound: Contains the number of cards found on successful reset.    */
/*   Return Value:  TRUE if reset successful.                                 */
/* ------------------------------------------------------------------------   */
/* BOOL SendCmd(LPSTR CommandString,LPBSTR ResponseString)                    */
/*   NOTE: Should use "SendCommand" function name.                            */
/*   Sends a command to the amplifier system.                                 */
/*     CommandString: String containing command and arguments.                */
/*     ResponseString: String containing response from amplifier.             */
/*   Return Value:  TRUE if command sent and processed successfully.          */
/* ------------------------------------------------------------------------   */
/* BOOL CloseCommPort(void)                                                   */
/*   Closes the COM port when terminating communication with the system.      */
/*   Return Value:  TRUE if port closed successfully.                         */
/* ------------------------------------------------------------------------   */
/* BOOL InitializePolling(void)                                               */
/*   Initializes the use of the Callback function to process error codes.     */
/*   Return Value:  TRUE if Callback exists and polling thread                */
/*     successfully initialized.                                              */
```

```
/* ------------------------------------------------------------------------    */
/* BOOL EndPolling(void)                                                       */
/*   Terminates the use of the Callback function to process error codes.       */
/*   Return Value:  TRUE if polling successfully terminated.                   */
/* ------------------------------------------------------------------------    */
/* BOOL SetFailureLimit(int *FailureLimit)                                     */
/*   Sets the maximum number of successive communications failures allowed     */
/*     before further communications attempts are ignored.                     */
/*     FailureLimit: Number of maximum allowed successive failures.            */
/*   Return Value:  Always returns TRUE in DLL Version 1.0.0.                  */
/* ------------------------------------------------------------------------    */
/* BOOL ResetFailureCount(void)                                                */
/*   Resets the number of successive communications failures and enables       */
/*     further communications with the amplifier system.                       */
/*   Return Value:  TRUE if communications were disabled because of            */
/*     failure limit reached, otherwise FALSE.  Always resets the count.       */
/* ------------------------------------------------------------------------    */
/* BOOL SetCardsPerRack(LPDWORD CardsPerRack)                                  */
/*   NOTE:  This function should only be used with UDS 32xx Systems!           */
/*   Sets the number of cards per rack.                                        */
/*     CardsPerRack:  Number of cards per rack.  Only 8 and 16 supported.      */
/*   Return Value:  TRUE if CardsPerRack is 8 or 16.                           */
/* ------------------------------------------------------------------------    */
/* BOOL GetErrorDesc(LPDWORD ErrorCode,LPDWORD ErrorSubCode,                   */
/*     LPBSTR Description)                                                     */
/*   NOTE: Should use "GetErrorDescription" function name.                     */
/*   Gets a string description of an error code.                               */
/*     ErrorCode: Specific error code.                                         */
/*     ErrorSubCode: Refers to level at which error occured.                   */
/*     Description: String containing description of error code.               */
/*   Return Value:  TRUE if valid error code.                                  */
/* ------------------------------------------------------------------------    */
/* Error Code Definitions:                                                     */
/* NOTE:  These definitions should be used in code rather than the actual      */
/* error numbers, as the numbers are subject to change.                        */
/* ------------------------------------------------------------------------    */
#define LEVEL_INT            0    /* Errors generated by DLL*/
/* ------------------------------------------------------------------------    */
#define ERROR_COMTIMEOUT     0   /* "Communications Timeout"                   */
#define ERROR_SYSTEMRESET    1   /* "System Reset"                             */
#define ERROR_BADRESET       2   /* "Bad Reset"                                */
#define ERROR_INTERNALERROR  3   /* "Internal Error"                           */
#define ERROR_TOOMANYFAILURES 4  /* "Too Many Failures"                        */
#define ERROR_UNKNOWN        5   /* "Unknown Error"                            */
/* ------------------------------------------------------------------------    */
#define LEVEL_CARD           1    /* Errors referring to a card                */
/* ------------------------------------------------------------------------    */
```

```
#define ERROR_COM_RESUMED     0    /* "Communications Resumed"               */
#define ERROR_NORESPONSE      4    /* "No Response From Card"                */
#define ERROR_SYNCLOST        5    /* "Synchronization Lost"                 */
#define ERROR_RESET           6    /* "Card Reset"                           */
#define ERROR_DEACTIVATED     8    /* "Card Deactivated"                     */
/* ------------------------------------------------------------------------    */
#define LEVEL_CHANNEL         2    /* Errors referring to a channel          */
/* ------------------------------------------------------------------------    */
#define ERROR_NOTPRESENT      3    /* "Channel Not Present"                  */
#define ERROR_NOSUCCESS       7    /* "Command Not Successful"               */
#define ERROR_LOSTLOCK        9    /* "Channel Lost Lock"                    */
#define ERROR_LIMIT_FOR_HIGH  10   /* "Forward High Limit Exceeded"          */
#define ERROR_LIMIT_FOR_LOW   11   /* "Forward Low Limit Exceeded"           */
#define ERROR_LIMIT_REF_HIGH  12   /* "Reflected High Limit Exceeded"        */
#define ERROR_LIMIT_REF_LOW   13   /* "Reflected Low Limit Exceeded"         */
/* ------------------------------------------------------------------------    */
#define LEVEL_SBC             3    /* Errors generated by embedded control   */
/* ------------------------------------------------------------------------    */
#define ERROR_BADCMD          1    /* "Bad Command"                          */
#define ERROR_BADARG          2    /* "Bad Argument”                         */
#define ERROR_EMERG_1         20   /* "Emergency Shutdown 1 Activated"       */
#define ERROR_EMERG_2         21   /* "Emergency Shutdown 2 Activated"       */
#define ERROR_EMERG_3         22   /* "Emergency Shutdown 3 Activated"       */
#define ERROR_EMERG_4         23   /* "Emergency Shutdown 4 Activated"       */
/* ------------------------------------------------------------------------*/
#define ERROR_POLLING         1000 /* Additive factor used for polling       */
/* ------------------------------------------------------------------------*/

/* Definitions for Special Response Strings:                                 */
/* NOTE:  These definitions should be used in code rather than the actual    */
/* strings, as the string content is subject to change.                      */
/* ------------------------------------------------------------------------*/
#define ERRORSTRING         "ERROR"            /* Indicates System Error    */
#define RESETSTRING           "RESET"          /* Indicates System Reset    */
#define TIMEOUTSTRING         "TIMEOUT"        /* Indicates Timeout Error   */
#define BADRESETSTRING        "BADRESET"       /* Indicates Bad Reset       */
#define INTERNALERRORSTRING   "INTERNAL ERROR" /* Indicates Error in DLL    */
#define TOOMANYFAILURESSTRING "TOOMANYFAILURES" /* Indicates Too Many Fails  */
#define LEVELSTRING           "LEVEL"          /* Used for Stack Prog.      */
#define OKSTRING              "OK"             /* Response to every cmd.    */
#define READYSTRING         "SYSTEM READY"     /* Sent After Reset.         */
/* ------------------------------------------------------------------------    */

#define DLLExport extern "C"
#ifdef __cplusplus
        extern "C" {
#endif
```

```
void __stdcall GetDLLVersion(LPDWORD DLLVersion);
BOOL __stdcall GetDLLSpecifics(LPDWORD Index, LPBSTR DetailString);
BOOL __stdcall SetTimeouts(LPDWORD CommandTimeout,LPDWORD ResetTimeout);
typedef BOOL (__stdcall *STDFUNCPTR)(LPDWORD ErrorCode,LPDWORD ErrorSubCode,
                                                                    LPDWORD
ErrorArg,LPBSTR ErrorString);
BOOL __stdcall SetErrCallback(STDFUNCPTR ErrorFunction);
BOOL __stdcall OpenCommPort(int PortNumber,int BaudRate);
BOOL __stdcall ResetAmplifierSystem(int *CardsFound);
BOOL __stdcall SendCmd(LPSTR CommandString,LPBSTR ResponseString);
BOOL __stdcall CloseCommPort(void);
BOOL __stdcall InitializePolling(void);
BOOL __stdcall EndPolling(void);
BOOL __stdcall SetFailureLimit(int *FailureLimit);
BOOL __stdcall SetCardsPerRack(LPDWORD SuggCardsPerRack);
BOOL __stdcall ResetFailureCount(void);
BOOL __stdcall GetErrorDesc(LPDWORD ErrorCode,LPDWORD ErrorSubCode,LPBSTR Description);

#ifdef __cplusplus
        }
#endif
```

Appendix

B

Error Codes

The following error codes can appear during the operation of the system:

Error Code	Error Name	Error Description
0	ERROR_COM_RESUMED	Communication was resumed after having been lost.
1	ERROR_BADCMD	The command was not recognized.
2	ERROR_BADARG	The argument was inappropriate.
3	ERROR_NOTPRESENT	Attempt was made to access a channel that does not exist.
4	ERROR_NORESPONSE	The amplifier card did not respond.
5	ERROR_SYNCLOST	Communications error with the amplifier card.
6	ERROR_RESET	The amplifier card reset.
7	ERROR_NOSUCCESS	No success talking to the amplifier card. This can also mean that the modulation data did not verify correctly when trying to set up frequency modulation.
8	ERROR_DEACTIVATED	The card was deactivated, probably because too many errors occurred during communication.
1010	ERROR_OVER_FOR_HIGH	The forward high power limit was exceeded.
1011	ERROR_OVER_FOR_LOW	The forward low power limit was exceeded.
1012	ERROR_OVER_REF_HIGH	The reflected high power limit was exceeded.
1013	ERROR_OVER_REF_LOW	The reflected low power limit was exceeded.
1019	POWER_ENABLE_TIMEOUT	No command received before power enable timer collapsed.

Appendix

C

RS-232 Cable Connections

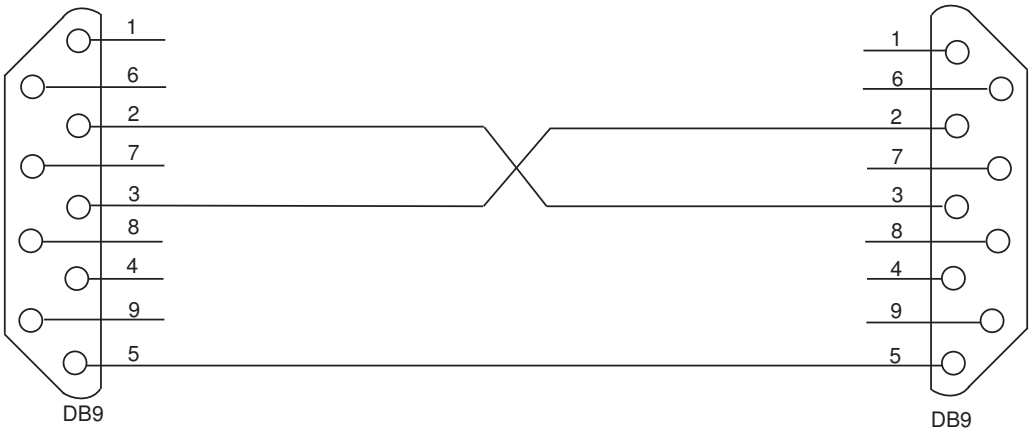


Figure 7 – Null Modem Cable. RS-232 wire connections between the computer and generator for 9-pin D-Sub connectors.

RF Connector Pinout

ITT/Cannon P/N DL1-156P

SH = Coax Shield

P = Center Conductor

	1	2	3	4	5	6
A	P1	SH2	P3	SH4	P5	SH6
B	SH1	P2	SH3	P4	SH5	P6
C	P7	SH8	P9	SH10	P11	SH12
D	SH7	P8	SH9	P10	SH11	P12
E	P13	SH14	P15	SH16	P17	SH18
F	SH13	P14	SH15	P16	SH17	P18
G	P19	SH20	P21	SH22	P23	SH24
H	SH19	P20	SH21	P22	SH23	P24
J	P25	SH26	P27	SH28	P29	SH30
K	SH25	P26	SH27	P28	SH29	P30
L	P31	SH32				
M	SH31	P32				
N						
P						
R					P33	SH34
S					SH33	P34
T	P35	SH36	P37	SH38	P39	SH40
U	SH35	P36	SH37	P38	SH39	P40
V	P41	SH42	P43	SH44	P45	SH46
W	SH41	P42	SH43	P44	SH45	P46
X	P47	SH48	P49	SH50	P51	SH52
Y	SH47	P48	SH49	P50	SH51	P52
Z	P53	SH54	P55	SH56	P57	SH58
a	SH53	P54	SH55	P56	SH57	P58
b	P59	SH60	P61	SH62	P63	SH64
c	SH59	P60	SH61	P62	SH63	P64

Appendix

D

Warranty Information

Advanced Surgical Systems, Inc. warranties the *RF Generator 500-020* to be free from defects in materials and workmanship for a period of one (1) year from the date of shipment. If any such product proves defective during this warranty period, *Advanced Surgical Systems, Inc.*, at its option, either will repair the defective product without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, the customer must notify *Advanced Surgical Systems* of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. The customer shall be responsible for packaging and shipping the defective product to the service center designated by *Advanced Surgical Systems*, with shipping charges prepaid. *Advanced Surgical Systems* shall pay for the return of the product to customer if the shipment is to a location within the United States. The customer shall be responsible for paying all shipping charges, duties, taxes, and any other charges for products returned to any other locations.

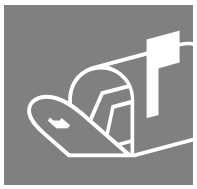
This warranty shall not apply to any defect, failure, or damage caused by improper use of the device or improper or inadequate maintenance and care. *Advanced Surgical Systems* shall not be obligated to furnish service under this warranty to (a) repair damage resulting from attempts by non-*Advanced Surgical Systems* personnel to install, repair, or service the product; (b) to repair damage resulting from improper use or connection to incompatible equipment; or (c) to service a product that has been modified or integrated with other products when the effect of such modification or integration increases the time or difficulty of servicing the product.

Appendix

E

Technical Support

Technical support is available at no charge throughout the life of the product. Additionally, *Advanced Surgical Systems* is willing to work with its customers to make sure that the equipment is used to its maximum potential, whether this be regarding interfacing with the transducers, developing control software, or any other issues that may arise throughout the lifetime of the UDS system.



Postal Address

Advanced Surgical Systems, Inc.

5030 East Calle Barril

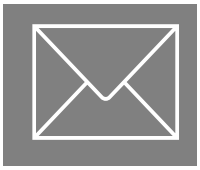
Tucson, Arizona 85718

Telephone

(520) 529-1796

Fax

(520) 529-5541

j 

E-mail Address

support@advanced-surgical.com

Specifications