OPERATOR MANUAL 990-552
Revision F August 2010

OPERATOR MANUAL

FOR THE

LMF SERIES
FIBER LASER MARKERS

Model	Nominal Power	Laser Specification
LMF3500-HP	35W	High Performance
LMF2000-HP	20W	High Performance $M^2 < 2.0$
LMF2000-SM	20W	Single Mode $M^2 < 1.3$
LMF2000	20W	Standard
LMF1000	10W	Standard

Revision Record

Revision	EO	Date	Basis of Revision
A	31255	4/09	Production Release
B	31335	5/09	Added Enhanced head and updated operating instructions.
C	31393	6/09	Updated technical information.
D	40373	5/10	Updated connection schematics.
E	40498	6/10	Not Released for production.
F	40607	8/10	Addition of New Models LMF3500-HP, LMF2000-SM. Manual is now generic for LMF Series Markers. Updated marking field information.

Your LMF Fiber Laser Marker Shipment Contains The Following Items:

1. LMF Laser Marker and Power Supply
2. *Quickstart* Guide For The *Winlase* LAN Laser Marker Software 990-550
3. Operator Manual For The LMF Fiber Laser Marker 990-552
4. Laser Safety Manual 990-502
5. Ship Kit # 4-81192-01
   - Cord, CE, #18-3, M/F, Black (1 Each)
   - Cable, Db15 Ext, W/ Ferrites (1 Each)
   - Cable, Crossover, Cat5 (1 Each)
   - Angle Bracket,19" Rack Mount (2 Each)
   - Screw Hex, Socket, Button Head, M5x14 (6 Each)
   - Flat Round Washer, 5mm (6 Each)
   - Spring Lock Washer, Split (6 Each)
   - Cable Db25 Ext Shield Twisted Pair (1 Each)
6. Customer specified beam expander and *f*-theta lens installed in the marker

CONTENTS

Page

CONTENTS (Continued)

Page

CONTENTS (Continued)

Page

CONTACT US

Thank you for purchasing a Miyachi Unitek™ LMF Series Fiber Laser Marker.

Upon receipt of your equipment, please thoroughly inspect it for shipping damage prior to its installation. Should there be any damage, please immediately contact the shipping company to file a claim, and notify Miyachi Unitek at:

1820 South Myrtle Avenue
P.O. Box 5033
Monrovia, CA 91017-7133

Telephone: **(626) 303-5676**

FAX: **(626) 358-8048**

E-Mail: **Info@muc.Miyachi.com**

The purpose of this manual is to provide the information required for proper and safe operation and maintenance of the Miyachi Unitek™ LMF Fiber Laser Marker.

We have made every effort to ensure that information in this manual is both accurate and adequate. If you have any questions or suggestions to improve this manual, please contact us at the phone number or address above.

Miyachi Unitek Corporation is not responsible for any loss or injury due to improper use of this product.

SAFETY PRECAUTIONS

General

This Operator's Manual describes the Operation and Maintenance of the LMF Series Fiber Laser Marker, and provides instructions relating to its SAFE use. Procedures described in this manual ***must*** be performed as detailed by ***qualified*** and ***trained*** personnel.

NOTE: For the rest of this manual, all models of LMF Series Fiber Laser Markers will simply be referred to as "***the Marker,***" except in specific instances where unique descriptions are required such as specifications, connections, etc.

For SAFETY, and to effectively take advantage of the full capabilities of the Marker, please read this instruction manual and the Laser Safety Manual (Part Number 990-502) thoroughly ***before*** attempting to use the Marker.

After reading this manual, retain it for future reference when any questions arise regarding the proper and SAFE operation of the Marker.

Operation

When operating or servicing the Marker, ***always*** wear Protective Goggles having an optical density of at least 7^{+} at a wavelength of 1060-1150 nanometers for the operation of the Marker.

Appoint a Laser Safety Officer. The Laser Safety Officer (LSO) must provide personnel with sufficient training so that personnel can operate, maintain and service the Laser Marker safely. The LSO must take charge of the key to the Key Switch to ensure that ***only*** qualified and authorized personnel operate the Laser Marker.

Establish and control a dedicated Laser Operation Area. The Laser Safety Officer must isolate the Laser Operation Area from other work areas and display signs warning that the Laser Operation Area is off-limits to unauthorized personnel.

Maintenance/Service

Before performing any maintenance on the Marker, read *Chapter 4, Maintenance* thoroughly. Use the appropriate tools for terminating the connecting cables, being careful not to nick the wire conductors.

Procedures other than those described in this manual or not performed as prescribed in this manual, may expose personnel to electrical and/or laser radiation hazards.

Do ***not*** modify the Marker without prior written approval from the Miyachi Unitek Corporation.

Before using this equipment, read the **Safety Precautions** carefully to understand the correct usage of the equipment.

- These precautions are given for the safe use of the Marker and for prevention of injury to operators or others.
- Be sure to read ***each*** of the instructions, as they are all important for safe operation.
- The meaning of the words and symbols are as follows:

⚠ CAUTION

Denotes operations and practices that may result in personal injury or damage to the equipment if not correctly followed.

⚠ WARNING

Denotes operations and practices that may result in serious injury or loss of life if not correctly followed.

⚠ DANGER

Denotes operations and practices that may imminently result in serious injury or loss of life if not correctly followed.

These symbols denote **PROHIBITION**. They are warnings about actions that should ***not*** be performed because they can damage the equipment and will void the warranty.

These symbols denote actions which operators ***must*** take.

Each symbol with a triangle denotes that the contents gives notice of **DANGER**, **WARNING**, or **CAUTION** to the operator.

⚠ DANGER

Do *not* touch inside the Marker when it is turned ON.

Doing so may result in electric shock.

***Never* attempt to disassemble, repair, or modify the Marker.**

Doing so may result in electric shock or fire.
Refrain from any mechanical adjustment other than the maintenance procedures specifically described in the operation manual.

***Never* expose eyes or skin to laser irradiation.**

Exposure to direct or scattered laser light is extremely hazardous.
Direct exposure of the eye to laser beams may result in blindness.

WARNING

Wear protective eyewear.

Always wear protective eyewear when using the Marker.
Keep in mind that exposure of the eyes to direct laser irradiation may result in blindness, even when wearing protective eyewear.

Never aim the laser at any part of your own body or other people.

Exposure to laser beams will cause severe burns.
Never aim the laser at yourself or at anyone else.

Do *not* touch workpieces during or just after marking.

Workpieces may still be very hot.

Use only the specified cables. Make sure they are firmly connected.

Using cables of inadequate current capacity or connecting cables loosely may result in fire or electric shock.

Avoid damaging power or connecting cables.

Do not step on, twist, or pull cables.
Damaged cables may result in electric shock, short circuits, or fires.
To repair or replace cables, contact your Miyachi dealer or the Miyachi Unitek Corporation.

Avoid damaging the delivery fiber.

Do not twist or kink the fiber.

Do not attempt to coil or bend the fiber tighter than a 6 inch radius.

Do not attempt to remove the fiber

Doing any of these actions will required factory refurbishment of the laser and will void the warranty.

Stop using the Marker if any problems arise.

Continuing to use the Marker in the presence of abnormalities (fumes, unusual sounds, excessive heat, smoke, and so forth) may result in electric shock or fire.
In this case, immediately turn the Marker OFF and contact your Miyachi dealer or the Miyachi Unitek Corporation.

Ground the Marker.

Failure to ground the Marker may result in electric shock if the Marker is damaged or if electrical leaks occur.

People using pacemakers must *NOT* approach the Marker.

Unless a physician has consented, pacemaker users must not approach the Marker in use, or even approach the working area.
The Laser Marker generates electromagnetic fields that may affect pacemaker function.

CAUTION

	**Avoid spilling or splashing water on the Marker.** The presence of water on electrical parts may result in electric shock or short circuits. Liquid spills may degrade the unit's insulation, resulting in electric leaks or fire.
	**Use the appropriate tools to terminate the power cable (wire strippers, crimp tools, etc.).** Failure to use the appropriate tools may result in damage to the wire core, resulting in fire or electric shock.
	**Install the Marker on a solid, level surface.** Should the Marker tip over or fall, injury or damage to the unit may result.
	**Keep combustible materials away from the Marker.** Sparks or spattering material may ignite combustible matter. To avoid the risk of fire, never apply the laser beam to flammable or combustible materials.
	**During use, do *not* cover the Marker with a blanket, cloth, or similar articles.** When using the Marker, do ***not*** cover with a blanket, cloth, or similar articles. The Laser Marker may become extremely hot, resulting in fire.
	**Do *not* use the Marker for any purpose other than laser marking.** Using the unit for nonspecified applications may result in electric shock or fire.
	**Wear protective gear.** Use protective gloves, long-sleeve garments, leather aprons, or other appropriate protective gear. Sparks or spattering material may burn the skin on contact.
	**Keep a fire extinguisher nearby.** Keep a fire extinguisher in the marking area in case of fire.
	**Maintain and inspect the unit at periodic intervals.** Maintain and inspect the unit at periodic intervals. Repair any damage before resuming use.

Guidelines for Normal Use

1. Appoint a Laser Safety supervisor. Ensure that the supervisor has as much expertise and experience with lasers and laser equipment as possible.

   The supervisor, who will be in charge of the laser key switch, is responsible for familiarizing users with safety issues and for coordinating laser marking.

2. Partition off all areas that may be exposed to laser light.

   The supervisor is responsible for posting signs to keep unauthorized personnel out of the marking area.

3. Install the Marker on a solid, level surface.
   - To prevent errant marking, place workpieces on the same stand as the Marker Head so that the workpieces do ***not*** vibrate during marking.

4. To ensure optimal marking quality, use the Marker in a location where ambient temperatures are 41ºF to 95ºF (5°C to 35°C), free of sudden temperature fluctuations and a relative humidity less than 90% (non-condensing). Do ***not*** use the marker in any of the following locations:
   - Locations with excessive dirt or oil mist
   - Locations in which the unit may be subject to vibration or impact
   - Locations in which the unit may be exposed to chemicals
   - Locations near sources of high-frequency noise, or
   - Locations in which condensation may form on the unit's surface.

5. If the room temperature changes quickly (as when a heater is turned ON in cold weather), moisture may condense on the optical components, resulting in fogging or collection of dust.

   Avoid sudden changes in temperature. Under the conditions in which condensation may occur, wait for a period of time after turning the unit ON before beginning operations.

6. If the exterior of the unit becomes soiled, wipe it with a soft dry or lightly moistened cloth.

   Clean heavily soiled areas with a cloth moistened with diluted neutral detergent or alcohol. Do ***not*** use paint thinner, acetone, benzene, or similar chemicals, which may discolor or damage the unit.

7. Never place screws or other foreign objects inside the marker. Such objects can damage the unit.

8. Operate the switches and buttons gently by hand.

   Applying excessive force or using the tip of a screwdriver, pen, or other instrument may damage the Marker.

9. Operate only one switch or button at a time.

   Attempting to operate several at a time may damage the Marker.

WARNING

Always wear protective goggles when maintaining the Marker. Goggles must have an optical density of at least 7+, at a wavelength of 1060-1150 nanometers for the operation of the Marker.

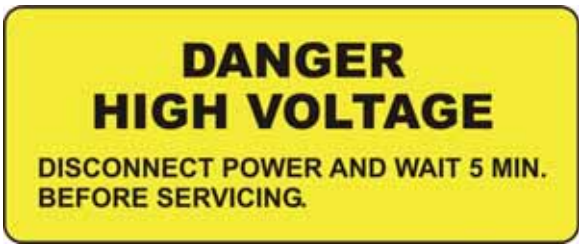
10. For more consistent marking, allow the unit to thermally stabilize for approximately 10 to 30 minutes before use. (The appropriate warm-up time will depend on the ambient temperature and work piece material.)

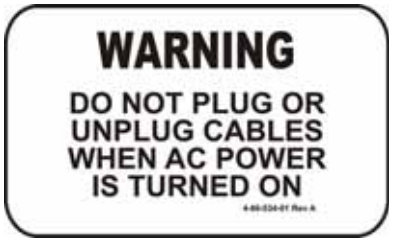
    Refer to the following standards for more information on managing laser equipment:

    - **IEC60825-1 Edition1.2** "Safety of laser products Part1: Equipment Classifications, requirements and user's guide."
    - **Miyachi Unitek Laser Safety Manual** (Part Number 990-502)

Warning Labels

The Laser Marker carries the following labels. Read and follow the label instructions to ensure correct use.






DANGER
AVOID EYE OR SKIN
EXPOSURE TO DIRECT OR
SCATTERED RADIATION.


DANGER
VISIBLE AND/OR INVISIBLE
LASER RADIATION WHEN OPEN
AND INTERLOCK DEFEATED.


AVOID EXPOSURE
VISIBLE AND/OR INVISIBLE
LASER RADIATION IS EMITTED
FROM THIS APERTURE.
4-38089-01 REV B


CAUTION
DO NOT
ATTEMPT
TO REMOVE
THE FIBER
UNDER ANY
CIRCUMSTANCES.
DOING SO WILL
DESTROY THE
FIBER AND VOID
THE WARRANTY.

DNB

5969 Robinson Ave
Riverside, CA 92503
(951) 637-2630
FAX (951) 637-2704
www.dnbengine.com

22 Jan 2009
RV98009A-002

Declaration of Conformity

Application of Council Directive: 2004/108/EC

Standards To Which Conformity Is Declared:

EN61326: 2006

EN55011 Class A Group 1
EN61000-4-2
EN61000-4-3
EN61000-4-4
EN61000-4-5
EN61000-4-6
EN61000-4-8
EN61000-4-11

Manufacturer's Name: **Miyachi Unitek**

Manufacturer's Address: **1820 S. Myrtle Avenue**
Monrovia, CA 91016
626-303-5676

Equipment Description: **Laser Marker**

Equipment Class: **Electrical Equipment Measurement, Control & Laboratory Use - Industrial**

Model Numbers: **LMF2000/-HP/-1000/ML-7311/ML-7320/ML-7321**

I the undersigned, hereby declare that the equipment specified above, conforms to the above Directive(s) and Standard(s).

MONROVIA CA USA
Place:

David Cielinski
Signature:

DAVID CIELINSKI
Full Name:

VP STANDARD PRODUCT DEVELOPMENT
Position:

CHAPTER 1
SYSTEM DESCRIPTION

Section I: Features

The LMF Series Fiber Laser Markers are high-precision scanning fiber laser markers, either pulsed or Q-switched depending on the configuration as delivered. For the rest of this manual, all models of the LMF Series Fiber Laser Markers will simply be referred to as "***the Marker,***" except in specific instances where unique descriptions are required such as specifications, connections, etc.

This Marker is configured to operate with *WinLase* laser control software. For the rest of the manual this software will simply be referred to as ***the software***. Please refer to the appropriate software manual for details.

Laser Marking

- **Permanent marking.**

  In contrast to ink-based printing, laser marking is permanent, since the laser beam changes the material itself.
- **Environmentally friendly.**

  No ink is needed, so no solvent is used.

  Use of recycling marked materials is easier because they contain no ink.
- **Non-contact marking.**

  Permits marking of curved and concave surfaces.

Control with Standard Head - ≤ 20W Models Only

Features

- **Compact and lightweight.**

  Ideal for production lines where space is limited.
- **Energy efficient.**

  Low power consumption, thanks to a highly efficient laser diode.
- **Compact IP-54 head available.**
- **Fully air-cooled.**

  Easy maintenance. No coolant or coolant filters needed.

Compact Head – All Models

- **Compatible with PCs running with the *Windows XP* ™ operating system.**

  The *WinLase* application runs in Windows XP and allows the marking data to be laid out easily on a large drawing screen. The software is user friendly and fully featured enabling the most complicated of mark operations to be easily programmed. When using WinLase, one PC can simultaneously administer multiple markers.

- **High-speed marking.**

  Capable of marking at extremely high speeds (≥ 5000mm/s) for the fastest possible marking time. The maximum speed for each process is dependant on the selected optical configuration and the material being marked.

- **Built-in guide beam.**

  A visible 650nm red guide beam for positioning makes it easy to align marking positions.

- **Full-featured drawing functions allow more efficient production of marking data.**

  Functions include: move, rotate, copy, enlarge, reduce, compress, mirror text, reverse marking, undo, redo, grid, ruler.

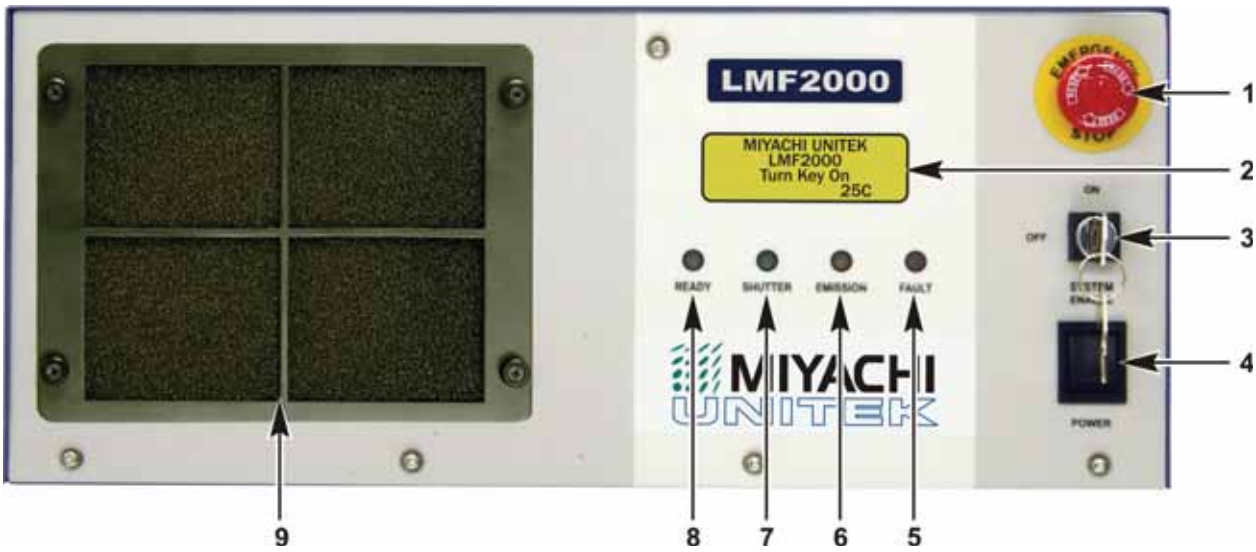
- **Stand-alone capabilities.**

  Depending on the hardware configuration of the Marker, it can be run independently of a PC. When configured with the WinLase marking software, the marker can store marking jobs on the built-in 32MB memory (expandable) and then the jobs can be run via the External I/O, RS-232, TCP/IP, or an external pendant. .

Section II: Part Names and Functions

Control Unit (Front)

The control unit incorporates the controller, electronic cooling unit, interface hardware, and power supply unit. You can monitor the operating status from a computer using the TCP/IP protocol.



1. Emergency Stop Button

Press to stop all marking processes immediately during an emergency. This will immediately suspend marking, close the shutter, provide a fault signal to the operator, cease motion, and shut down the laser diode power supply. Once pressed, the button will remain depressed. To reset this button, turn the knob in the direction indicated by the arrows. To reset the laser, either cycle the key switch or send an I/O signal to "Fault Reset" as described in *Appendix B, Electrical and Data Connections*.

CAUTION

Use this button to stop the device ***only*** in emergency situations. Use the key switch and main power switch for normal use.

2. LCD Display Window.

Describe various messages on the status of the Marker and any system faults or errors that may occur during marking.

3. SYSTEM ENABLE Key Switch

Enables marking. This key switch must be in the ON position to open the safety shutter and enable marking. If the key switch is in the OFF position the marker is incapable of laser emission. The marker can still be operated using the guide beam only with the key switch in the off position.

4. POWER Switch

This switch allows the Operator to turn the Marker ON and OFF. The **INPUT POWER** switch on the rear panel must be turned ON in order for this switch to become active.

5. FAULT Light

Indicates that an error has occurred. The specific fault is indicated by a coded series of flashes. Please note the sequence and refer to this manual for resolution. Identical information is available on the LCD display.

6. EMISSION Light

Indicates that marking is underway. This indicator will be briefly illuminated during initial power up as a self check mechanism, but emission does not occur.

7. SHUTTER Light

Indicates that the Safety Shutter in the laser head is open.

8. READY Light

Indicates that the Marker is ready to mark.

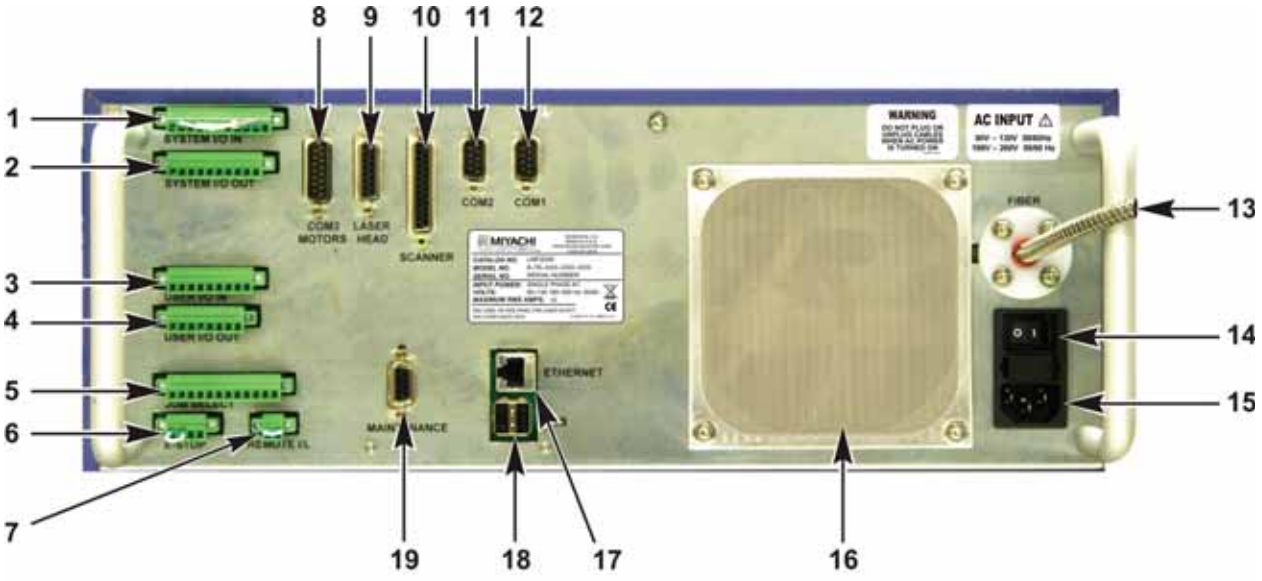
NOTES:

- The **READY** indicator is illuminated when no faults are detected and the laser power supply is ready to mark.
- The Marker may be powered OFF at any time.
- If the **READY** indicator does not illuminate within 90 seconds after power is applied a fault has occurred and the **FAULT** indicator will illuminate.

9. Cooling Fan Air Intake

Clean the air filter at regular intervals. See *Chapter 4, Maintenance* for more details.

Control Unit (Rear)



1. SYSTEM I/O IN Connector

This connector allows for control of marker functions using external control. Features that can be controlled include Mark Start, the Safety Shutter, Laser Enable, Fault Reset, etc. Please see *Appendix B, Electrical and Data Connections* for details.

2. SYSTEM I/O OUT Connector

This connector allows the user to interface with status outputs using external control. Signals that can be monitored include Fault conditions, **Emission**, **System Ready**, **Mark in Progress**, etc. See *Appendix B, Electrical and Data Connections* for details.

3. USER I/O IN Connector

This connector contains programmable inputs to the Marker. The behavior exhibited when these signals are sent is programmed by the operator using the software suite. Depending on the software used, some of these signals will be unavailable.

4. USER I/O OUT Connector

This connector contains programmable outputs controlled by the software suite. Depending on the software used, some of these signals will be unavailable.

5. JOB SELECT Connector

This connector allows the user to address 8-bit I/O job selection. Please refer to *Appendix B, Electrical and Data Connections* for details on how to wire this connector and the Operation chapter for details on software configuration and setup.

6. E-STOP (Emergency Stop) Connector

This connector allows the user to integrate an external **Emergency Stop** button to the marker. The signal should be a dry-contact type and must be closed to operate the marker. Please refer to *Appendix B, Electrical and Data Connections* for details.

7. REMOTE I/L (Interlock) Connector

This dry contact connection is used to control the laser's remote interlock. If the remote interlock is activated the safety shutter is immediately closed and an error message is sent to the software if a marking operation is in progress.

8. COM3 MOTORS Connector

This connector allows the user to connect serial smart motors or an encoder to be used with Mark on the Fly. *Please see Appendix B, Connections* for more details.

9. LASER HEAD Connector

This 15-pin D-Sub cable contains communication signals between the marker and laser head including head interlocks, shutter power and sensor signals, etc. If this cable is disconnected the marker will be in the Emergency Stop state.

10. SCANNER Connector

This 25-pin D-Sub cable contains power and digital communication signals for the laser scan head. Please do not operate the marker without this connected as it may damage the scanhead or cause unaimed laser emission. Do **not** plug or unplug this connector when power is **ON.**

11. COM2 Connector

This connector is primarily used to control a RS232 pendant. It also allows the user to communicate via a simplified 3-wire RS232 implementation using pins 2, 3, and 5 for **Received Data**, **Transmitted Data**, and **Ground** respectively when **Hardware Flow Control** is disabled. Pin 9 is connected to +5V to provide pendant power, so ensure that any other devices that connect to this port can not be damaged.

12. COM1 Connector

This connector is primarily used for RS232 connection to the *Winlase* Remote API. It allows the user to communicate via a simplified 3-wire RS232 implementation using pins 2, 3, and 5 for **Received Data**, **Transmitted Data**, and **Ground** respectively when **Hardware Flow Control** is disabled.

13. FIBER

Optical fiber that connects the Laser output to the Laser head. ***NEVER*** allow the fiber bend radius to decrease to less than 6". ***NEVER*** kink or crush the fiber. Fiber damage will require refurbishment of the laser power supply. The fiber is shipped installed and aligned. The fiber output assembly can ***not*** be removed and replaced except by trained personnel. Some alignment may be required if the laser module is removed and reinstalled in the standard head.

14. INPUT POWER Switch

Turns unit power ON/OFF. Immediately below this switch is a small drawer containing two fuses. Please inspect these fuses if there are problems powering up the marker. Select Fast-Blow fuses based on the local power supply. For US market machines please use (5mm × 20mm) fuses rated for 10A at 250VAC.

15. INPUT POWER Connector

Used for connecting the mains power cable. Cables are available in many common configurations and lengths.

16. Cooling Air Exhaust.

NOTE: Do ***not*** block, make sure there is enough space for adequate ventilation.

17. ETHERNET Connector

Use this port to connect the marker to a Category 5e cable if using the *WinLase* software package. Select your Category 5e cable depending on the network configuration you plan to use. If the marker is to be directly connected to a PC use a Cat 5e crossover type cable. If the marker is to be connected to a Local Area Network via a hub or switch, use a standard Cat 5e patch cable.

18. USB Connectors

This port is used to add additional job and graphic storage to the unit. Use a USB flash drive only. Not all flash drives are supported. One recommended brand/model is the *PNY Optima Attache*.

19. MAINTENANCE Connector

Reserved for factory diagnostic use.

Laser Head (Rear)

(Standard Head Shown Here)

1. Laser Scanner Connector

This 25-pin D-Sub cable contains power and digital communication signals for the laser scan head. Please do not operate the marker without this connected as it may damage the scanhead and do not plug or unplug this connector when power is **ON**.

2. Laser Head Control Connector

This 15-pin D-Sub cable contains the communication signals between the marker and laser head including head interlocks, shutter control and sensor signals, etc. If this cable is disconnected the marker will be in the **Emergency Stop** state.

3. Fiber

Optical fiber that connects the Laser output to the Laser head. ***NEVER*** allow the fiber bend radius to decrease to less than 6". ***NEVER*** kink or crush the fiber. Fiber damage will require refurbishment of the laser power supply. The fiber is shipped installed and aligned. The fiber output assembly can ***not*** be removed except by trained personnel.

Laser Head (Standard)

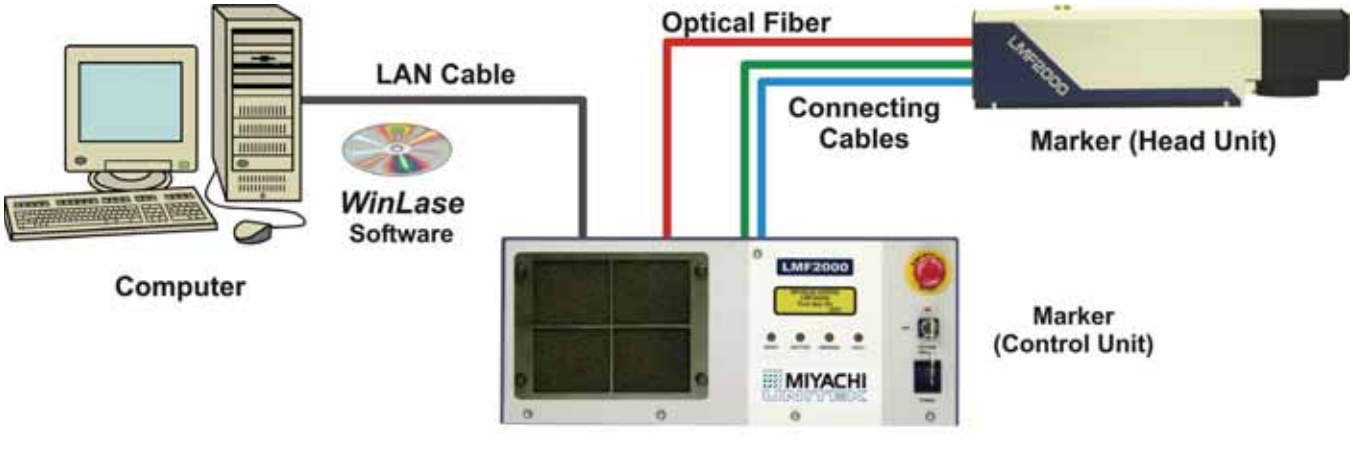
The standard laser head is a versatile unit designed for easy beam expander changes. It has a large degree of internal optical adjustability and is extremely flexible in configuration. It is not protected against harsh environments. This unit is fully interlocked, contains a bright red alignment laser and integral safety shutter.

Laser Head (Compact)

The compact laser head is a sealed unit designed for harsh industrial environments. It is protected against water splash and the ingress of dust. It is extremely small and convenient to mount. In addition, the compact laser head includes an extremely robust fixed alignment mechanism. No alignment of laser optical components is possible or necessary. This unit is fully interlocked, contains an on-axis red alignment laser and integral safety shutter. Due to its robust design for industrial applications this unit requires a slightly more involved beam expander change procedure. The no-alignment design makes this procedure simple but more disassembly is required than the standard head unit. Two sizes of compact heads are available, one for laser markers ≤20W and one for high-power models.

Section III: System Configuration

Connection Diagram



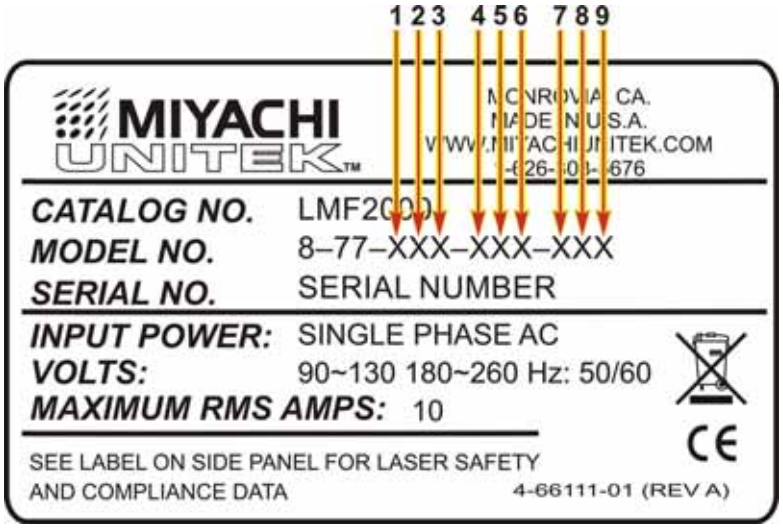
Options

The following items are available as options:

Component	Description
Protective Glass	The Marker is shipped with a protective glass affixed to the *f*-theta lens. If this glass becomes soiled or cracked a replacement may be purchased.
IMS MDrive Smart Motors	Up to 4 motors can be connected for stepper motor control with optional encoders. Contact MUC for details.
Air Filter (Control Unit)	An air filter is installed in the Control Unit at the time of purchase. Additional filters are available. MUC Part Number 4-65491-01
Lithium System Battery	CR2032 (Backup battery for internal memory) – MUC Part Number 145-017
Cat 5e Crossover Cable	For communications with a computer. MUC Part Number 205-318

Section IV: Identifying Model Number Components As Originally Shipped 8-76, 8-77 Laser Models

See Specific Unit Label for Power Requirements



1	**Laser**	P = 25W SM (Single Mode M^2<1.3, Full Waveform Control) Identifies LMF2000-SM Q = 35W HM (Full Waveform Control) Identifies LMF3500-HP, ML-7340C R = 20W HS, ST (Full Waveform Control, M^2<2.0) Identifies LMF2000-HP, ML-7320B/C T = 20W RM (single waveform) Identifies LMF2000, ML-7321B/C V = 10W RM (single waveform) Identifies LMF1000, ML-7311 B/C

2	**Control Hardware**	A = NONE L = Winlase

3	**Scan Head**	A = NONE N = XLR8

4	**Environmental**	A = Normal B = Enhanced (Compact Head, ≤20W) C = Enhanced High Power (Compact Head, >20W)

5	***f*** **Theta Lens** As shipped. Includes mounting collar.	A = NONE B = 100mm C = 160mm D = 163mm (Obsolete) E = 254mm F = 420mm G = 150mm

6	**Beam Expander** As shipped.	A = NONE B = S-7x, Standard Head Only C = S-5x, Standard Head Only D = S-4x, Standard Head Only E = S-2.6x, Standard Head Only K = G-1.5x, All Head Options L = G-2x, All Head Options M = G-3x, All Head Options N = G-4x, All Head Options O = G-6x, All Head Options P = G-8x (≤20W, special order only, consult Factory)

7	**OEM Labeling**	A – Miyachi Unitek Corporation LMF Series B – Miyachi Corporation ML-73xx Series

8	**Control Option** As shipped.	A = NONE B = PC C = Notebook PC D = Pendant

9	**Reserved**	A = NONE

CHAPTER 2
INSTALLATION AND SETUP

Section I: Planning

When planning for the installation of the Marker, make sure that the following conditions are met:

- The Marker should be placed in a dedicated laser operation area. The person responsible for the area (the Laser Safety Officer) must isolate the laser operation area from other work areas and display signs warning that the laser operation area is off limits to unauthorized personnel.
- See *Appendix A: Technical Specifications* for specific weight and dimensional requirements.
- Use proper tools (wire strippers, pressure wire connectors, etc.) for termination of the connecting cables.
- The Marker should be placed on a firm, level surface that is free from vibration. Install the head and workpiece on the same stand to prevent marking distortion due to vibration.
- Mount the head on a platform of sheet metal at least 0.394" (10 mm) thick.
- Do ***not*** operate the unit where there is considerable dirt, dust, oil mist, chemicals, fumes, moisture, or near a high-frequency noise source.
- The ambient temperature should be between 41°F to 95°F (5°C to 35°C) free of sudden temperature fluctuations and a relative humidity less than 90% (non-condensing). The area should have no rapid temperature fluctuations, which may cause dew condensation on the optical surfaces.
- If the outside of the Marker is stained, wipe it with a dry or slightly moistened cloth. If it is badly stained, use a neutral detergent or alcohol to clean it. Do ***not*** use paint thinner, acetone, benzene, etc. which can discolor or deform the parts.
- To accommodate the standard cable lengths, the Control Unit, and Laser Head must be located within 2m (6.5 ft) of each other. The computer can be located anywhere as long as both the marker and PC have network accessibility.
- When installing two opposing laser heads, install them so that neither unit is aimed at the other.
- Install the Laser Head in any orientation desired.
- Make sure that the bend radius of the optical fiber on the back of the Laser Head is greater than 6" (15 cm).

Allow adequate clearance on all sides of the Marker to allow for cooling, maintenance and servicing. Both the Control Unit and the Laser Head should have a minimum of 6 inches (15 cm) behind the units for proper ventilation and cable clearance. The Marker Control Unit is designed for standard 19" rack mounting. Ensure that the front and back of a rack mounted unit are clear for proper ventilation. Observe the other ***minimum*** distances shown below:

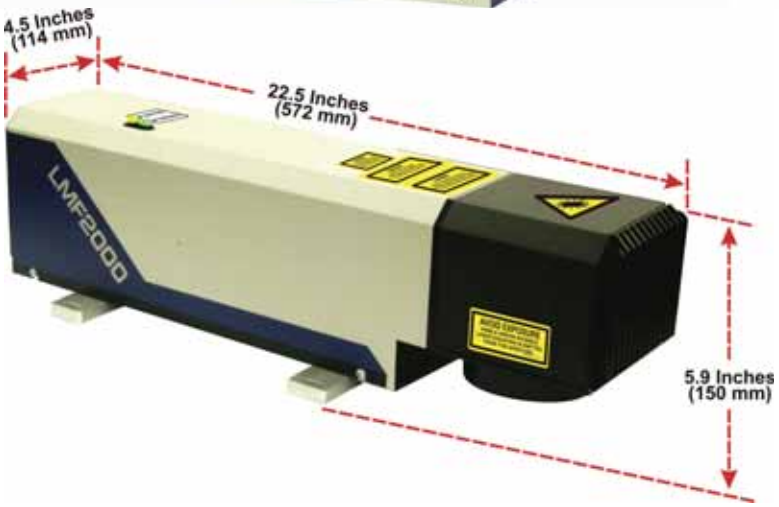
When locating, ensure that the front air filter and side and rear exhausts are clear for proper ventilation. A minimum of 6" clearance behind the control unit is required for proper fiber routing.

If mounted in a 19" rack ensure that the front and back are clear and the side vent is not completely blocked.

The laser head is a sealed unit that requires no airflow clearance considerations other than exposure to ambient air.

Ensure that there exists at least 6" of clearance behind the marker head for cable and optical fiber clearance.

NOTE: There are minor differences on the rear connector panel on each marker head but the connections and functions are identical for all heads.



Standard Head

Compact Head ≤ 20W

Compact Head > 20W

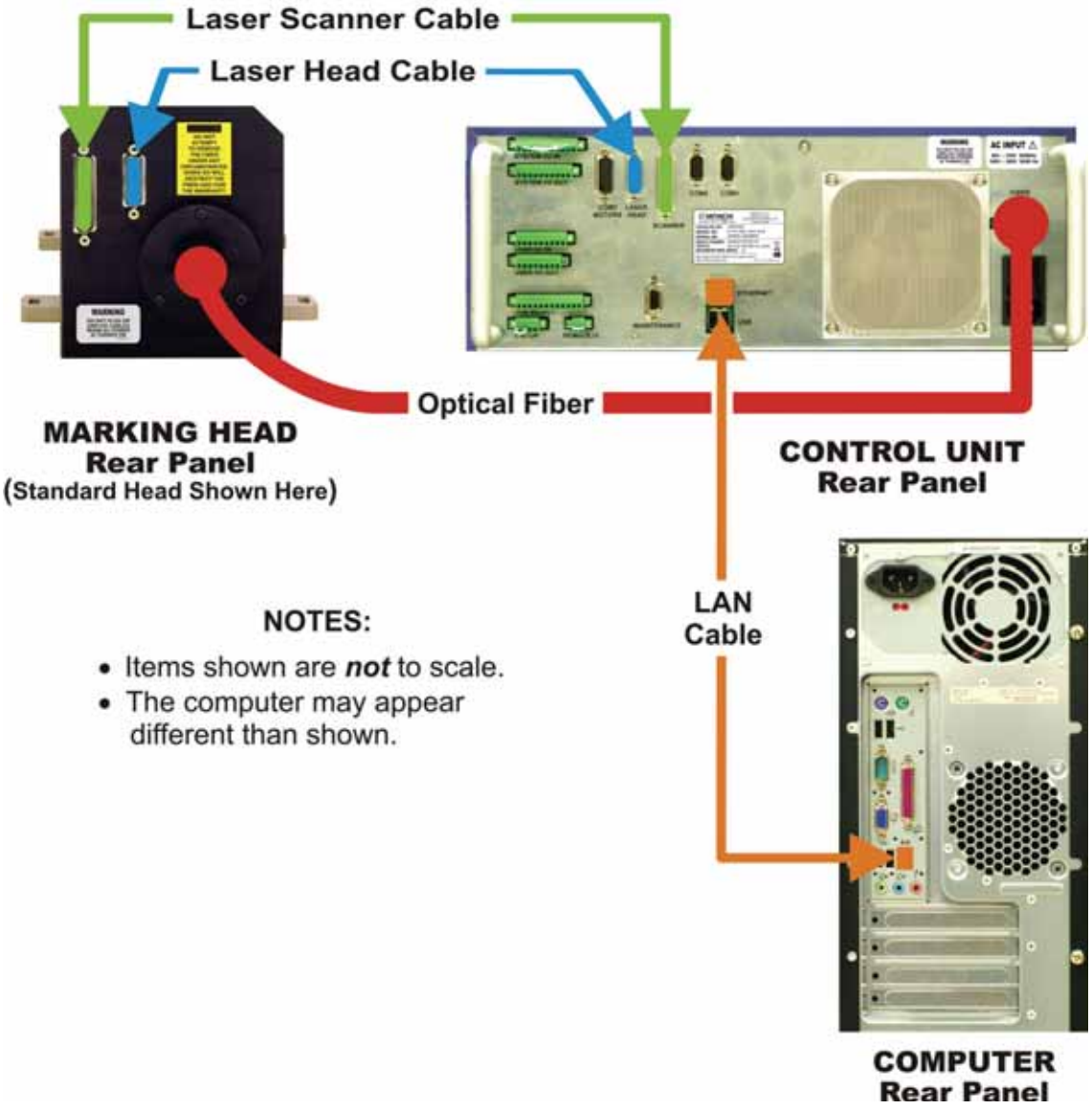
Section II: Installation

CAUTION

Do ***not*** attempt to remove the fiber at the rear of the marker head under any circumstances. Doing so will ***destroy*** the fiber and void the warranty. The Miyachi Unitek Corporation assumes no liability for such action, the fiber will have to be replaced at the customer's expense.

Connect the Signal Cables

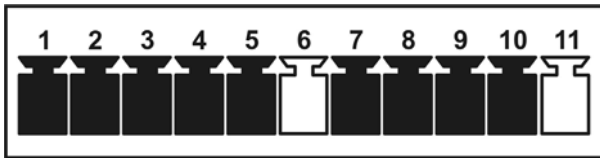
Connect the interconnect cables between the Marker Control Unit, Marker Head, and the Computer. Connect all cables as shown below. Secure all cables by fastening the connector backshells.



Verify the I/O Configuration is Correct

System I/O Inputs

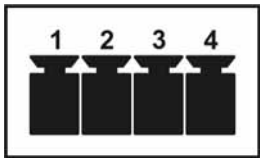
If you plan to use customized External I/O, make sure that it is configured in accordance with *Appendix B, Electrical and Data connections*. If you will not be operating the Marker through the External I/O, verify that the factory-installed **System I/O In** jumper is configured as follows:



Jumper #1	Pin 2 connected to Pin 8 (Laser Enable)
Jumper #2	Pin 4 connected to Pin 8 (Shutter Open)
Jumper #3	Pin 9 connected to Pin 10 (Optocoupler Bias)
Jumper #4 (Not shown above)	Pin 3 connected to Pin 8 (Guide Beam On, OPTIONAL)

Emergency Stop

If an external **Emergency Stop** is to be used, make sure that it is wired in accordance with *Appendix B, Electrical and Data connections*. If you will not be connecting the external **Emergency Stop** function, verify that the factory-installed jumper is configured as follows:



Jumper	Pin 1 connected to Pin 2

Remote Interlock

If the **Remote Interlock** is to be used, make sure it is wired in accordance with *Appendix B, Electrical and Data connections*. If you will not be connecting the Remote Interlock function, verify that the factory-installed jumper is configured as follows:

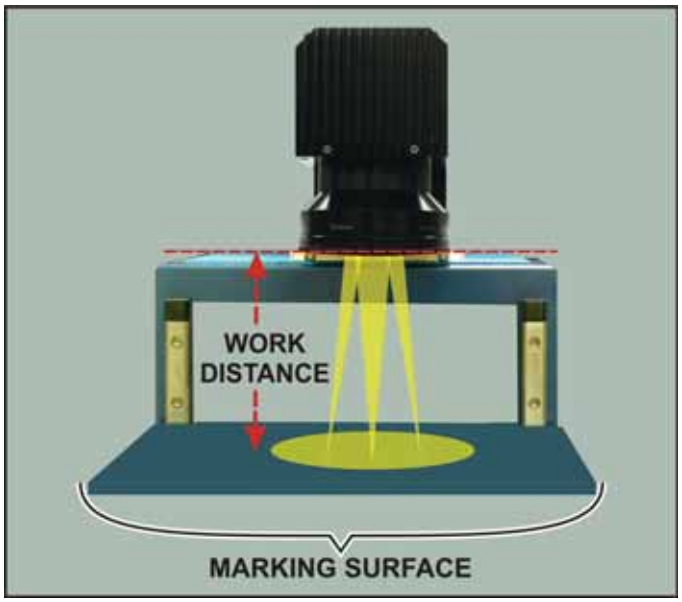
Jumper	Pin 1 connected to Pin 2

Set the Working Distance

In order to properly mark material, the working distance must be set in order to focus the laser beam onto the surface of the marking material. An incorrectly set working distance, will not produce an acceptable mark.

Set the distance as shown on the right. If the marking quality is unacceptable, then slightly adjust the marking material towards and away from the lens until the marking quality is acceptable.

Working Distance	
***f*-theta lens**	**Working Distance**
f 100 mm	3.86 ± 0.04 in. (98 ± 1 mm)
f 150 mm	6.69 ± 0.08 in. (170 ± 2mm)
f 160 mm	6.90 ± 0.08 in. (176 ± 2mm)
f 163 mm (Obsolete)	7.28 ± 0.08 in. (185 ± 2mm)
f 254 mm	11.65 ± 0.12 in. (296 ± 3 mm)
f 420 mm	19.45 ± 0.20 in. (494 ± 5 mm)



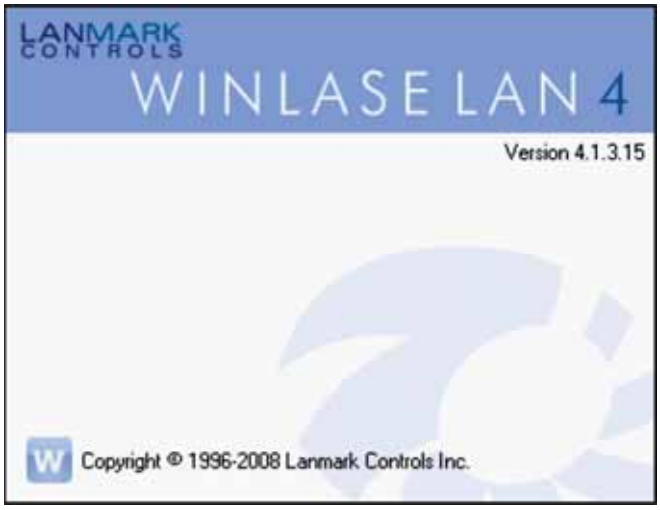
Section III: Software Installation and Set-up

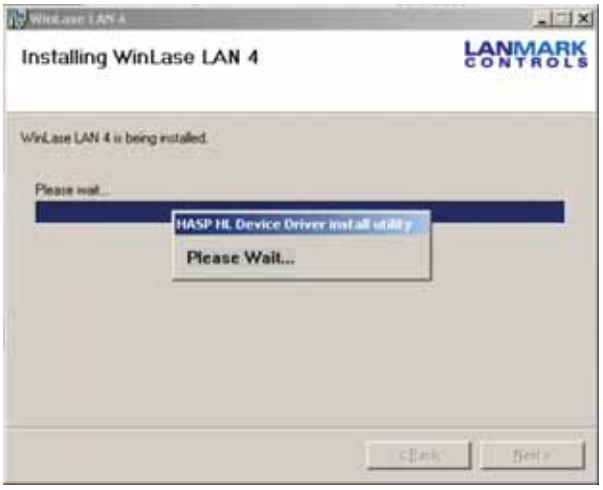
When Using a Factory-Supplied Computer

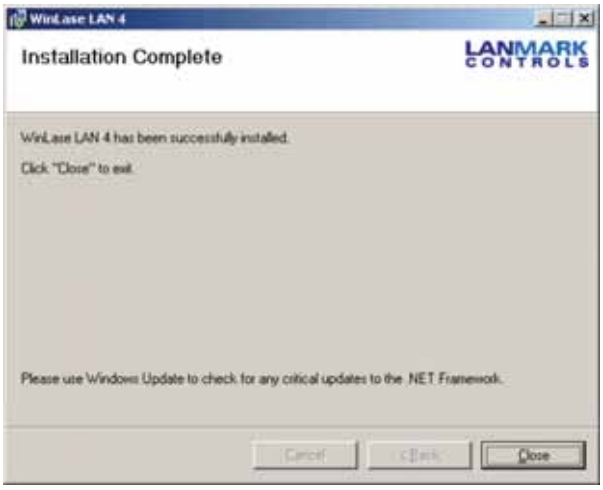
WinLase Marker Software is factory-installed making the Marker ready-to-use. No further installation is required. Please ensure the USB hardware key (dongle) is present for operation. Go to *Chapter 3* for operating instructions.

When Supplying your own Computer

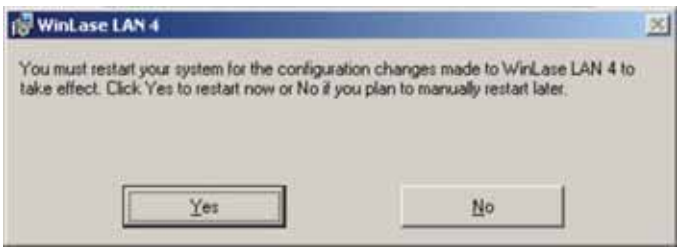
1. Insert the *WinLase* CD ROM into your computer and follow the installation procedures.



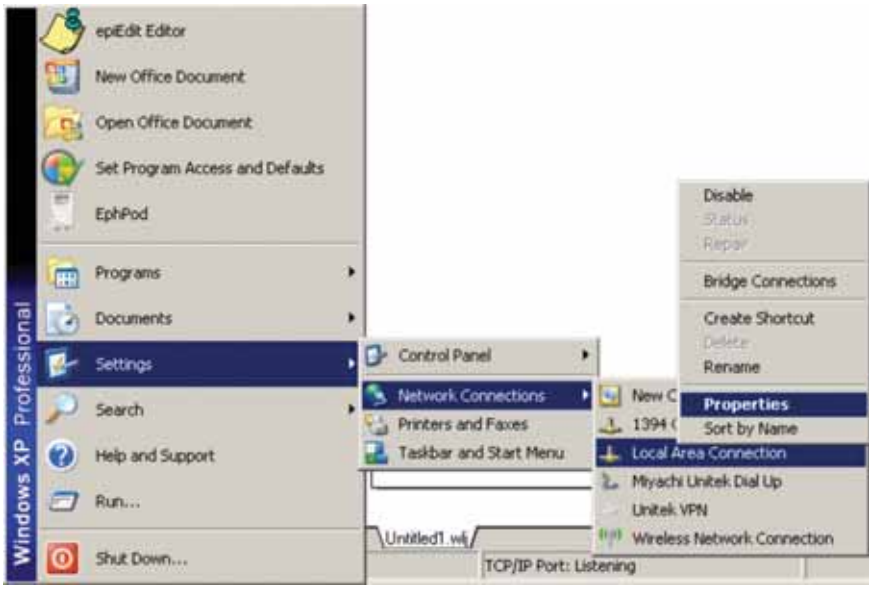




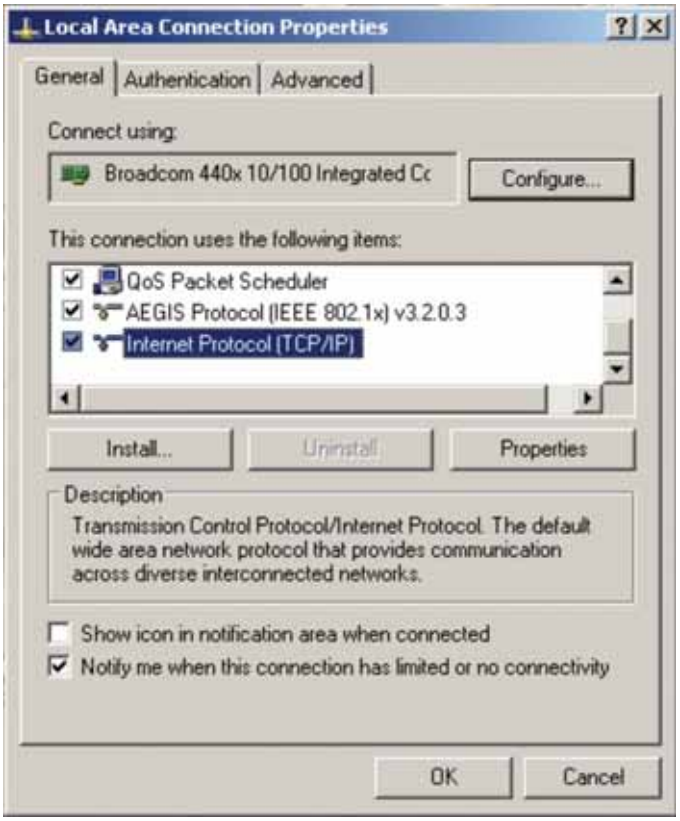
NOTE: When installation is complete you will see the "restart" prompt as shown on the right. Select **No**, because other changes need to be made ***before*** restarting your computer.



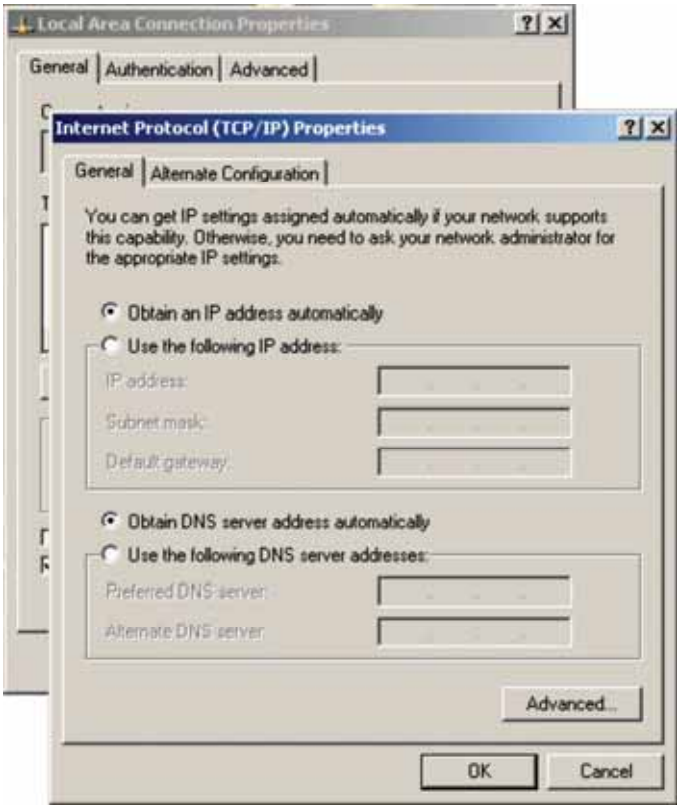
2. Please verify correct network settings. The Marker and Control PC must be on the same subnet for DHCP configuration or the static IP addresses must be assigned and visible.
3. View the properties of the **Local Area Connection**.



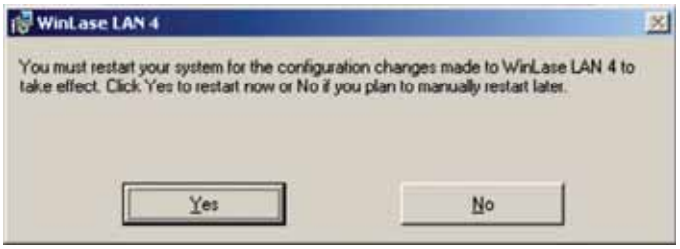
4. Select the Internet Protocol (**TCP/IP)** connection and select **Properties.**



5. If the Marker is connected directly to the PC via a crossover Cat 5e cable, set the PC to obtain the IP and DNS information automatically as shown, followed by **OK**. Otherwise, enter the static IP address (if known).

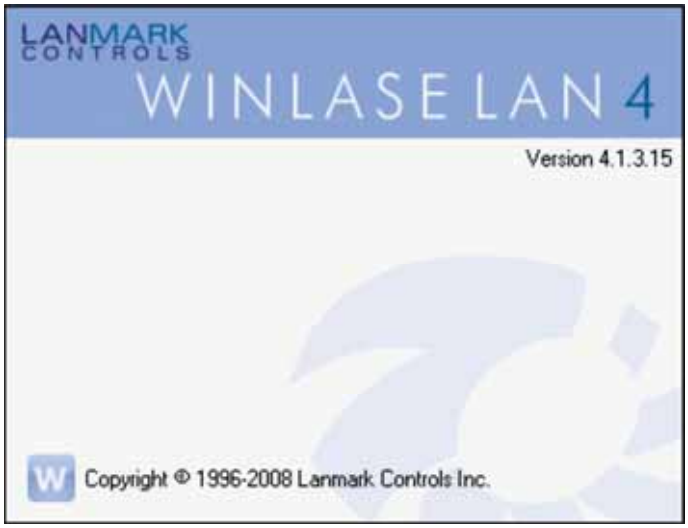


6. Click **Yes** to restart your PC.



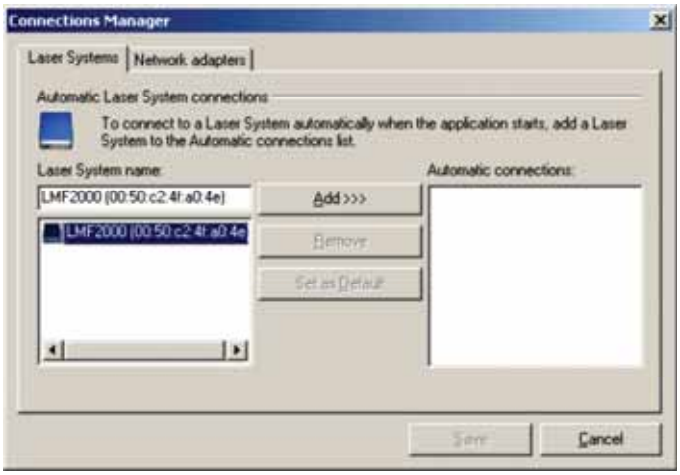
7. Restart *WinLase* and turn the Marker ON.

   NOTE: Allow 30 - 60 seconds for the Marker to finish booting and be ready for network operations.

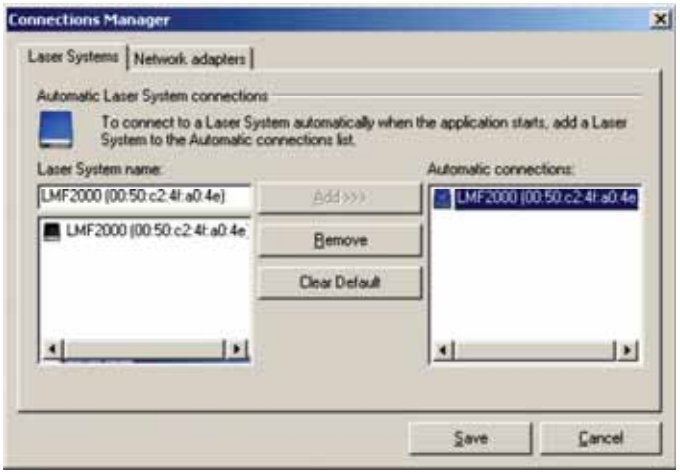


8. Once *WinLase* has restarted, select **Manage Laser System Connections** from the **Tools** pull-down tab

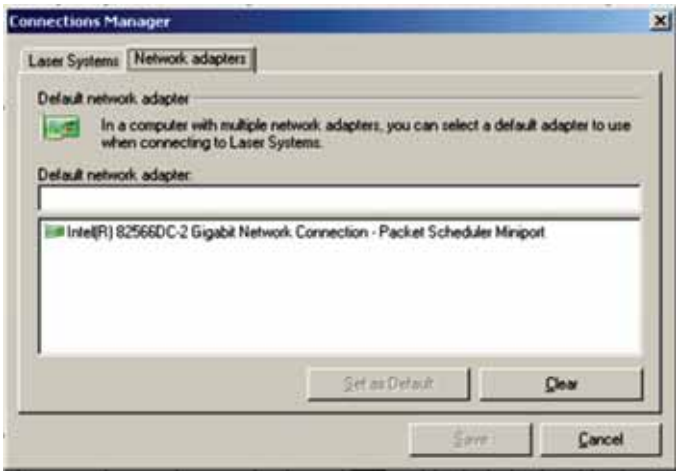
NOTE: If the Marker has booted and the network settings are correct, it will appear under **Laser System** in the left hand box of the **Connections Manager**. If the Marker has not completed the boot process you will see **Waiting for system connection** in the window.



9. Select the appropriate Marker and click **Add >>>**. If there are multiple devices, choose one and set it as the default device. Click **Save** to save these settings.

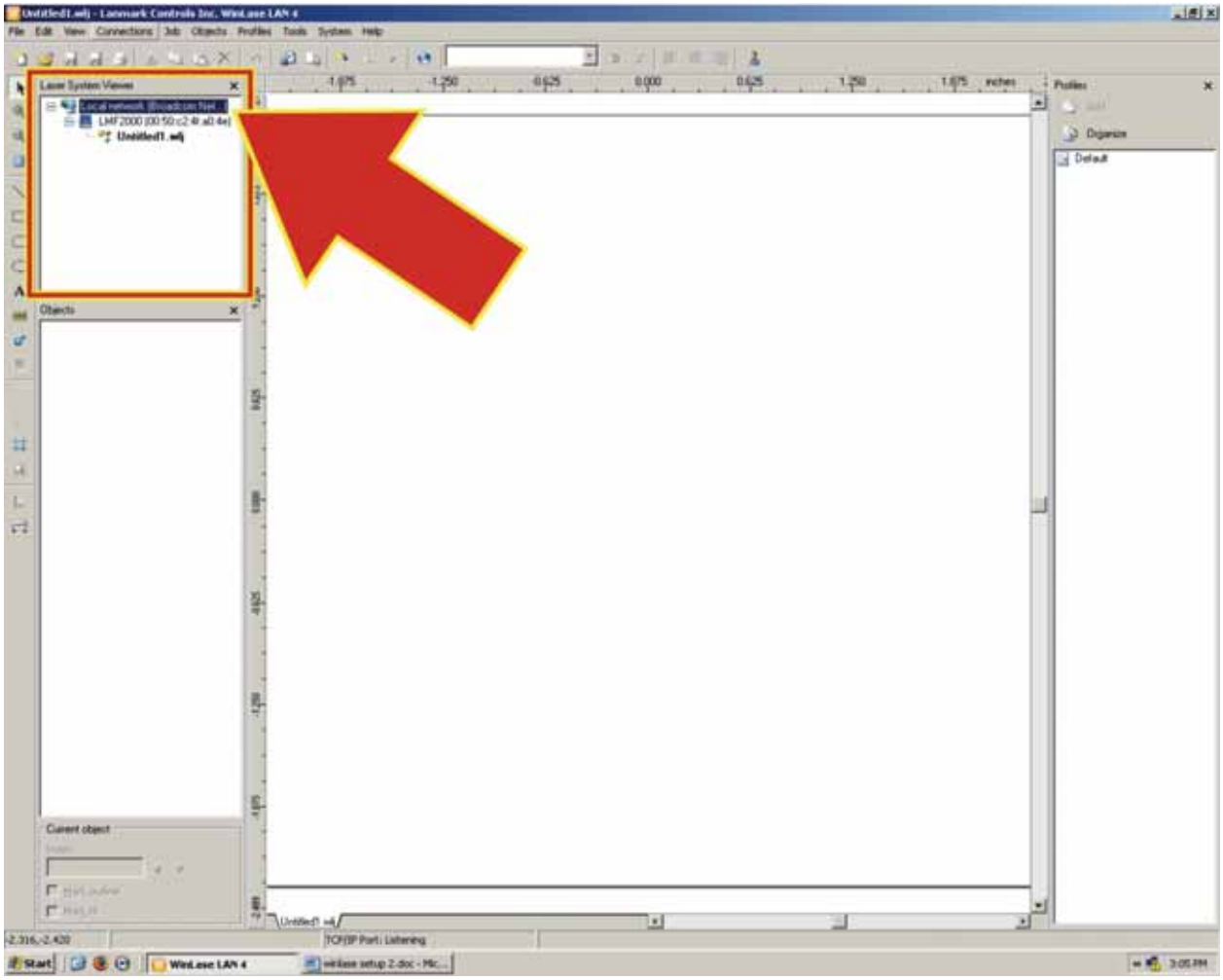


10. If your machine has multiple network adapters and the wrong one appears in the Laser System Viewer, you can use the "Network Adapters" tab in the Connections manager to select the correct default adapter.



Once the device has been added to the **Automatic Connections** list, a window will pop up indicating the connection status. Please wait until the status changes to **Connected** and the window has closed.

The connected device will be present in the **Laser System Viewer** window in the upper left corner of *WinLase*. A blue icon indicates a TCP/IP connection is present.

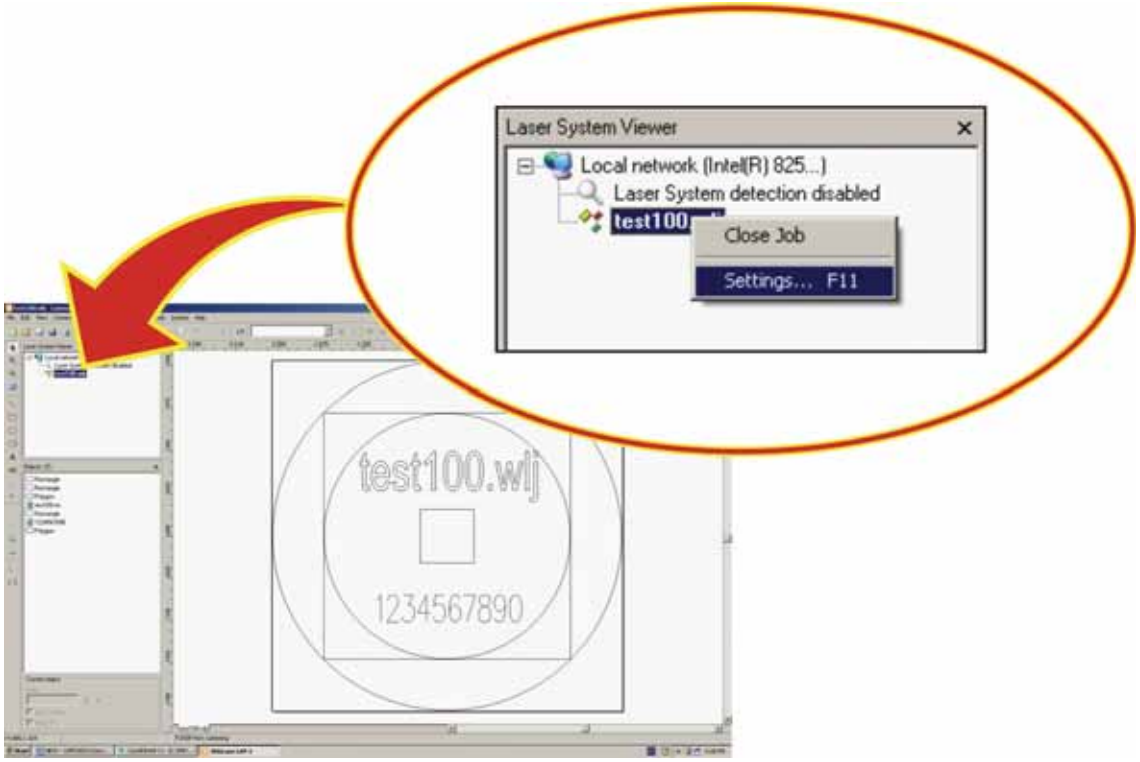


11. Select the pull-down **View** menu, and select a measurement standard.

    You are now ready to create a new marking job and operate the marker. Please refer to the *QuickStart Guide for WinLase* (Part Number 990-550) or the Marker **Help** file for more information.

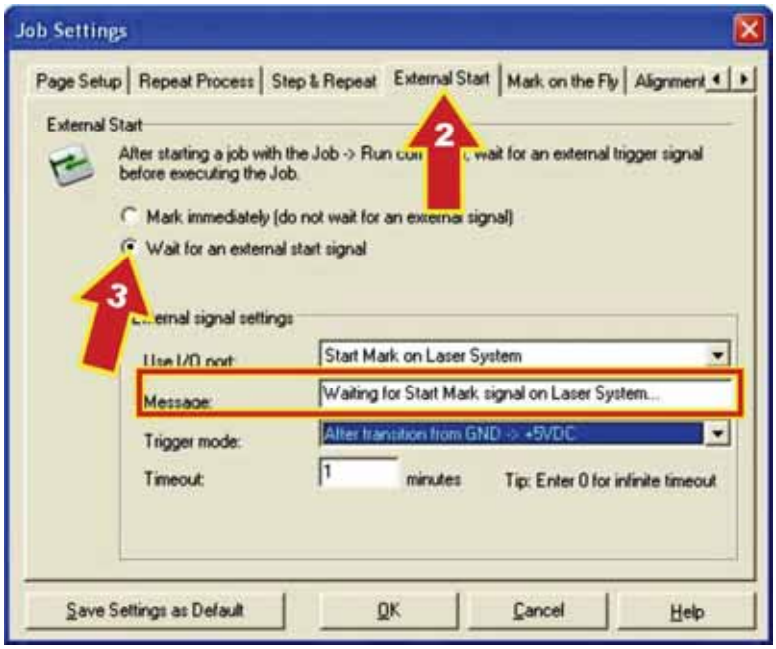
Section IV: External Start

1. Right click on the job and click **Settings… F11**.



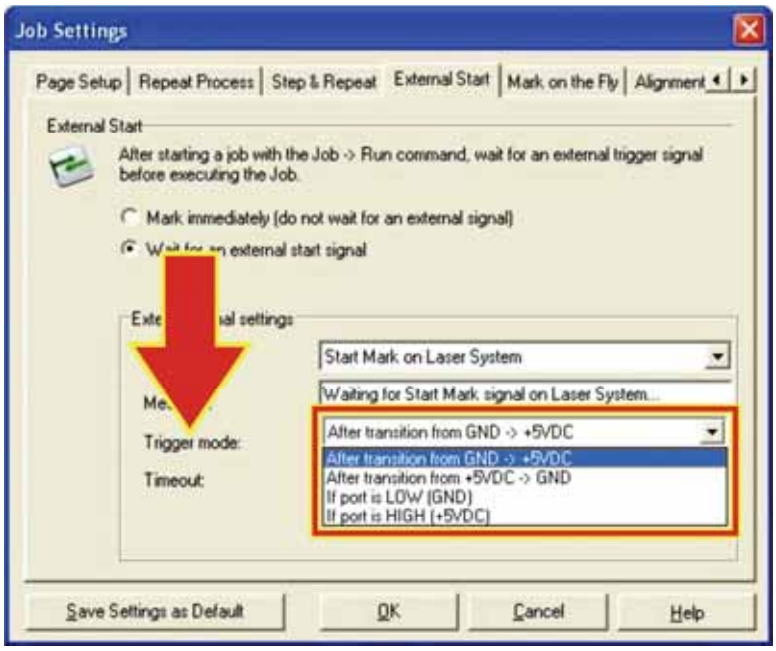
2. Select the **External Start** tab.
3. Click on **Wait for an external start signal**.

   NOTE: You may change the user **Message** if you wish. **Highlight the existing message**, then type in the new message.

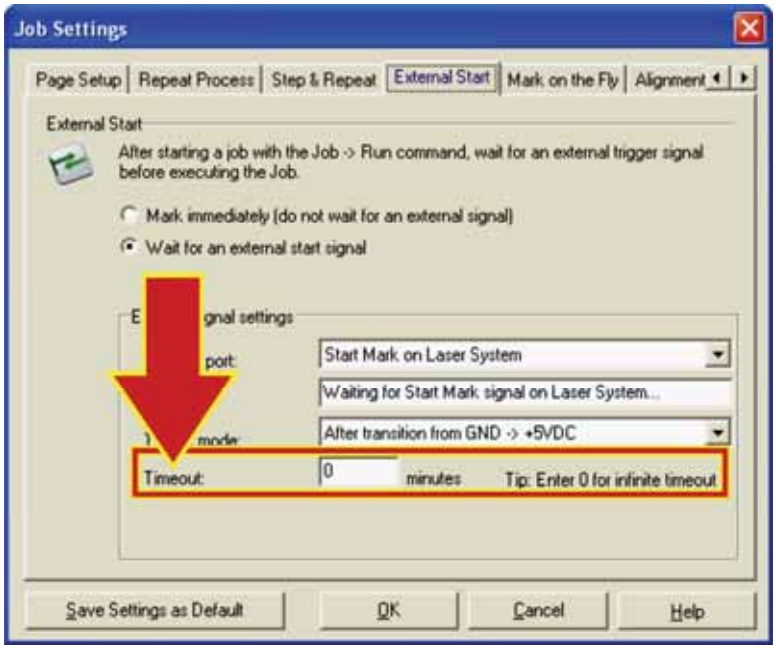


4. Make sure that the **Trigger Mode** is set to occur on a transition, **0 ➔ 5V or 5V ➔ 0** depending on if you want to trigger on the rising or falling edge of the signal.

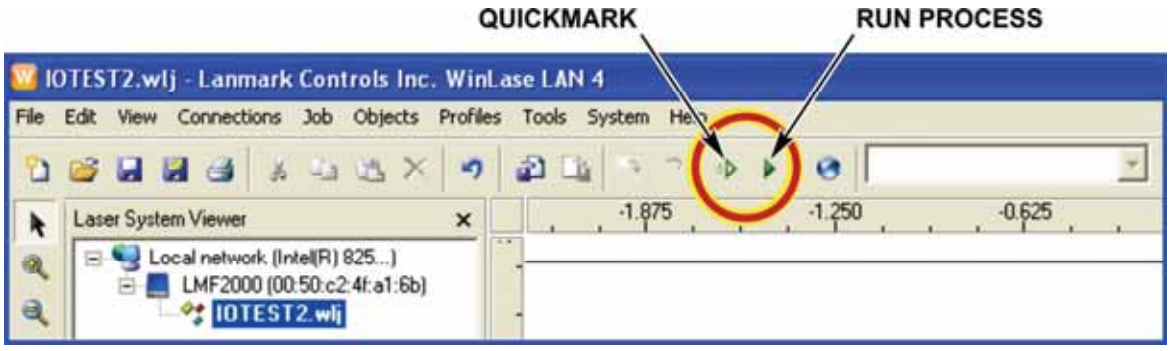
   To change the **Trigger mode**, click on the ▼ arrow, select another mode.



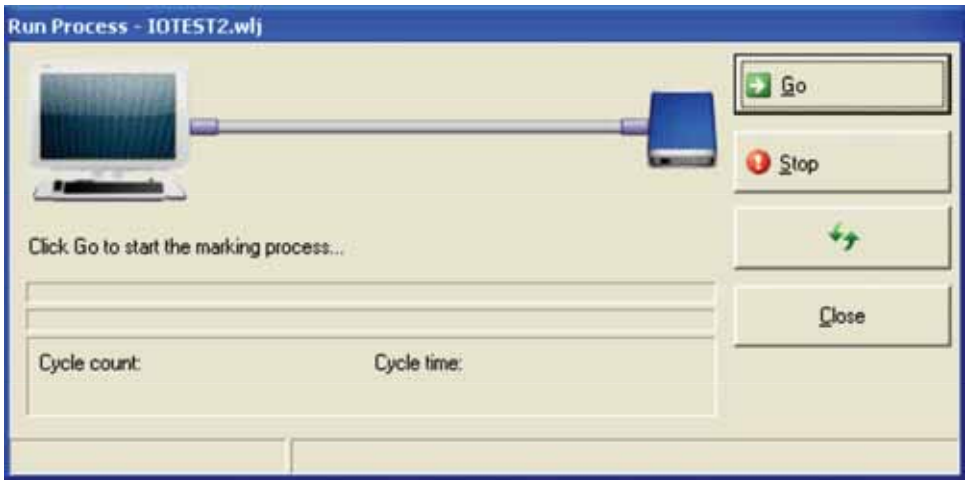
5. You may change the **Timeout** if you wish. **Highlight the existing time**, then type in the new number of **minutes**.



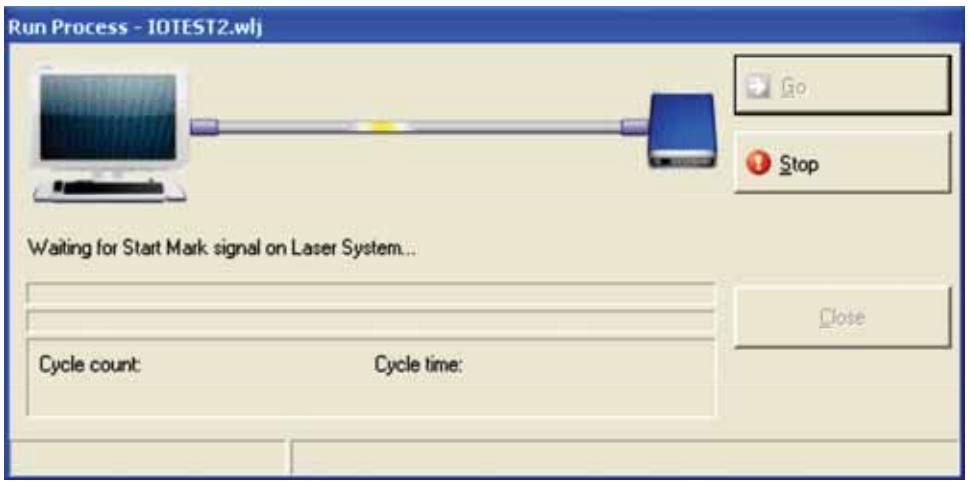
6. After you click **OK**, in the **Job Settings** screen, click on the **Run Process** icon shown below.



7. When you see the screen below, click on **Go**.



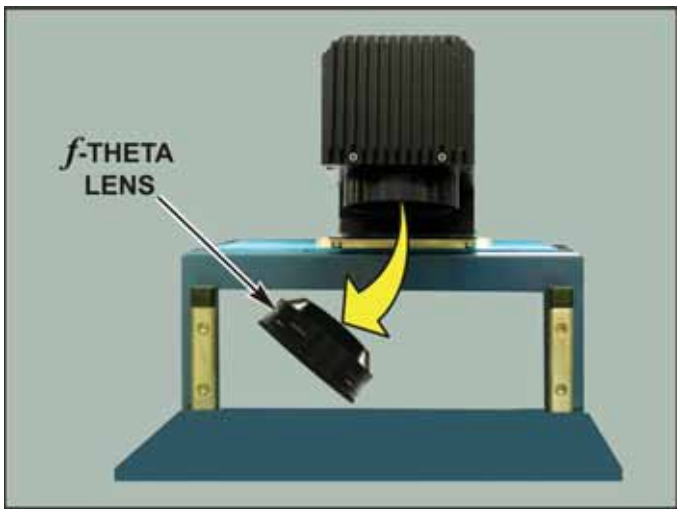
8. As soon as you see the **Waiting for Start Mark** message (or your own custom message if you changed it) you may begin marking using your normal external I/O Start Mark signal (foot pedal, START switch, PLC trigger, etc.).



Section V: F-Theta Lens Configuration

After installing the new *f*-theta lens, select the **System** pull-down menu, select **Preferences**, and select the **Hardware** tab. Select the lens and click **Change** to select the new *f*-theta lens.

If you wish to adjust the scaling, rotation, or offset you can click **Calibrate** and follow the instructions provided.



CHAPTER 3
OPERATING INSTRUCTIONS

Section I: Before You Start

Safety Precautions

DANGER

Always wear protective goggles when operating the Marker. Goggles ***must*** have an optical density of ***at least 7***$^{+}$ at a wavelength of 1060-1150nm for the marker. The central emission band is 1064 ± 5nm. ***Never*** look directly into the laser beam.

WARNING

Never operate the Marker in any manner ***other*** than described in this manual. Doing so may expose personnel to laser radiation or electrical hazards.

Before attempting to operate the Marker, have ***all*** personnel who will be working with the Marker read this manual ***and*** the *Laser Safety Manual* (Part Number 990-502) thoroughly.

Notes

- Verify that the electrical supply meets the electrical requirements, as shown in *Appendix A: Technical Specifications*. The electrical supply must meet all applicable local, state, and federal safety standards.
- ***Before*** operating the Marker, be sure the protective lens cover is removed from the output lens.
- Operate the switches and buttons carefully by hand. If they are operated roughly or with the tip of a screwdriver, a pen, etc. they may break or malfunction.

Section II: Operation

Operate the Marker following the procedures written in the 990-550 *Quickstart Guide for the WinLase LAN* Software manual. This *Quickstart Guide* only contains brief instructions in order to get you started with basic marking right away. If you need more detailed information, please refer to the complete **OEM Reference Manual** containing detailed instructions of all *WinLase* features. This Reference Manual is available through Miyachi Unitek. To get a copy of the **OEM Reference Manual**, please refer to the **CONTACT US** information located in the front section of this manual.

CAUTION

As you use the Marker, be sure to **SAVE** the created data periodically onto a floppy disk or other backup medium. If there is trouble with the Marker's hard disk drive, ***all the data saved on it will be lost.***

NOTE: If you are using software other than *WinLase*, contact the software manufacturer for their appropriate operating manuals and other technical updates.

Turning the Marker ON

1. Verify that the **INPUT POWER SWITCH** on the back of the control is turned ON.

2. Turn the **POWER SWITCH** on the front of the Marker ON. This will start a series of messages in the LCD Display Screen.

If the **SYSTEM ENABLE** key switch was turned OFF, when you turned the power ON, the **Laser Hours** message on the right will display. If a RM laser module is installed the Laser Hours display will read 00000 as status communication is not available with the laser module.

3. If you see the **Turn Key On** message, turn the **SYSTEM ENABLE** key switch ON. If the key switch was already ON, this message will not display. The temperature displayed is the laser module diode temperature. It will not be present if a RM laser module is installed since status communication is not available with the laser module.

The **Shutter Closed** screen will display until the rear I/O shutter input is activated.

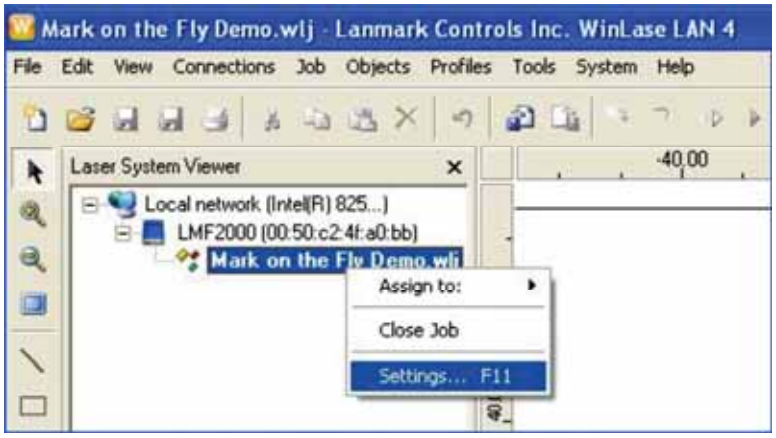
NOTE: If a the rear I/O shutter input is jumpered as provided by the factory the shutter will open immediately

The **Shutter Open** screen indicates that the shutter is now open, but the laser is not yet ready to fire

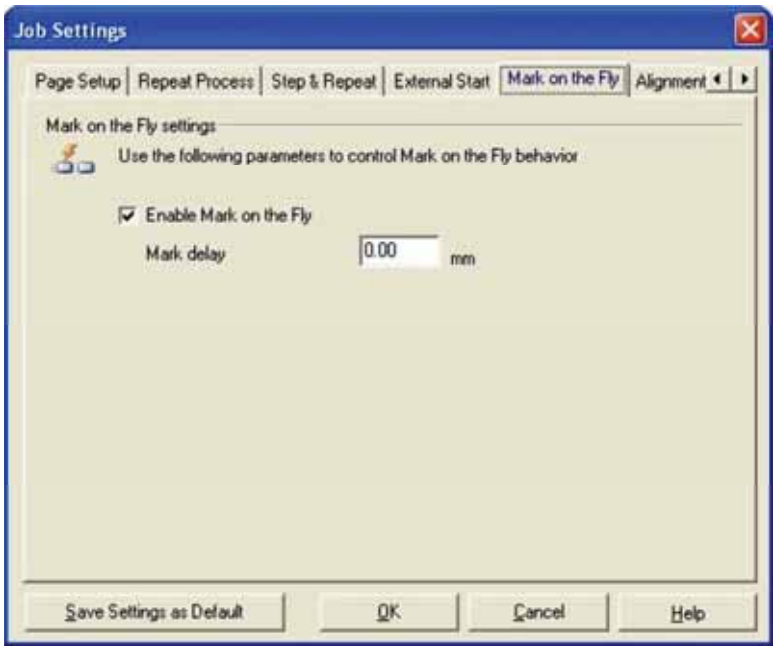
When you see this **Ready** message, the laser is ready for operation.

Marking On the Fly

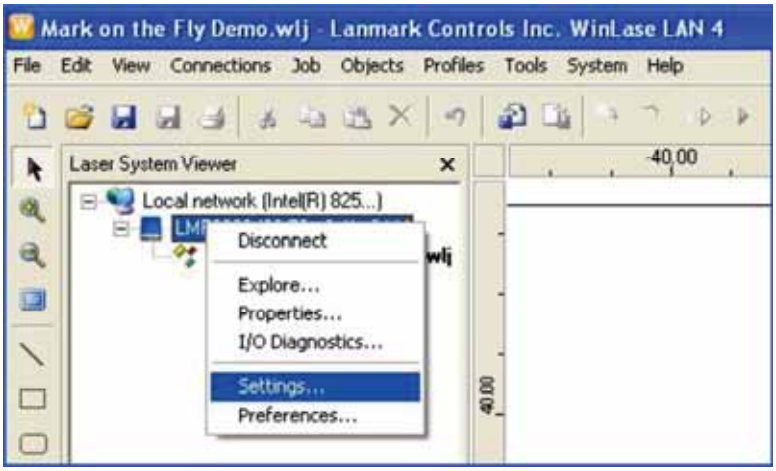
1. Configure **Mark on the Fly** settings. Right-click on a job in the **Laser System Viewer**.
2. Click on **Settings**.



3. Click the **Enable Mark on the Fly** checkbox.
4. Enter a **Mark delay** in millimeters of workpiece travel if desired.

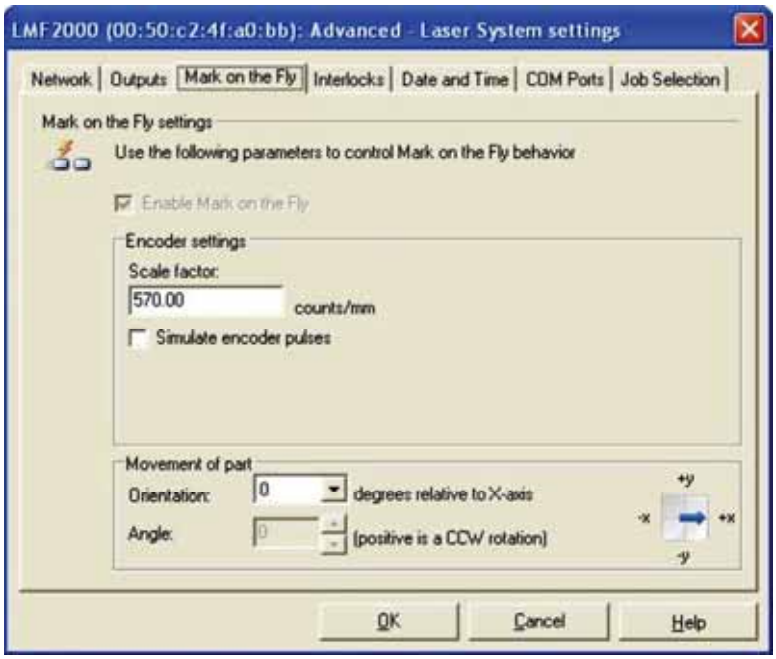


5. Right click on the marker in the **Laser System Viewer** then click on **Settings**.



6. Select the **Mark on the Fly** tab.
7. Set a **Scale factor** for the optical encoder by setting the orientation of the part relative to the marker's X-axis then calculate the **counts/mm** of your equipment.

   NOTE: If you wish to use a simulated encoder (***not*** recommended), use the **Simulate encoder pulses** checkbox to configure the simulation, then click **OK** when finished.
8. Configure the job start as explained in *Chapter 2, Section IV: External Start.*
9. Connect the encoder to the Marker per the diagram on page B-19 in *Appendix B, Electrical and Data Connections.*

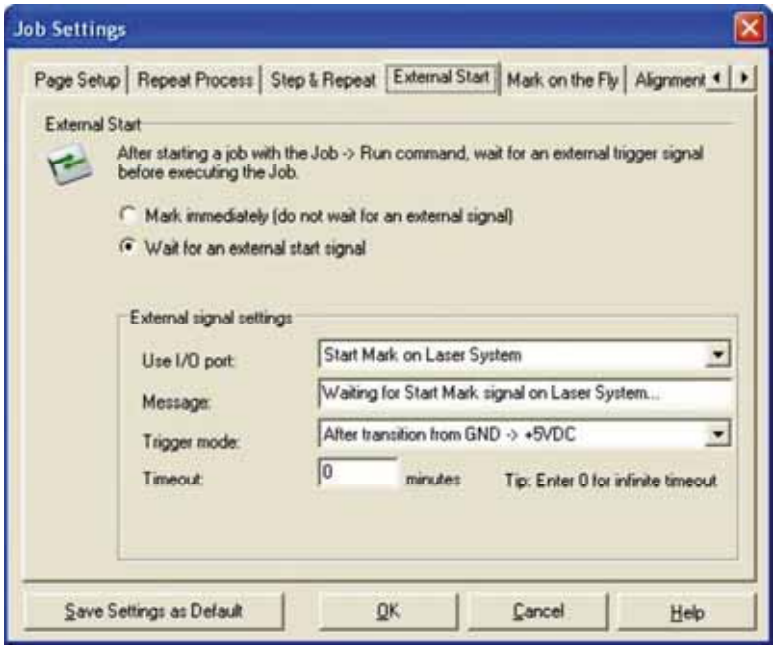


I/O Job Setup

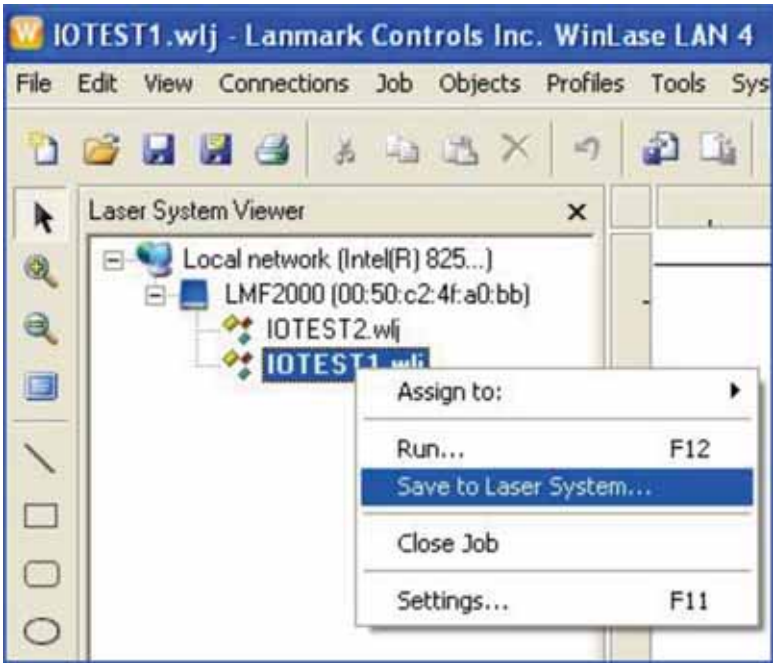
Set up the external start requirements for the jobs you wish to load over I/O.

1. Right click on the job file, and click **Settings**.

2. When you get to **Job Settings**, click on the **External Start** tab, then select the parameters you want for your marking job.

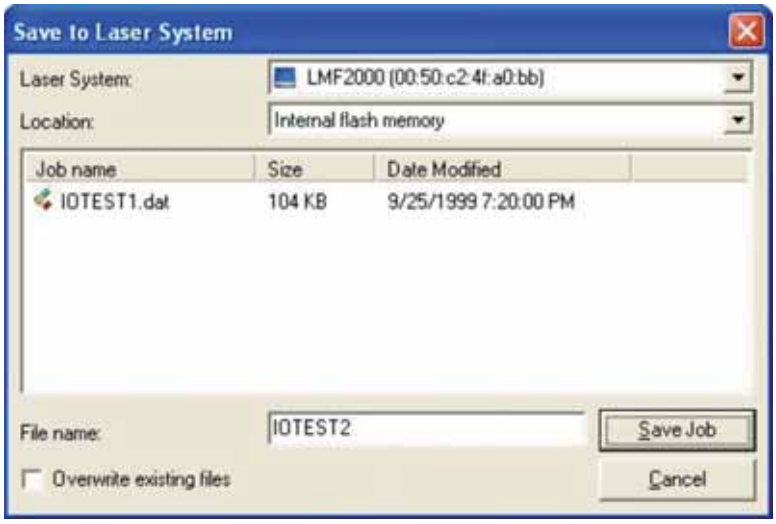


3. To save the job(s) to the laser internal memory, right click on the job, and click **Save to Laser System**.

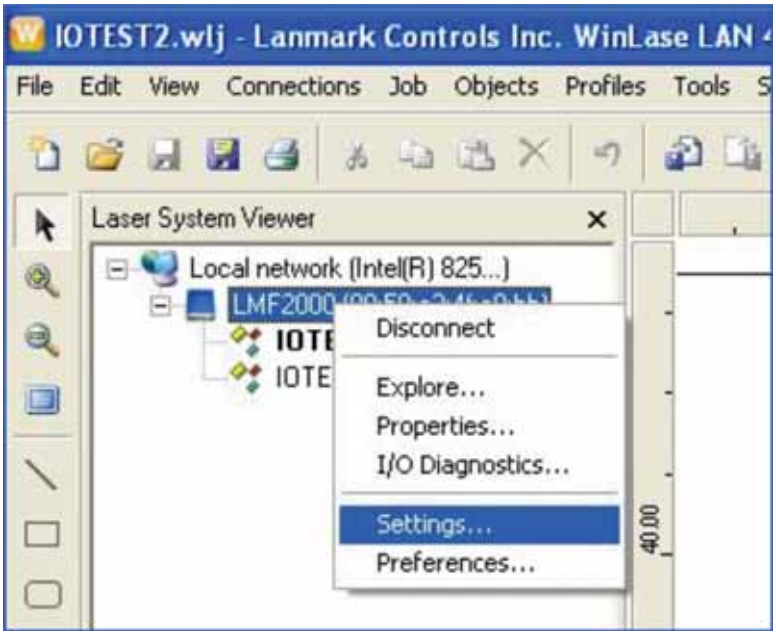


4. If you want to change the name, go to **Location** and select either **Internal flash memory** or **USB memory stick**.
5. When the memory you want displays, go to **File name:** and type in the name for the file, then click on **Save Job**.

6. Repeat Steps 1 to 5 for all the additional jobs you want to store.



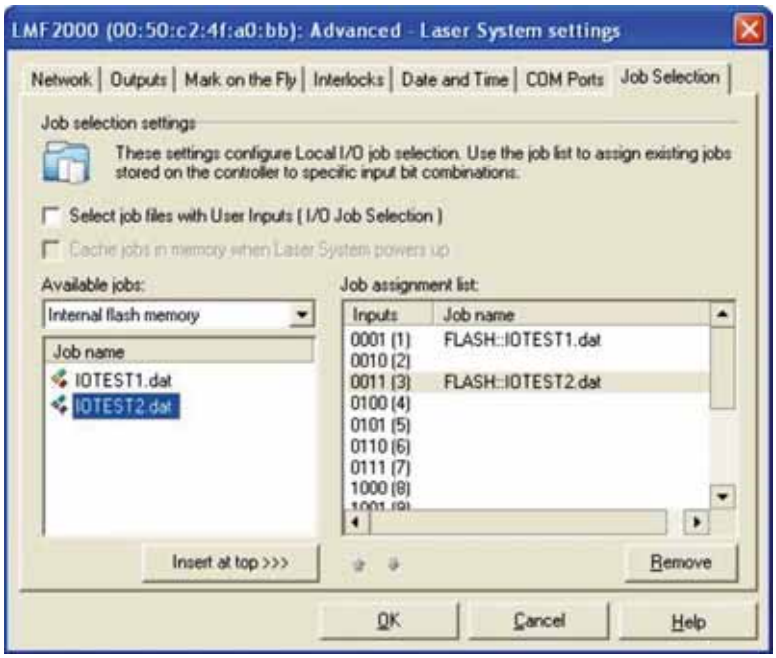
7. To enable **I/O Job Select**, right click on the marker and click **Settings** to get the **Advanced Laser Systems settings** menu on the next page.



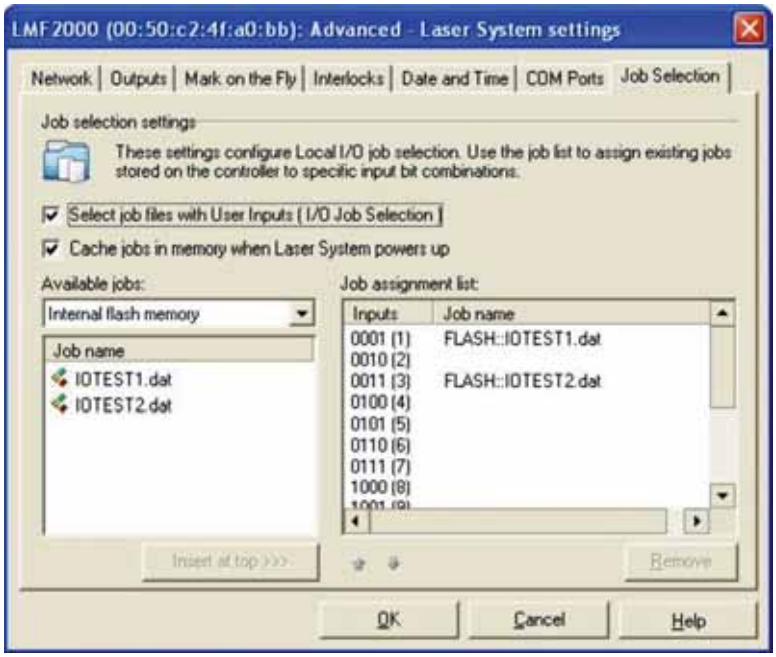
8. From the **Advanced Laser Systems settings** menu, click on the **Job Selection** tab.

9. Drag the jobs into the list on the right of the screen, or use the **Insert at Top** button.

   NOTE: The "input" code shown is the binary input combination that selects this job. This will be four or eight bit depending on the version of laser marker used.



Once the jobs have been assigned, decide whether or not you wish to use **Cache jobs in memory when Laser System powers up.** This cached mode loads jobs in <50ms. The disadvantage is that this mode stores all jobs into RAM. If there is more job data than available RAM an **Error** will occur.



Not selecting Cached job load mode will allow use of all laser memory, but jobs can take up to one second per mark object to load. Typically it is best to use Cached job load mode and keep the job size down so that the 10MB of RAM is not used up. Typical job files are only 100kB, but large files such as Bitmaps can use a great deal of memory.

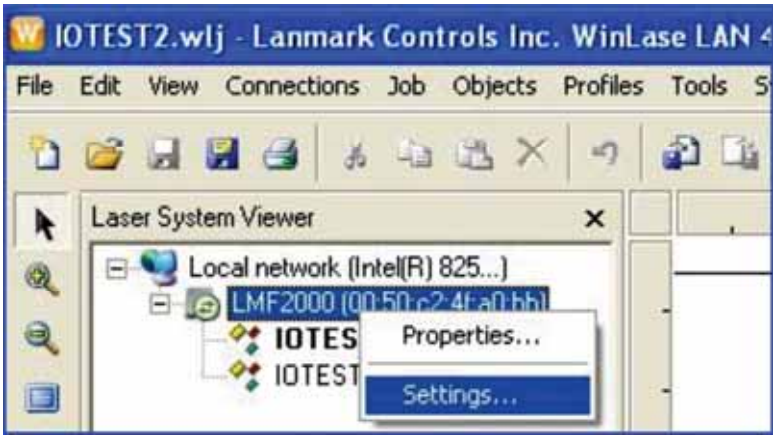
10. Click **Select job files with User Inputs (I/O Job Selection)** and click **OK** to enter **I/O job** mode.

NOTE: The laser icon in the upper left will change first to a Down arrow (downloading jobs into RAM) and finally to an icon showing two arrows pointing in opposing directions (right). This means that you are ready to select a job using the input bits on the back of the unit.

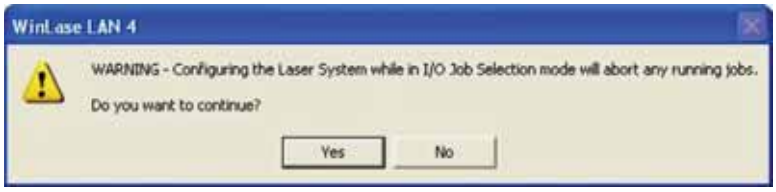
Please refer to the timing diagrams in *Appendix B, Electrical and Data Connections, Section II, Timing Diagrams* to operate in I/O job selection mode.

I/O Job Quit

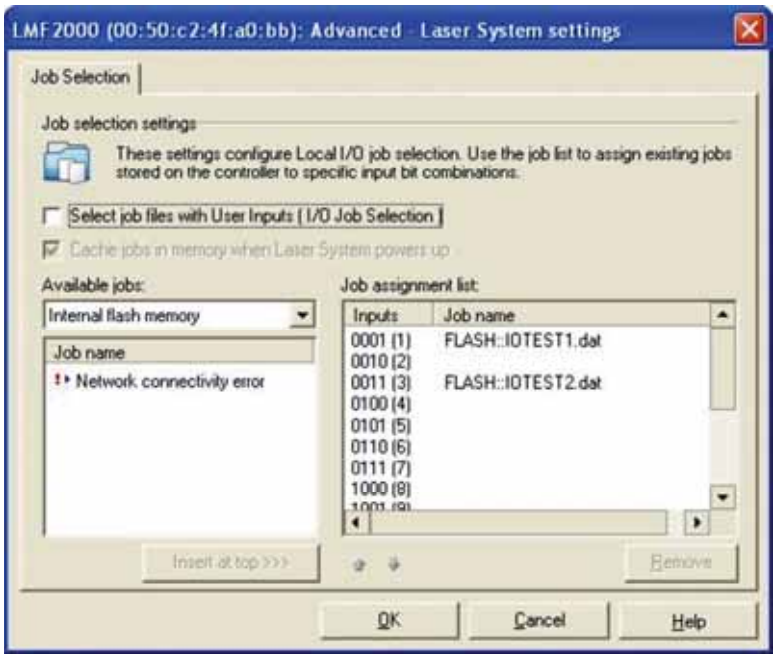
1. To exit I/O Job Selection mode, right click on the laser in **Job Select** mode and click **Settings**.



2. You will be warned that any in-process job will be aborted. Click **Yes** to proceed



3. Uncheck **Select job files with User Inputs (I/O Job Selection)** to exit this mode.



CHAPTER 4
MAINTENANCE

Section I: Safety Precautions

DANGER

Always wear protective goggles when maintaining the Marker. Goggles ***must*** have an optical density of ***at least*** 7^{+} at a wavelength between 1060-1150 nm for the Marker. The central emission band is 1064nm ± 5nm. Never look directly into the laser beam.

WARNING

- ***Before*** starting any maintenance procedure, read ***all*** of the instructions, including all caution and warning messages.
- ***Before*** starting maintenance work, turn the power to the Marker OFF and disconnect it from input AC power. Wait at least 5 minutes after turning OFF power before starting work.
- Keep the exterior of the Marker clean. If the outside of the Marker is stained, wipe it with a dry or slightly moistened cloth. If badly stained, use a mild detergent or alcohol to clean it. Do ***not*** use paint thinner, acetone, benzene, etc. which can discolor or deform the parts.

CAUTION

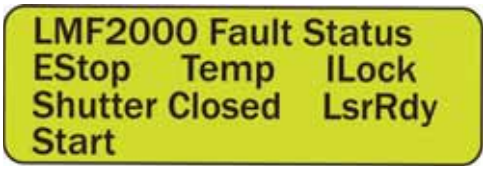
Do ***not*** attempt to remove the fiber at the rear of the marker head under any circumstances. Doing so will ***destroy*** the fiber and void the warranty. The Miyachi Unitek Corporation assumes no liability for such action, the fiber will have to be replaced at the customer's expense.

Section II: Troubleshooting

If the Marker develops any fault conditions, the type of trouble will display in the LCD Display Screen and the Fault Indicator LEDs will flash. Several fault messages may display at the same time as shown on the sample LCD screen below. If you see the LEDs flashing, check the message on the display screen and refer to the chart below for assistance.

NOTE: The shutter will close immediately on any fault, and a power supply will occur in most cases. This power supply fault is an induced fault and not the primary fault unless it occurs alone.

To reset from a fault condition, clear the cause of the fault and cycle the key switch or send an I/O signal to the Fault Reset I/O bit as detailed in *Appendix B*.



Error	Number of Flashes	LCD Screen Display	Cause/Corrective Measures
Emergency Stop	3	**EStop**	Emergency stop circuit is open. Verify the **E-Stop** connector on the rear panel is closed, the E-Stop switch on the front panel is not active and the laser head cover is securely installed.
Startup Fault System Timeout	7	**Start**	The system has reached the end of its timeout without the laser or control hardware reaching a ready state. Please ensure that no other faults exist and clear by cycling the key switch.
Shutter Fault	2	**Shutter**	A shutter fault has occurred. Please cycle the key switch to try and recover. If the fault continues please check the remote interlock circuit. Ensure that a start job signal is not being sent through External I/O or Remote API within 100ms of a shutter open event. If the trouble continues, contact Miyachi Unitek.
Laser Ready Fault	4	**LsrRdy**	A laser module fault has occurred. Contact Miyachi Unitek if the problem can not be cleared.
Laser Temperature	5	**Temp**	The laser temperature has exceeded the maximum allowable temperature for laser emission. Turn the machine off and allow the unit to cool down. If the laser temperature does not decrease after a period of time, the air filter may be clogged or airflow might otherwise be obstructed. Check that the ambient temperature is 94°F (35°C) or below. If the trouble continues, contact Miyachi Unitek. The laser module temperature is displayed on the front panel LCD screen for LMF Series "-HP" modules. The module temperature range varies between ambient and ambient +10 degrees C depending on the warmup state of the laser. Laser faults on non-HP models occur at 45C module temperatures, and 55C on –HP models.

Error	Number of Flashes	LCD Screen Display	Cause/Corrective Measures
Power Supply Fault	6	**PSup**	The internal power supplies are in a fault condition. This fault is typically caused by another fault state. Clear any existing faults and try again. If the power supply fault persists and no other faults are active, turn off the machine at the switch on the back of the unit. Wait 3 minutes and restart. If the problem returns contact Miyachi Unitek.
Remote Interlock Fault	8	**ILock**	The remote interlock was open when the laser was firing or instructed to fire. Close the remote interlock, remove the partially marked part if applicable, clear the fault, and try again.

Section III: Clean and Replace the Protective Glass

Each *f*-theta lens is equipped with an optically-coated protective glass. When marking materials over a period of time, many of the airborne particles produced during the marking process can dirty or fog the protective glass. ***It is important to keep the protective glass clean.*** If the protective glass is dirty and continually used, the contaminants may permanently damage the glass surface. When cleaning the surface, use only lens cleaning paper. A scratched glass surface will cause undesirable marking results.

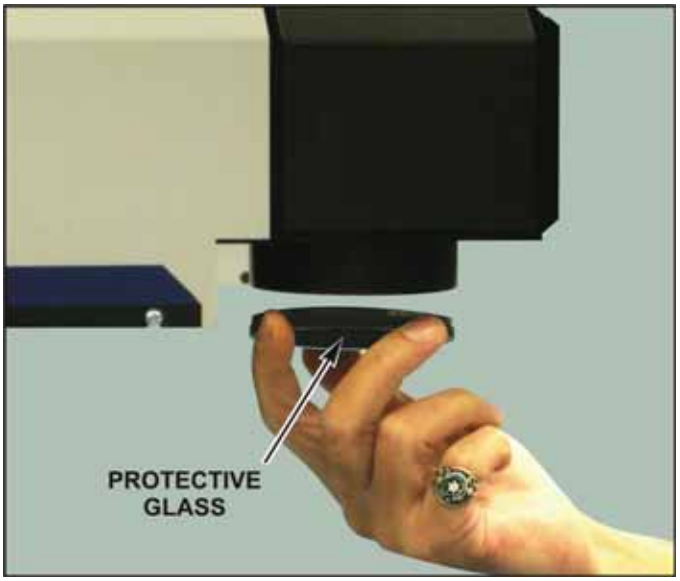
Required Items:

- Lens Cleaning Paper
- Acetone
- Air Blower
- Powder-free Vinyl Gloves or Finger Cots

1. Turn the power OFF.
2. Turn the protective glass holder CCW (counter-clockwise) to remove it. Take care ***not*** to drop it.

   NOTE: ***Before*** you clean the protective glass, put on a pair of powder-free vinyl gloves or finger cots.

   .



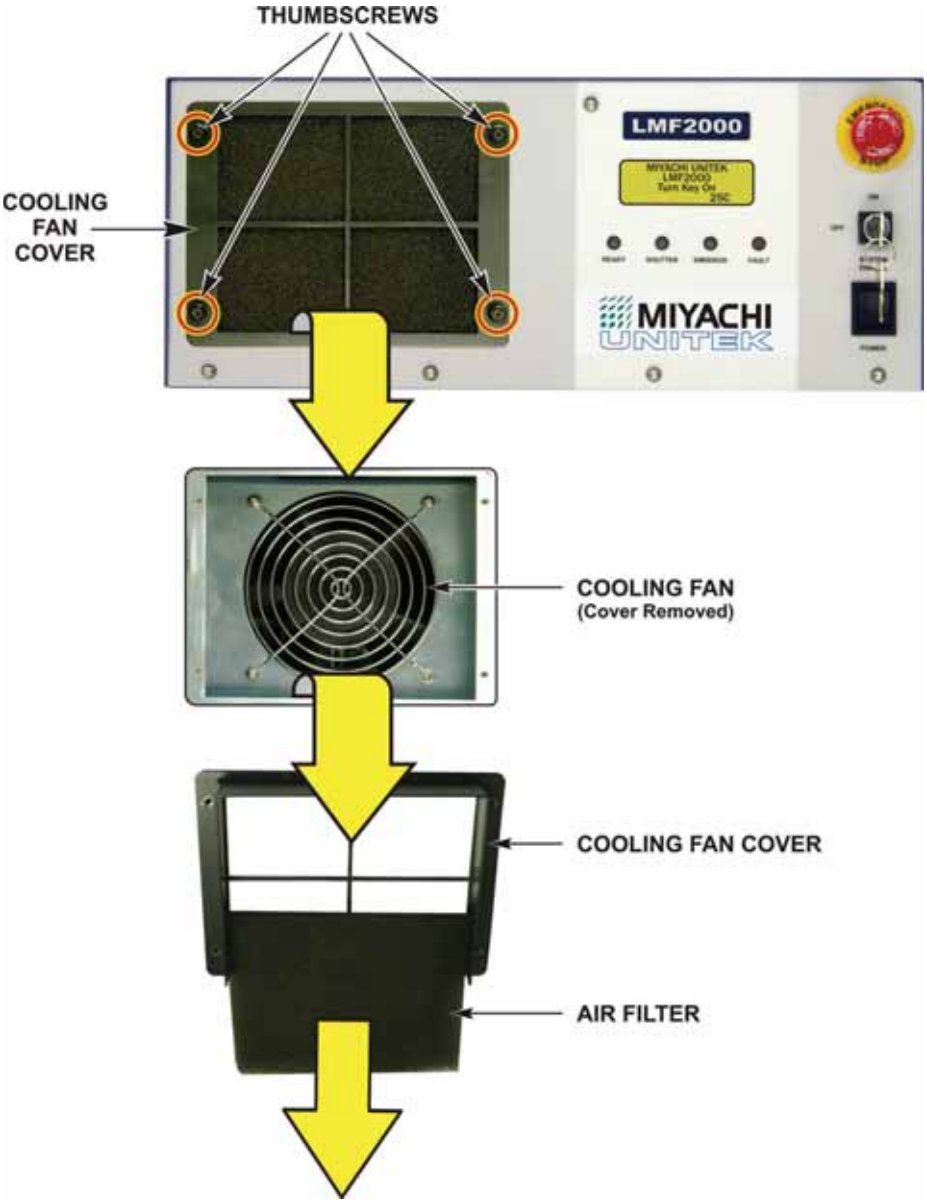
3. Add a few drops of acetone onto the lens and use the lens cleaning paper to wipe the protective glass. When cleaning, ***draw a spiral pattern from the center of the glass as shown***.

   NOTE: If you can ***not*** get the protective glass clean after several attempts, replace the protective glass with a new unit.
4. Install the protective glass back onto the Marker by turning it in the CW (clockwise) direction.

Section IV: Replace the Air Filter

NOTE: The air filters on the Marker need to be cleaned regularly to reduce the risk of overheating caused by restricted airflow into the device. Please change the filters every 1-6 months depending on the operating environment.

1. Verify that the Marker has been turned OFF.
2. Loosen the **THUMBSCREWS** on the **COOLING FAN COVER** and remove the cover.
3. Slide the **AIR FILTER** out of the **COOLING FAN COVER**.
4. Put a new or cleaned **AIR FILTER** into the **COOLING FAN COVER**.
5. Put the **COOLING FAN COVER** back on the font panel and tighten the **THUMBSCREWS**.



Section V: Firmware Update

WARNING

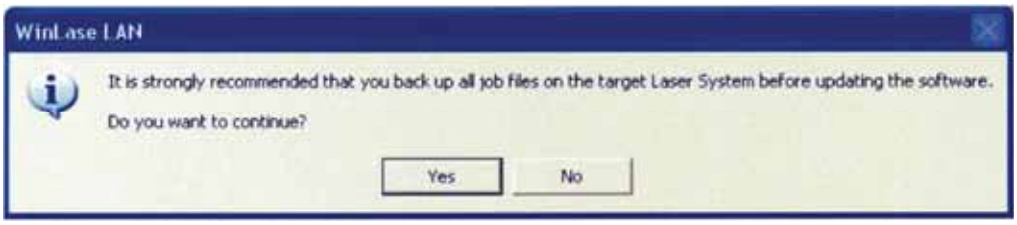
When updating the Controller Card firmware, ***all files stored locally on the Controller Card will be lost.*** Backup all job files that have been stored in the Controller Card's flash memory.

From time to time, *Lanmark Controls, Inc.* will release an updated version of the firmware that is resident on the Controller Card. This update may be in the form of a single file, or a web link. The procedure written below describes how to update the firmware in the Controller Card.

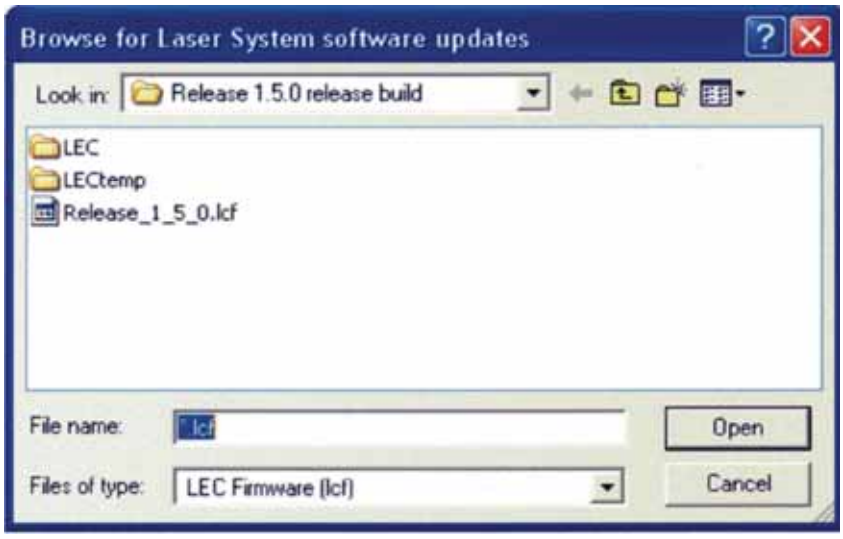
NOTE: After updating the firmware, the Controller Card board must automatically (or manually) be restarted for changes to take effect.

To install an update from a provided update file:

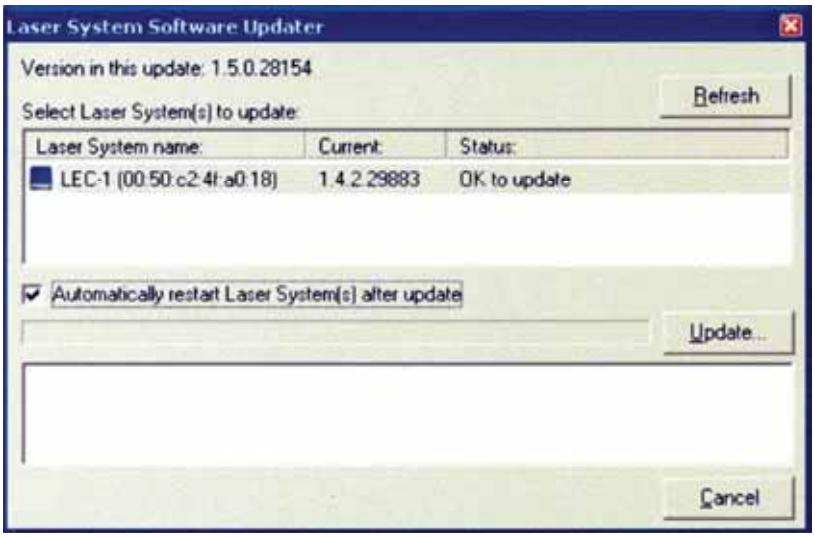
1. From the **Main menu** of *WinLase*, click **Help > Laser System device software update**. A back-up warning message will appear.



2. Click **Yes**, and the **Browse for Laser System software updates** dialog box appears.



3. Navigate to the Update file, select it and choose **Open**. The **Laser System Software Updater** will appear.



The settings listed in the **Laser System Software Updater** consist of the following:

- **Refresh** — Click this button to rescan the network for installed Controller Card.
- **Laser System name** — The name of a Controller Card board that has been detected on the network. The icon representing the Controller Card will also indicate its status.
- **Current** — The version of the firmware currently on the Controller Card.
- **Status** — Indicates whether the update file you have selected is newer, older, or the same version as the version currently on the Controller Card.
- **Automatically restart Laser System(s) after update** — On current firmware versions after 1.4.2, the Controller Card will automatically restart when the firmware update is complete. In order for changes to take affect, the LEC- 1 must be restarted after an update session. On LEC- 1 cards with current firmware previous to 1.4.2, this checkbox has no effect and the LEC- 1 must be restarted manually.
- **Update** — Used to update the firmware in the Controller Card.

4. Select the Controller Card board(s) that need updating, then select the **Update** button.

Section VI: Repair Service

If you have problems with your Laser Marker that you cannot resolve, please contact the Miyachi Unitek Corporation; see **CONTACT US** in the front of this guide to get in touch with us by e-mail, telephone, or regular mail.

CHAPTER 5
REMOTE INTERFACE

Section I: Using the Embedded Controller

Using the Embedded Controller Card

Because of the flexibility of the Controller Card's architecture, there are a number of different system integration possibilities available. All possible solutions rely on a combination of the following interfaces to control the Controller Card in a marking application**:**

- *WinLase LAN*™ GUI (Graphical User Interface)

  NOTE: For the rest of this chapter *WinLase LAN*™ and *WinLase Professional*™ will simply be referred to as ***WinLase***.
- Remote Command API (Application Programming Interface), for interaction with a PC or PLC through TCP/IP (Client/Server) and RS-232.
- Pendant Controller, for local interaction (option).

Section II: Using the WinLase LAN GUI

The *WinLase*™ LAN Software includes the following:

- *WinLase*™ GUI (Graphical User Interface).
- Lanmark Controls Inc. COM server object interface.
- Full documentation on the features of the included software tools.

GUI (Graphical User Interface) Features

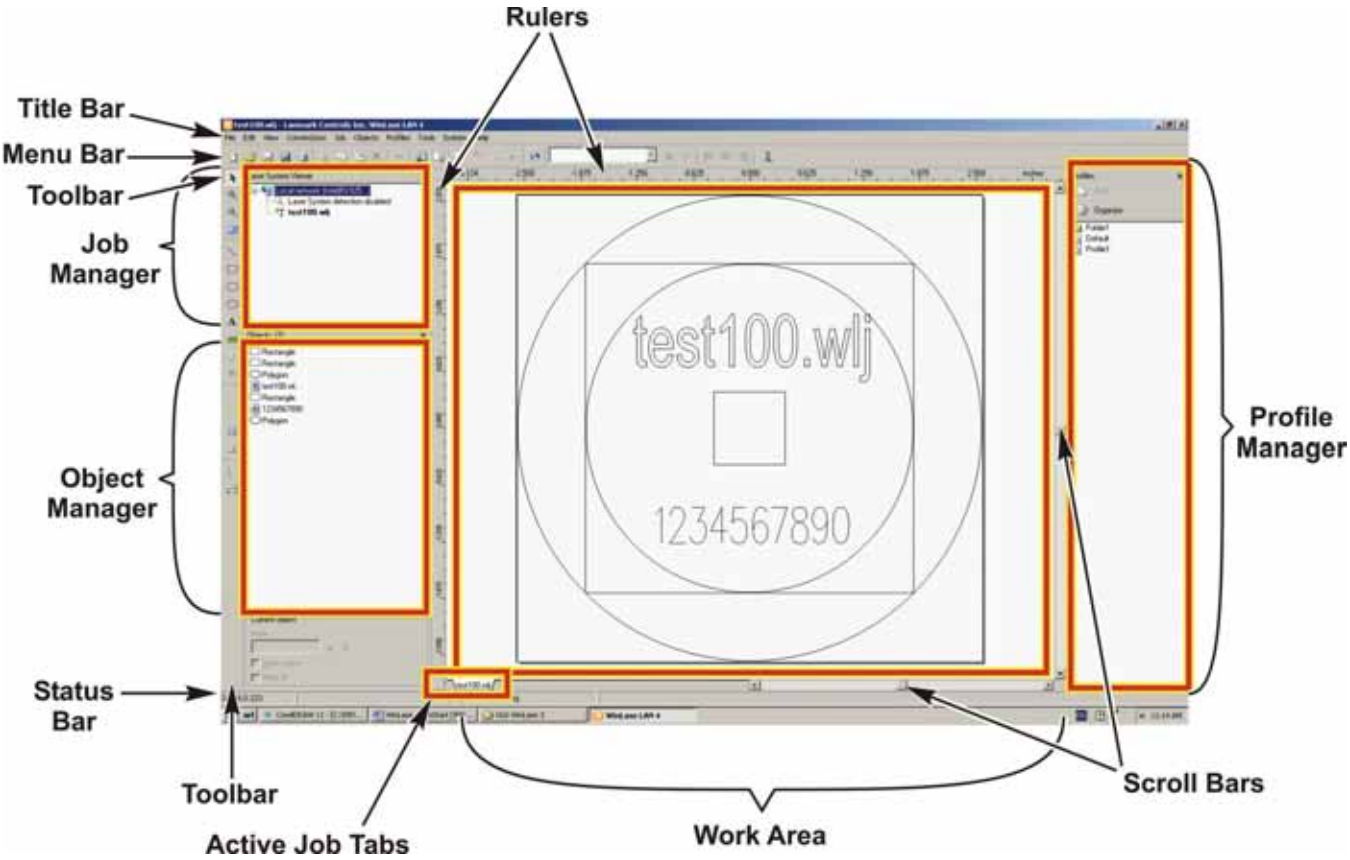
WinLase elements include**:**

- Line-art graphics: CAD, line-drawings, logos.
- Shaded graphics: photos, halftones & grayscale artwork.
- TrueType™ fonts, filled or outline-only.
- Single point or drill object arrays.
- AutoDate™, TextMerge™, Serialization, and Barcode.
- Automation: I/O control, 4-axis motor control, time delays, and custom operator messages.

WinLase facilitates the creation, editing, control, execution, and automation of all laser-marking tasks. The *WinLase*™ Software Suite includes the following features**:**

- Password-protected security lockout -- operators can be limited to only selected and running Jobs.
- Multiple open Jobs -- quickly switch between marking tasks with the click of a mouse.
- Background template -- place an image of the parts tooling in the background to aid in mark placement.
- *Runtime Fill*™ change fill levels "on the fly" to develop the optimum in image quality for each object type and material.
- Full support for lasers with visible pointers for real time positioning of the mark.
- HPGL (*.plt), WMF (*.wmf) EMF (*.emf), DXF (*.dxf), EPS (*.eps), JPEG (*.jpg), GIF (*.gif), PCX (*.pcx), and Windows Bitmap (*.bmp) graphic filters.
- Internal capability to generate linear and radial text, barcodes, AutoDate™, serialization, and hole-drilling.
- Complete TrueType™ font support -- engrave any TrueType™ font installed on the system.
- Scale, move, rotate, group, or reverse any object on the screen.
- Precise numerical control of laser-operation parameters.
- Automation scripts easily built via mouse clicks.
- Programmable alerts, warning, and run-time operator input of job numbers, batch numbers, etc.
- Built-in 4-axis motor control -- use rotary tables, linear motion, and/or X-Y tables.
- Text marking on cylindrical parts using an optional rotary indexer.

WinLase™ contains all of the elements of a multi-element **Job Editor**, automation sequencing tool (simplified ladder logic), and password-protected Operator's Interface. Most procedures can be efficiently served entirely from within the program.



COM Automation Server API (Application Programming Interface)

WinLase™ exposes a COM (Computer Object Module) Automation server, which offers external programs the ability to communicate with and control *WinLase*™. For detailed information on the COM Automation server interface commands, please refer to *Appendix D* of this manual.

Please refer to the *WinLase Reference Manual* for complete details on using the *WinLase*™ software package.

Section III: Using the Remote Command API

Remote Command API (Application Programming Interface)

The Embedded Controller Card was designed to be a powerful standalone controller, with the ability to accept commands and return responses. The Remote Command API provides extended functionality to load jobs, rename jobs, change administration settings, and many other functions.

There are three methods available for interfacing with the Remote Command API while in **Local** mode:

- Message based TCP/IP socket connection
- Message based RS-232 connection
- Message based wireless Bluetooth connection (future)

All interfaces are active simultaneously for interacting with the Remote Command API. All interfaces support making calls to get parameters. Some commands, however require the client to "Take Control" of the Controller Card Host device. When a client has control, the client can send execution commands as well as commands to set the laser parameters.

Using the API (Application Programming Interface)

The Controller Card Remote Command API uses a message based communication protocol. The client and the server (Host) must cooperate by sending messages back and forth in an alternating fashion. Messages sent to the Controller Card are text strings, and must end with a line feed. Messages received from the Controller Card are text strings and end in a line feed.

The **Remote Command API** provides a rich set of commands for communication and control of an Controller Card. The **API** allows you to get and set system parameters, as well as perform actions with locally stored job files, and to control the behavior of individual objects within the job files. If you are loading and controlling the execution of locally stored jobs, use the **TakeListControl** command to gain exclusive access to the server. Use the **GetFlashJobFileList** or **GetUSBJobFileList** command to discover the locally stored job files. Use the **LoadJobFromFlash** or **LoadJobFromUSB** command to load a job into memory. A job previously loaded into memory can be executed by using the **ExecuteJobOnce** or **ExecuteJobContinuous** command. When you are finished sending and receiving commands, use the **ReleaseHostControl** command to allow other hosts exclusive access to the device.

To read the status of server parameters, use any of the **Get** commands. Most of these commands do not require the client to have exclusive access to the server (Host).

To set server parameters, use the **FakeHostControl** command to gain exclusive access to the server. Use any of the **Set** commands to make changes to server parameters. When you are finished setting all parameters, use the **ReleaseHostControl** command to allow other hosts access to the API. If any IP settings have been changed, these settings will not take effect until the board goes through a power cycle, or you execute the **HardwareReset** command.

API (Application Programming Interface) Command Set

The interface provided by the **Remote Command API** is a message (character string) based protocol. All command and response strings must be terminated by a line feed. The **Remote Interface** commands and their intended use are listed in *Appendix D, Remote Interface Commands* and are presented in alphabetical order. Commands with multiple parameters are sent and returned in a comma delimited format.

NOTE: All commands are text strings and are expressed in the table enclosed in quotes(”). The quotation characters are ***not*** part of the command. This is also true for responses. Commands and arguments are case-sensitive.

When using the LAN host interface, all commands are available as either descriptive commands as in **SetLocalIP** or their numerical equivalent, as in **506**. When using the high speed RS-232 host interface, only the numerical version of the command is available.

NOTE: Not all commands are available on all host interfaces. All responses are sent as their numerical representations for all host interface types.

The following convention will be used when indicating additional parameters:

For example:

SetLocalIP,ipaddress is sent to the API as “SetLocalIP,192.168.42.1” (without exclamation marks).

A description of the command parameters follows each command.

Remote Command API List

For the complete list of commands, see *Appendix D, Remote Interface Commands*.

Section IV: Using the TCP/IP and RS-232 Remote Interface

In terms of simplicity, the RS-232 port is easier to set up than the TCP/IP interface, because the cable connection between the two computers is direct and troubleshooting is much easier because HyperTerminal can be used to troubleshoot the connection.

TCP/IP, on the other hand, is a Client/Server protocol that is a bit more difficult to set-up. TCP/IP is a networking protocol that has the ability to communicate over local area networks (LANs), wide area networks (WANs), and the Internet. In order for one program to communicate with another on a different computer, the remote device needs to make itself available on a specified Port, and the IP address of the remote device has to be known. It is this IP address and Port that ensures any message sent reaches the proper destination. In other words, in order for you to communicate over TCP/IP, you must know the IP address and Port of the device you are communicating with.

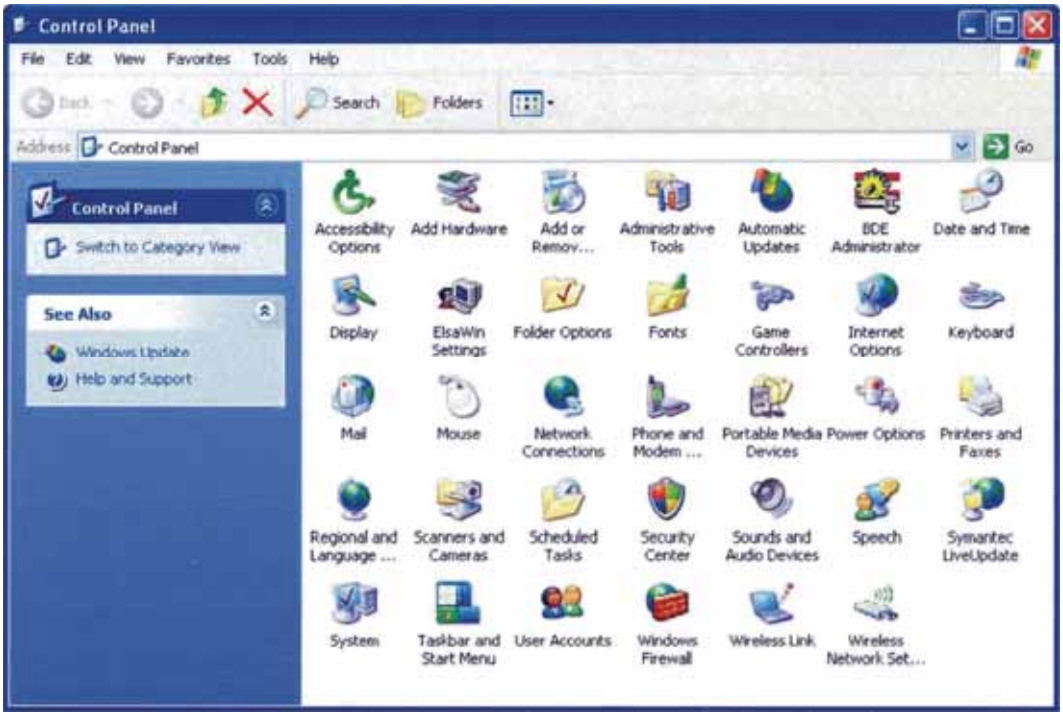
By default, *WinLase* makes the Remote Interface service available on Port *350*. This can be changed in the **Host Interface** setup box if Port 350 conflicts with another installed port on your machine. Determining the IP address of the computer that *WinLase* is running on depends on how the TCP/IP stack is configured in *Windows*. Computers can either have a static IP address, or one that is dynamically allocated when the computer signs on to the network.

By default, the Controller Card makes the Remote Command API Interface service available on Port 12500. The Controller Card can be configured to use a Static IP address, or to request an IP address each time it starts from a DHCP server. It is recommended in situations where the LAN based Remote Command API is used, to configure Static IP addressing for the Controller Card. Using this approach, the Remote Command API client will know the IP address of the Controller Card without having to use other discovery techniques. Remote control of the Controller Card can be established by any client computer that supports TCP/IP networking. This includes computers running *Microsoft Windows*™ , *Linux*™, or other operating systems. Communication with the board is established by opening a socket connection using the IP address on port 12500.

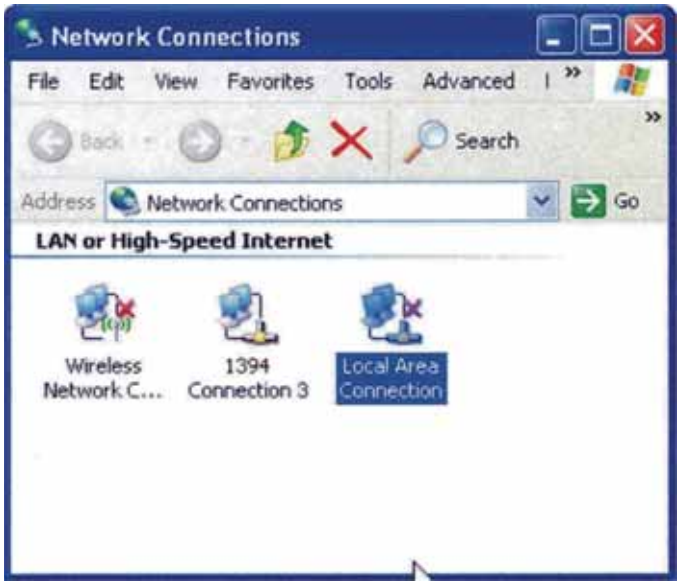
To determine how TCP/IP is configured on your *Windows XP*™ computer:

1. In *Windows XP*™, click **Start > Settings > Control Panel** to display the **Control Panel**.

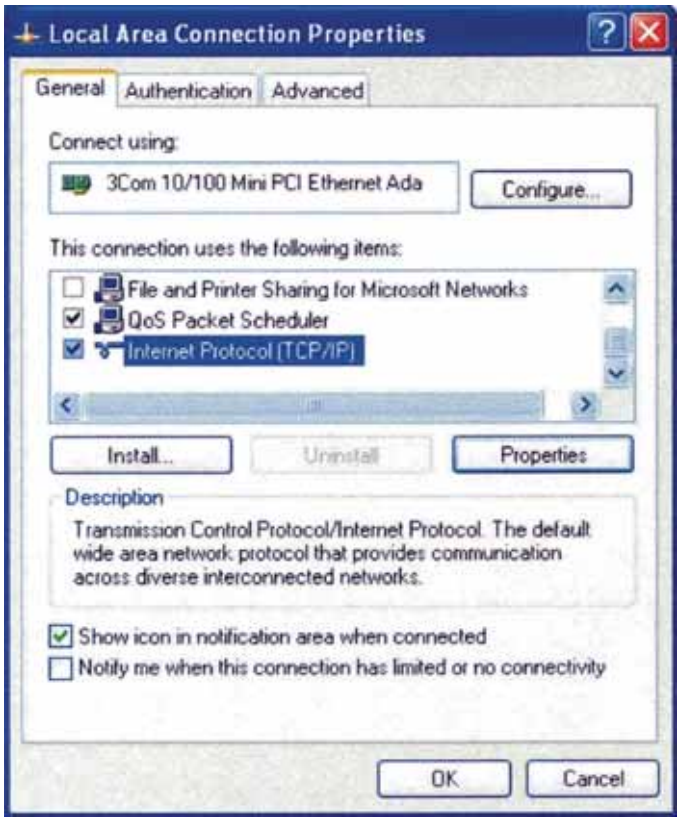
2. Click on the **Network and Internet Connections** icon.
3. When the **Network and Internet Connections** screen displays, double click the **Network Connections** icon.



4. The **Network Connections** window will display. Click on the **Local Area Connection** icon.

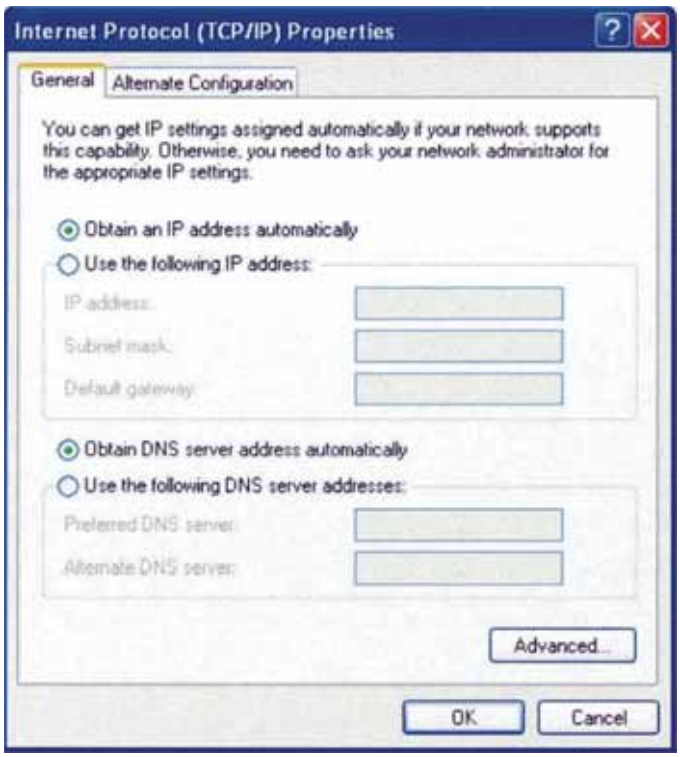


5. Using your right mouse button, select **Properties** to view the LAN Properties. The **Connection Properties** window will appear.
6. **Internet Protocol (TCP/IP)** must appear and be checked in the list of items. If not, install TCP/IP, referring to the *Windows XP*™ documentation as a guide.
7. Highlight **Internet Protocol (TCP/IP)**, and then click the **Properties** button.



8. If the **Use the following IP address:** radio button is selected, your computer uses a static IP address, which is shown in the **IP Address** box.
9. If the **Obtain an IP address automatically** radio button is selected, your computer is assigned a different IP address (dynamic address) each time it boots up on the network. If your computer uses a static IP address, you can use this address when communicating with *WinLase* over the TCP/IP port.

   If your computer uses a dynamically assigned address, another method has to be used to obtain the address.



RS-232 and TCP/IP Commands and Functions

The interface provided for RS-232 and TCP/IP is textual; commands are sent over either port as ASCII text strings. These strings are interpreted by *WinLase*, and are executed accordingly.

NOTE: The Marker also supports the *Fieldbus* interface protocol. Because *Fieldbus* is fundamentally different from the RS-232 and TCP/IP Remote Interface, it is not discussed in this manual. Please refer to the appropriate *Fieldbus* documentation for more information.

RS-232 and TCP/IP Command List

For the complete list of commands, see *Appendix D, Remote Commands*.

Section V: Using the Remote Pendant

Using the Remote Pendant

A Remote Pendant can be used to control the Laser Marker. This is an option available through Miyachi Unitek. Please contact the Miyachi Unitek Corporation; see **CONTACT US** in the front of this guide to get in touch with us by e-mail, telephone, or regular mail.

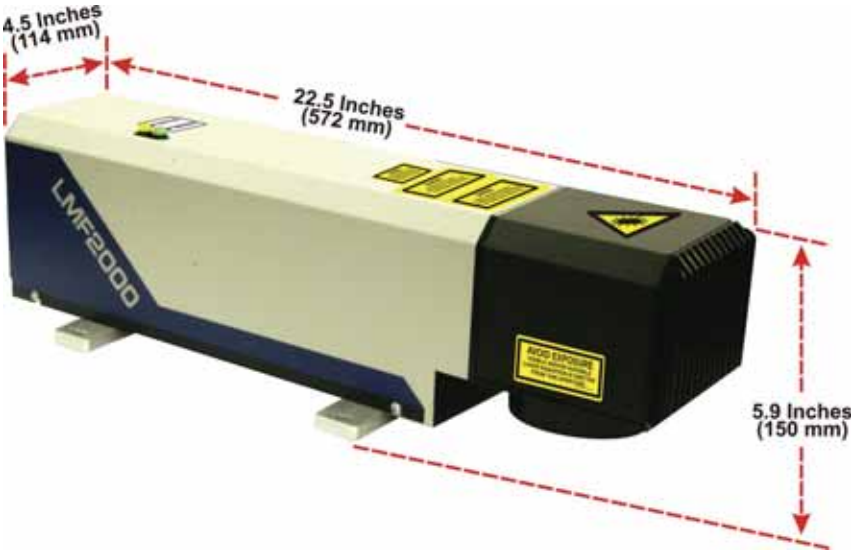
Pendant Error Codes

The following error codes may be displayed when the pendant is executing jobs locally.

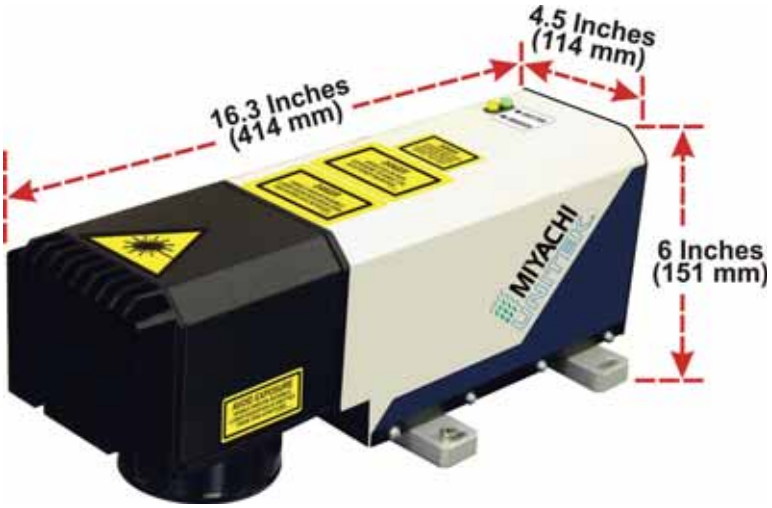
LoadFail	=	100
NoObjects	=	101
NoProperties	=	102
WriteFail	=	103
FileFormat	=	104
FileException	=	105
UnknownObject	=	106
UnknownType	=	107
NotSupported	=	108

APPENDIX A
TECHNICAL SPECIFICATIONS

Section I: Laser Specifications

PARAMETER	SPECIFICATIONS
Control Unit Dimensions: **Control Unit Weight:** **61 lbs (27.7 kg)**	
Standard Head Dimensions: **Weight:** **18 lbs (8.2 kg)**	
Compact Head Dimensions: **≤20W Output** **Weight:** **16 lbs (7.3 kg)**	

APPENDIX A: TECHNICAL SPECIFICATIONS

PARAMETER	SPECIFICATIONS
Compact Head Dimensions: **>20W Output** **Weight:** **17.2 lbs (7.8 kg)**	
Environmental **Enhanced Environment Head** Ambient Temperature: Relative Humidity: Installation Site:	 IP-54 rated 41-95F (5-35ºC) Less than 90% (non-condensing) Do not use where there is considerable dirt, dust, oil mist, chemicals, fumes, moisture, vibration or near a high frequency noise source.
Electrical Requirements Power Supply: Maximum Running Current: Recommended AC Service:	 90-130, 180-260 VAC ± 10% 50/60 Hz, single-phase 7A @110VAC 10A @110VAC

APPENDIX A: TECHNICAL SPECIFICATIONS

PARAMETER	SPECIFICATIONS
Laser Specifications	
LMF1000	
Oscillation Wavelength:	1060-1150, central emission at 1064 ±5 nm
Oscillation Mode:	TEM00-Mode (at or near)
M^2	≤2.0
Minimum Continuous Laser Power from Engine:	>10W @ 20-100kHz, Power output available from 2-100kHz
Minimum Continuous Output Power at Aperture:	>8.9W @ 25-100kHz, Power output available from 2-100kHz
Type of Oscillation:	Pulsed oscillation 2-100kHz
Pulse Frequency:	
Number of Unique Pulse Tune Waveforms:	1
Guide Beam Wavelength:	650 nm, <1mW at output
Laser Specifications	
LMF2000	
Oscillation Wavelength:	1060-1150, central emission at 1064 ±5 nm
Oscillation Mode:	TEM00-Mode (at or near)
M^2	≤2.0
Minimum Continuous Laser Power from Engine:	>18.8W @ 25-100kHz, Power output available from 2-100kHz
Minimum Continuous Output Power at Aperture:	>16.8W @ 25-100kHz, Power output available from 2-100kHz
Type of Oscillation:	Pulsed oscillation 2-100kHz
Pulse Frequency:	
Number of Unique Pulse Tune Waveforms:	1
Guide Beam Wavelength:	650 nm, <1mW at output

PARAMETER	SPECIFICATIONS
Laser Specifications	
LMF2000-HP	
Oscillation Wavelength:	1060-1150, central emission at 1064 ±5 nm
Oscillation Mode:	TEM00-Mode (at or near)
M^2	≤2.0
Minimum Continuous Laser Power from Engine:	>20.0W @ 25-500kHz, Power output available from 2-500kHz
Minimum Continuous Output Power at Aperture:	>17.9W @ 25-500kHz, Power output available from 2-500kHz
Type of Oscillation:	CW or Pulsed oscillation 2-500kHz
Pulse Frequency:	
Number of Unique Pulse Tune Waveforms:	24
Guide Beam Wavelength:	650 nm, <1mW at output
Laser Specifications	
LMF2000-SM	
Oscillation Wavelength:	1060-1150, central emission at 1064 ±5 nm
Oscillation Mode:	TEM00-Mode (at or near)
M^2	≤1.3
Minimum Continuous Laser Power from Engine:	>20.0W @ 40-500kHz, Power output available from 2-500kHz
Minimum Continuous Output Power at Aperture:	>17.9W @ 40-500kHz, Power output available from 2-500kHz
Type of Oscillation:	CW or Pulsed oscillation 2-500kHz
Pulse Frequency:	
Number of Unique Pulse Tune Waveforms:	24
Guide Beam Wavelength:	650 nm, <1mW at output

APPENDIX A: TECHNICAL SPECIFICATIONS

PARAMETER	SPECIFICATIONS
Laser Specifications	
LMF3500-HP	
Oscillation Wavelength:	1060-1150, central emission at 1064 ±5 nm
Oscillation Mode:	TEM00-Mode (at or near)
M^2	≤3.7
Minimum Continuous Laser Power from Engine:	>38.0W @ 30-500kHz, Power output available from 2-500kHz
Minimum Continuous Output Power at Aperture:	>34.2W @ 30-500kHz, Power output available from 2-500kHz
Type of Oscillation:	CW or Pulsed oscillation 2-500kHz
Pulse Frequency:	
Number of Unique Pulse Tune Waveforms:	18
Guide Beam Wavelength:	650 nm, <1mW at output
Software	See the **separate** Software Manuals.

Marking Area

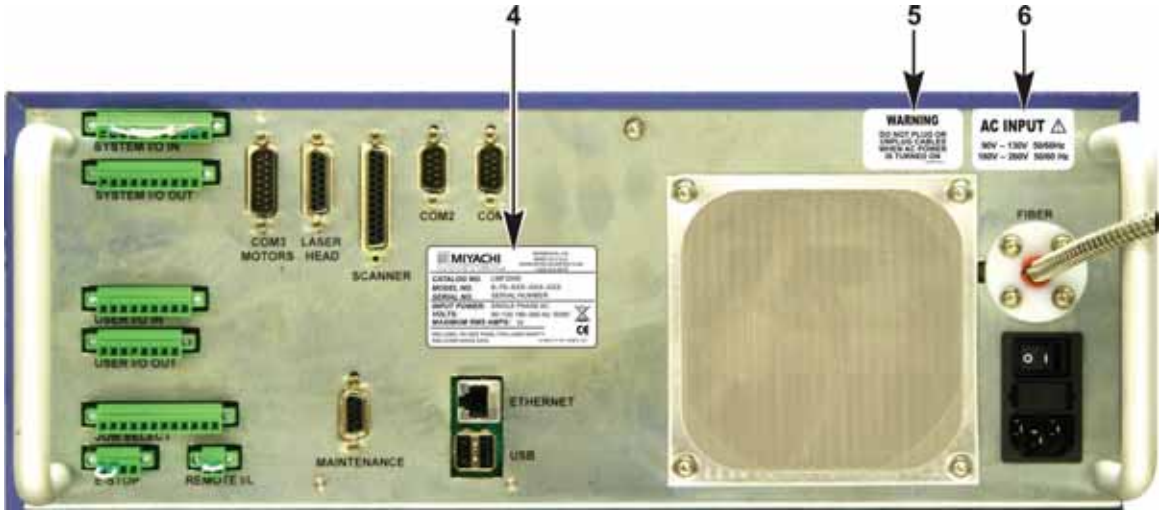
LMF Lens Marking Area			
f θ Lens Unit	***f* = 150mm**	***f* = 160mm**	***f* = 163mm (Obsolete)**
Scanning Method	Galvanometer Scanner		
λ (wavelength)	1060-1150, central emission 1064 ± 5 nm		
Marking Area	3.34 in. × 3.34 in. (84.8mm × 84.8mm)	3.89 in. × 3.89 in. (98.9mm × 98.9mm)	4.52 in. × 4.52 in. (114.8mm × 114.8mm)
Working Distance (approximate)	6.69 ± 0.08 in. (170 ± 2mm)	6.93 ± 0.08 in. (176 ± 2mm)	7.28 ± 0.08 in. (185 ± 2mm)
Position Resolution	0.00024 in. (6µm)	0.00028 in. (7µm)	0.00028 in. (7µm)

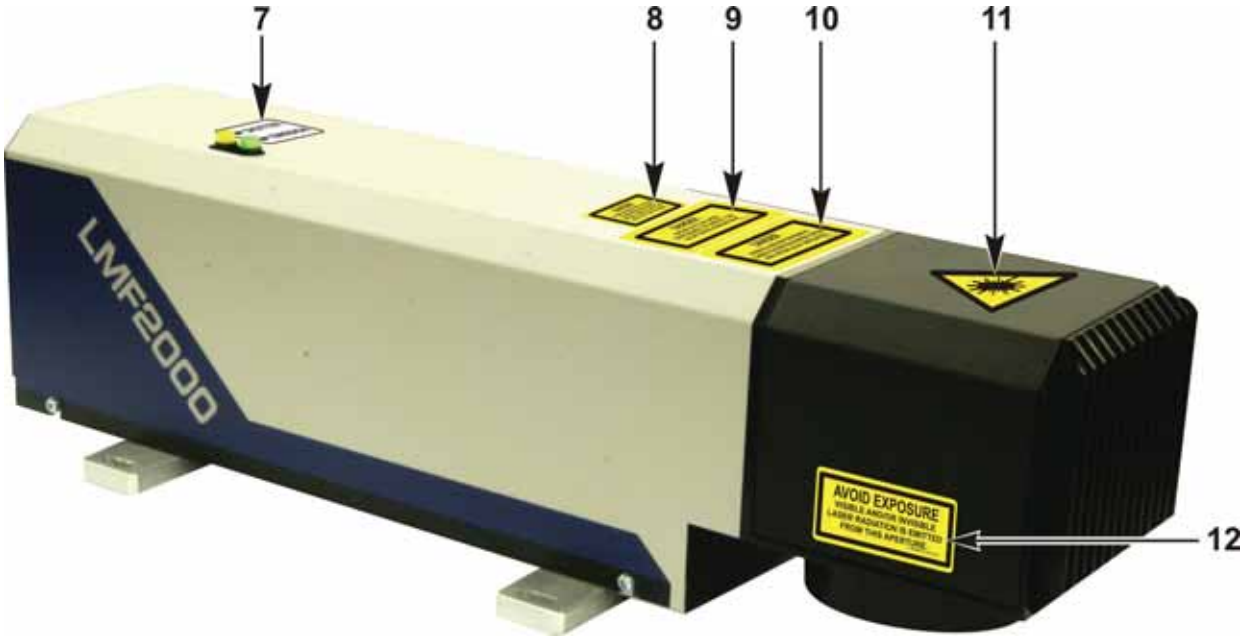
LMF Lens Marking Area		
f θ Lens Unit	***f* = 254mm**	***f* = 420mm**
Scanning Method	Galvanometer Scanner	
λ (wavelength)	1060-1150, central emission 1064 ± 5 nm	
Marking Area	6.18 in. × 6.18 in. (157 mm × 157mm)	11.44 in. × 11.44 in. (290.6mm × 290.6mm)
Working Distance (approximate)	11.65 ± 0.12 in. (296 ± 3 mm)	19.45 ± 0.20 in (494 ± 5 mm)
Position Resolution	0.00044 in. (11µm)	0.00088 in. (22µm)

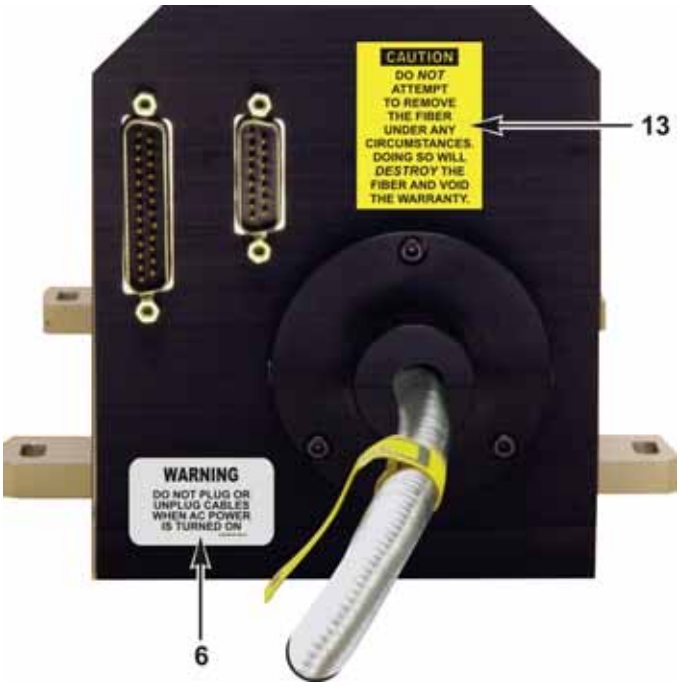
LMF Lens Marking Area		
f θ Lens Unit	***f* = 100mm** (≤20W Output Power)	***f* = 100mm** (>20W Output Power)
Scanning Method	Galvanometer Scanner	
λ (wavelength)	1060-1150, central emission 1064 ± 5 nm	
Marking Area	2.42 in. x 2.42in. (61.5mm x 61.5mm)	2.05 in. × 2.05 in. (52 mm × 52mm)
Working Distance (approximate)	3.86 ± 0.04 in. (98 ± 1mm)	
Position Resolution	0.00016 in. (4µm)	

Section II. Warning and Identification Labels

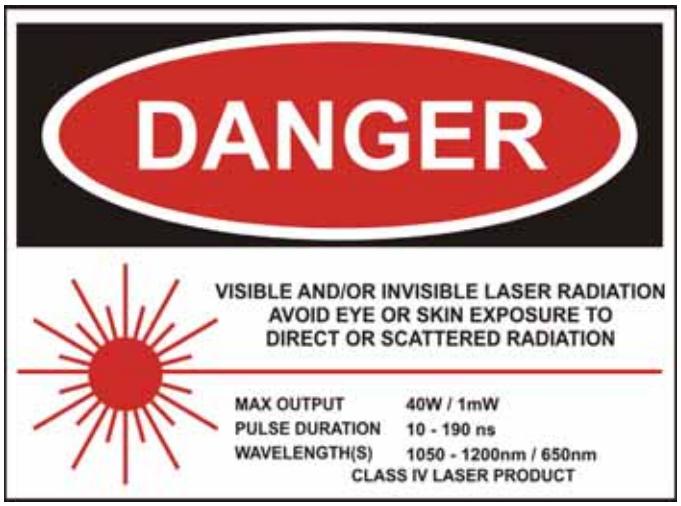
Location on Equipment



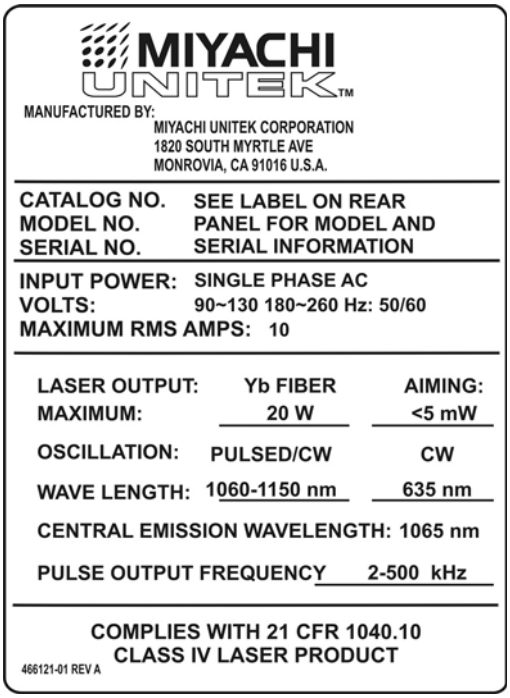




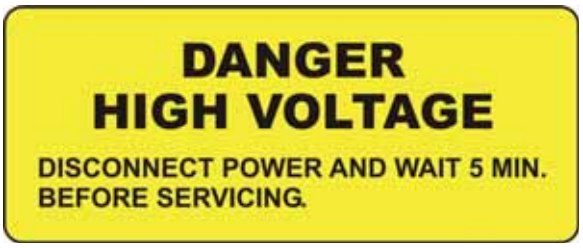
(Standard Head Shown Here)


DANGER
VISIBLE AND/OR INVISIBLE LASER RADIATION
AVOID EYE OR SKIN EXPOSURE TO
DIRECT OR SCATTERED RADIATION
MAX OUTPUT 40W / 1mW
PULSE DURATION 10 - 190 ns
WAVELENGTH(S) 1050 - 1200nm / 650nm
CLASS IV LASER PRODUCT

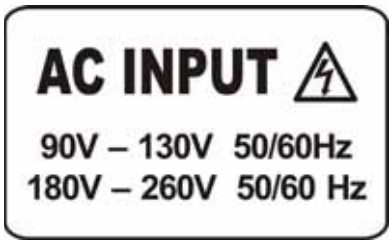
1


MIYACHI UNITEK™
MANUFACTURED BY:
MIYACHI UNITEK CORPORATION
1820 SOUTH MYRTLE AVE
MONROVIA, CA 91016 U.S.A.
CATALOG NO. SEE LABEL ON REAR
MODEL NO. PANEL FOR MODEL AND
SERIAL NO. SERIAL INFORMATION
INPUT POWER: SINGLE PHASE AC
VOLTS: 90~130 180~260 Hz: 50/60
MAXIMUM RMS AMPS: 10
LASER OUTPUT: Yb FIBER AIMING:
MAXIMUM: 20 W <5 mW
OSCILLATION: PULSED/CW CW
WAVE LENGTH: 1060-1150 nm 635 nm
CENTRAL EMISSION WAVELENGTH: 1065 nm
PULSE OUTPUT FREQUENCY 2-500 kHz
COMPLIES WITH 21 CFR 1040.10
CLASS IV LASER PRODUCT
466121-01 REV A

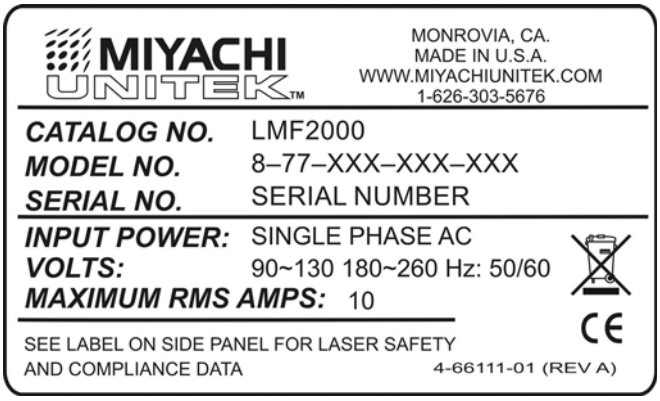
2


DANGER
HIGH VOLTAGE
DISCONNECT POWER AND WAIT 5 MIN.
BEFORE SERVICING.

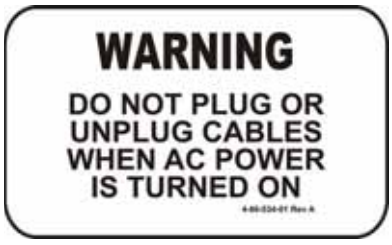
3


AC INPUT
90V – 130V 50/60Hz
180V – 260V 50/60 Hz

4


MIYACHI UNITEK™
MONROVIA, CA.
MADE IN U.S.A.
WWW.MIYACHIUNITEK.COM
1-626-303-5676
CATALOG NO. LMF2000
MODEL NO. 8–77–XXX–XXX–XXX
SERIAL NO. SERIAL NUMBER
INPUT POWER: SINGLE PHASE AC
VOLTS: 90~130 180~260 Hz: 50/60
MAXIMUM RMS AMPS: 10
CE
SEE LABEL ON SIDE PANEL FOR LASER SAFETY
AND COMPLIANCE DATA
4-66111-01 (REV A)

5


WARNING
DO NOT PLUG OR
UNPLUG CABLES
WHEN AC POWER
IS TURNED ON

6

7

8

9

10

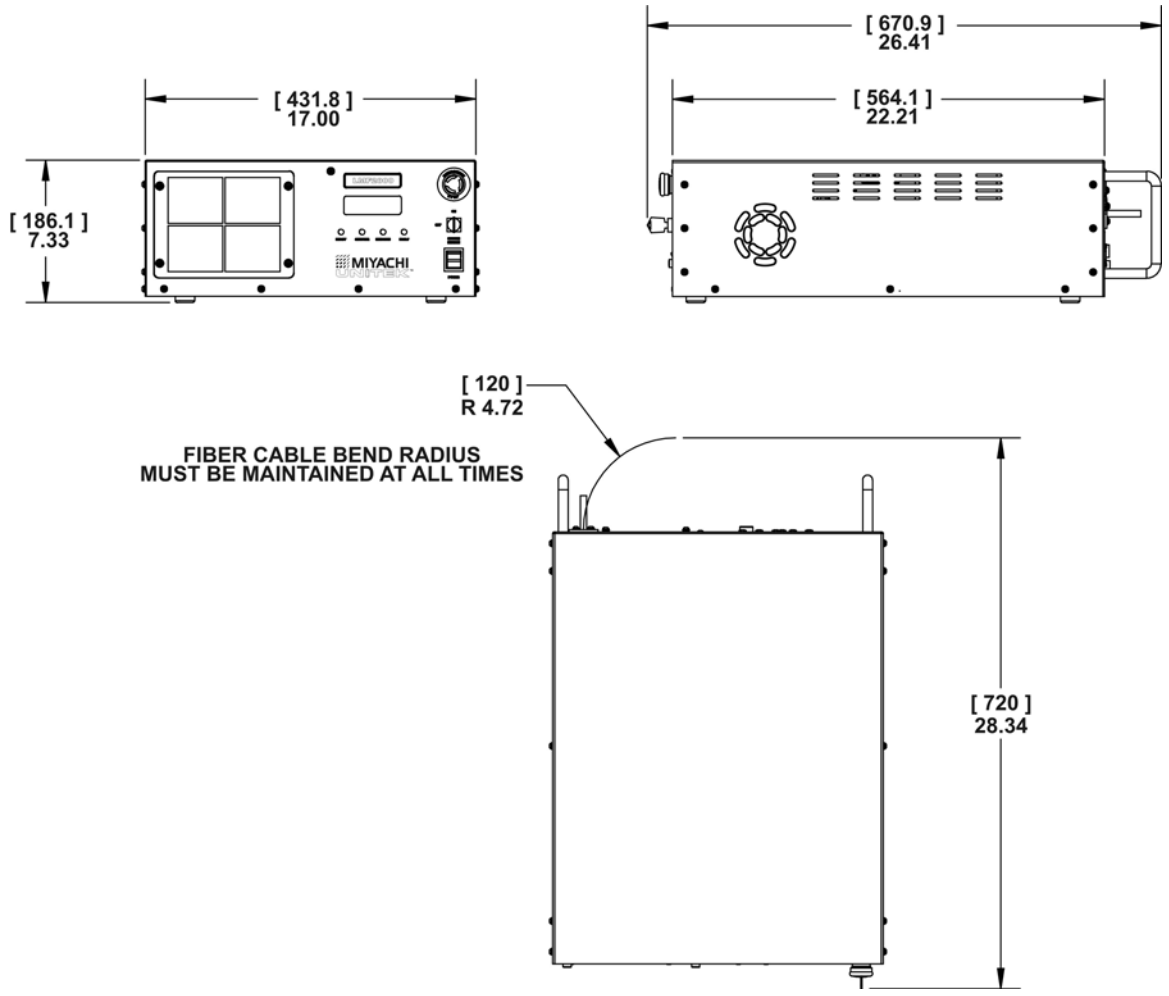
11

12

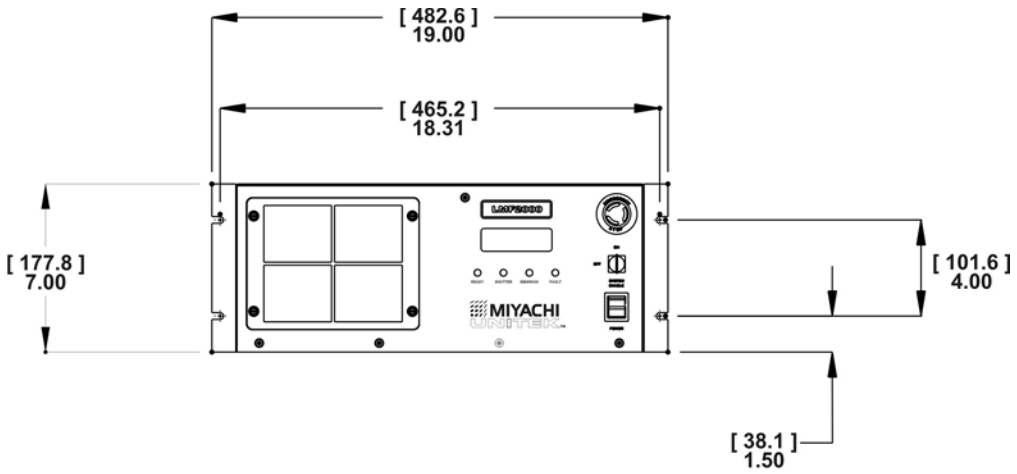
13

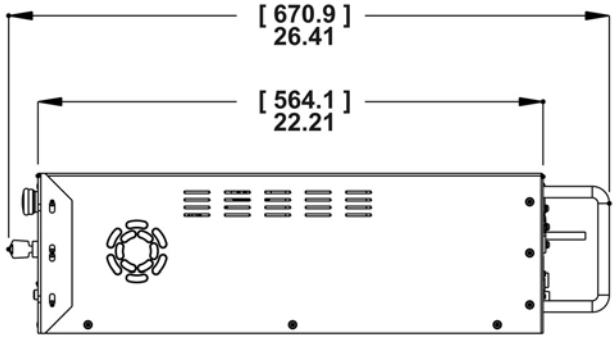
Section III. Engineering Drawings

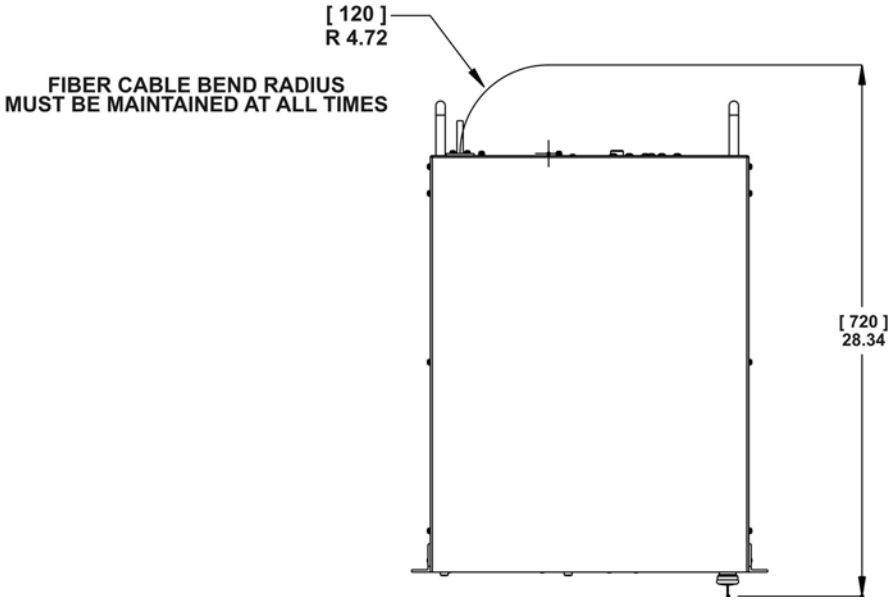
Control Unit (Benchtop Configuration)



Control Unit (Rack Configuration)



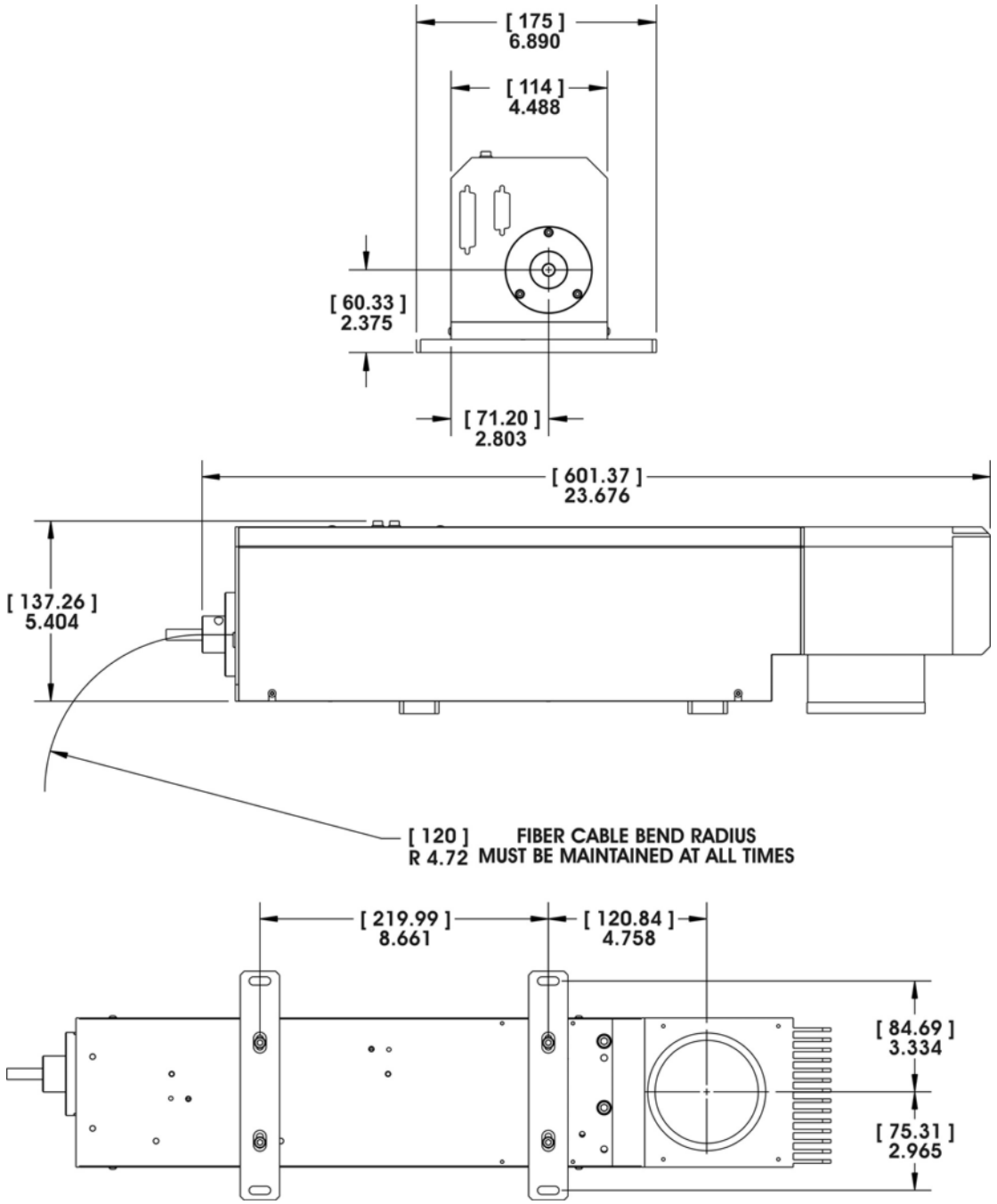




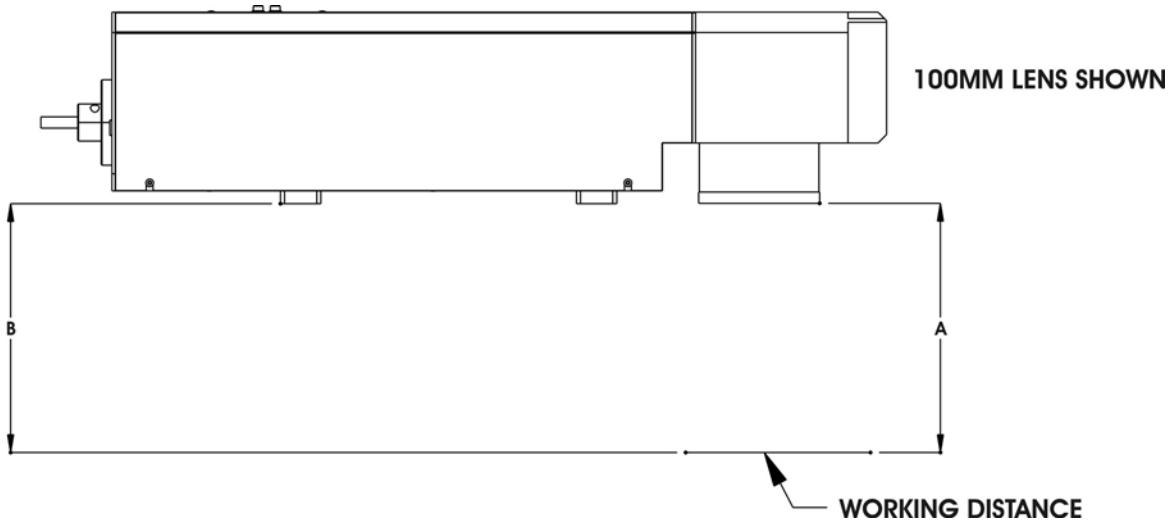
Standard Head

CAUTION

Do ***not*** attempt to remove the fiber at the rear of the marker head under any circumstances. Doing so will ***destroy*** the fiber and void the warranty. The Miyachi Unitek Corporation assumes no liability for such action, the fiber will have to be replaced at the customer's expense.



Standard Head Working Distance

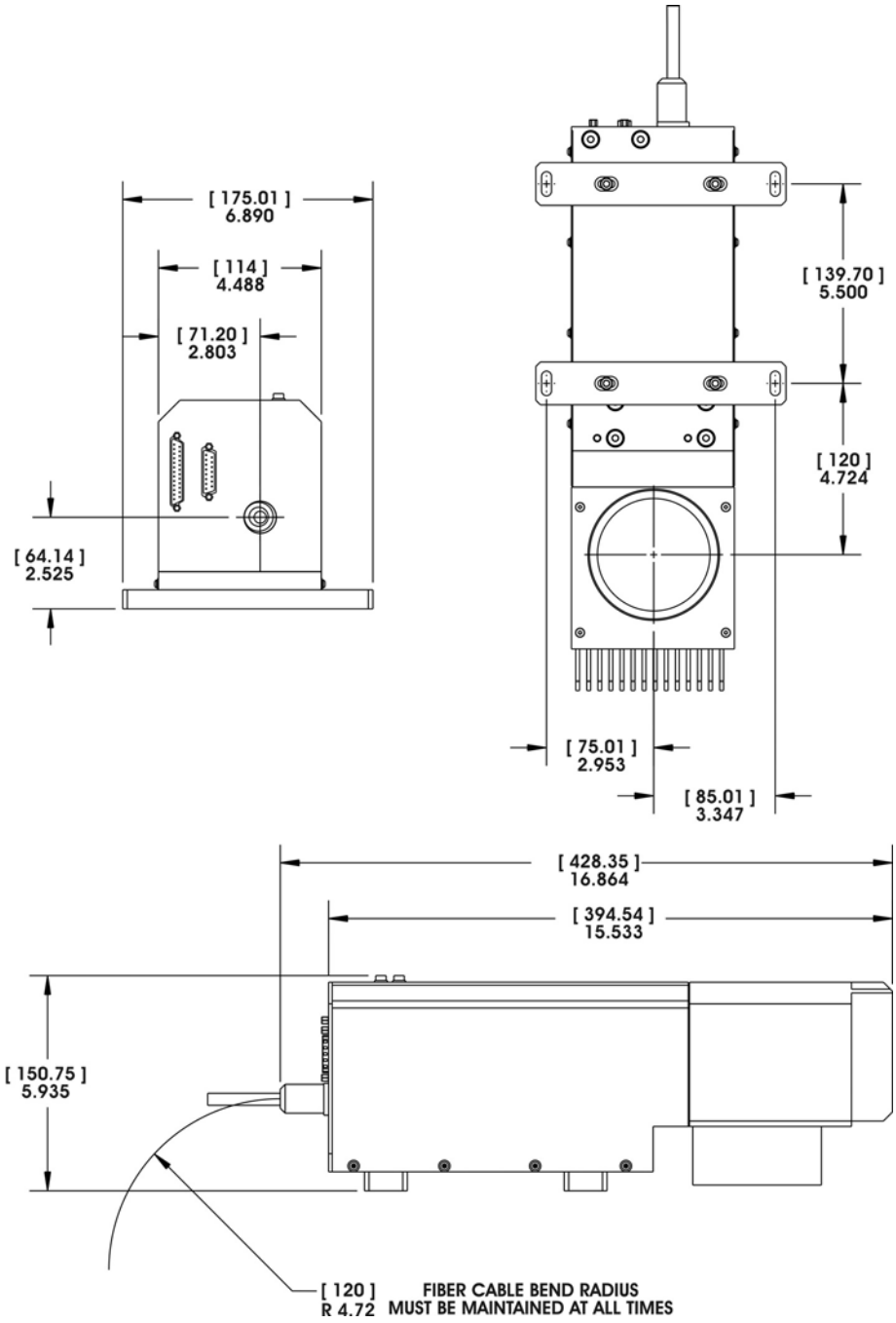


f-Theta Lens						
Dimension	**100mm**	**150mm**	**160mm**	**163mm (Obsolete)**	**254mm**	**420mm**
A	3.86” ±.04” (98mm ±1mm)	6.69” ±.08” (170mm ±2mm)	6.93” ±.08” (176mm ±2mm)	7.28” ±.08” (185mm ±2mm)	11.65” ±.12” (296mm ±3mm)	19.45” ±.20” (494mm ±5mm)
B	3.84” ±.04” (97.5mm ±1mm)	6.97” ±.08” (177mm ±2mm)	6.91” ± .08” (175mm ±2mm)	7.91” ±.08” (200.8mm ±2mm)	12.72” ±.12” (323.2mm ±3mm)	20.52” ±.20” (521.2mm ±5mm)

Compact Head ≤ 20W

CAUTION

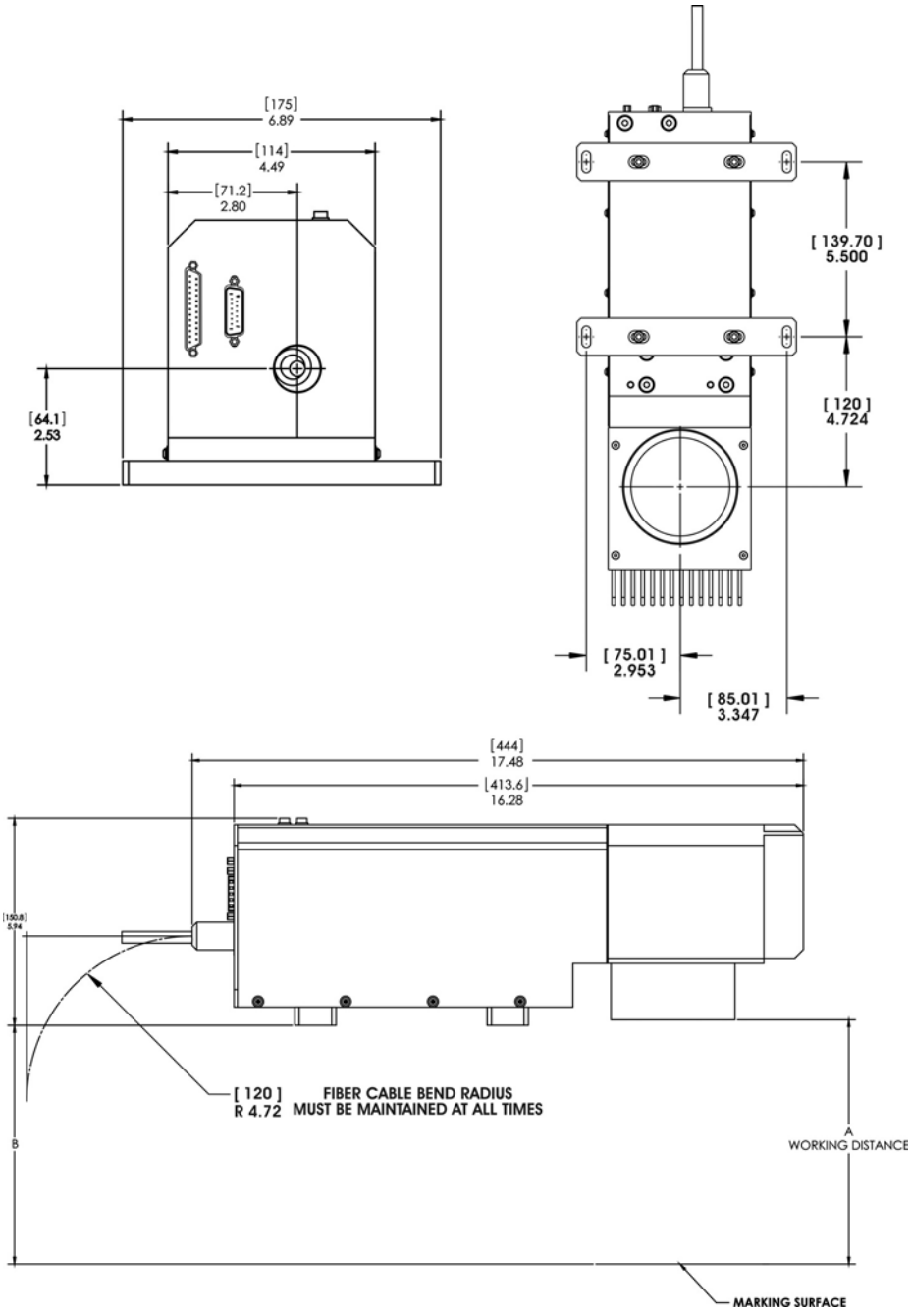
Do ***not*** attempt to remove the fiber at the rear of the marker head under any circumstances. Doing so will ***destroy*** the fiber and void the warranty. The Miyachi Unitek Corporation assumes no liability for such action, the fiber will have to be replaced at the customer's expense.



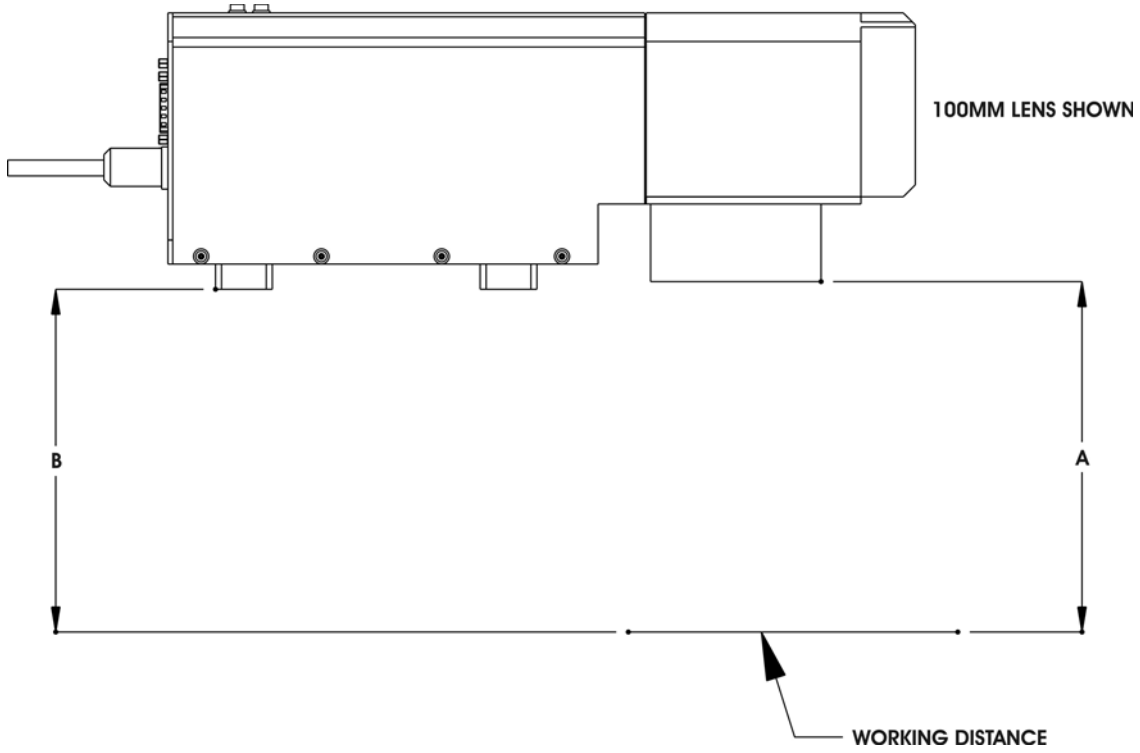
Compact Head > 20W

CAUTION

Do ***not*** attempt to remove the fiber at the rear of the marker head under any circumstances. Doing so will ***destroy*** the fiber and void the warranty. The Miyachi Unitek Corporation assumes no liability for such action, the fiber will have to be replaced at the customer's expense.



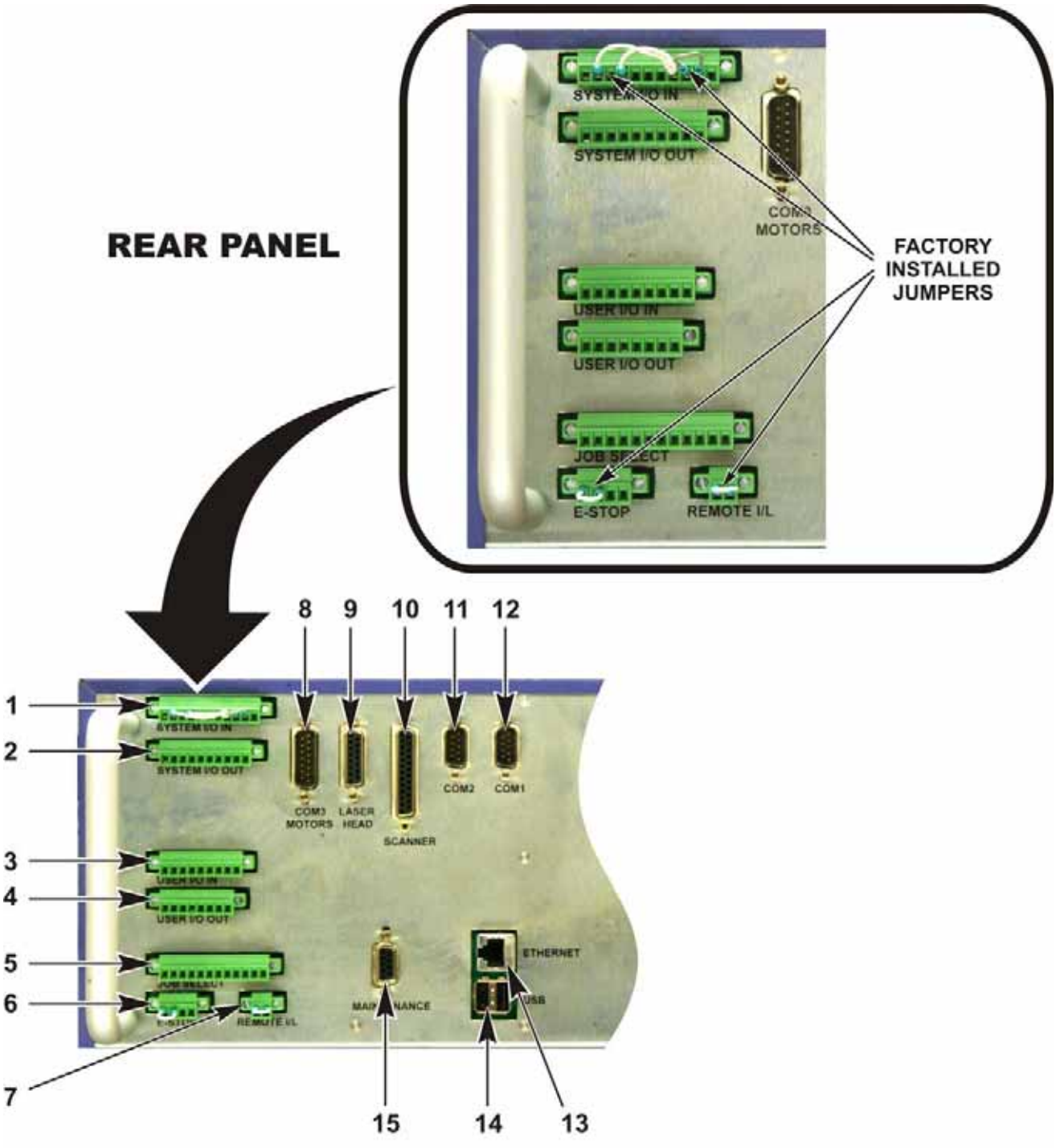
Compact Head Working Distance (all)



f-Theta Lens						
Dimension	**100mm**	**150mm**	**160mm**	**163mm (Obsolete)**	**254mm**	**420mm**
A	3.86” ±.04” (98mm ±1mm)	6.69” ±.08” (170mm ±2mm)	6.93” ±.08” (176mm ±2mm)	7.28” ±.08” (185mm ±2mm)	11.65” ±.12” (296mm ±3mm)	19.45” ±.20” (494mm ±5mm)
B	3.69” ±.04” (93.7mm ±1mm)	6.84” ±.08” (173.7mm ±2mm)	6.76” ± .08” (171.7mm ±2mm)	7.76” ±.08” (197mm ±2mm)	12.57” ±.12” (319.35mm ±3mm)	20.37” ±.20” (517mm ±5mm)

APPENDIX B
ELECTRICAL AND DATA CONNECTIONS

Section I. Rear Panel Connectors



I/O Connections

This marker uses +24V logic as configured by the factory. Using the built in I/O power supply all inputs can be configured to accept either a sourced “high” (+24V) or a sinking “low” (0V) logic level and may be configured with switches, NPN or PNP transistors. Likewise all outputs can be configured to source a “high” (+24V) or sink a “low” (0V) logic level when turned ON. Both the System and User I/O Connectors include an opto-coupler bias input that is used to bias the I/O connector with source or sink capabilities. An external Power Supply (+5 to +24V) may be used to bias the I/O connections at some voltage in this range, or the internal +24V power supply may be used. The maximum I/O current is limited to 50mA.

In all of the examples shown below, the internal +24V supply is used to bias all I/O bits to allow all inputs and outputs to “source”.

Phoenix Connectors

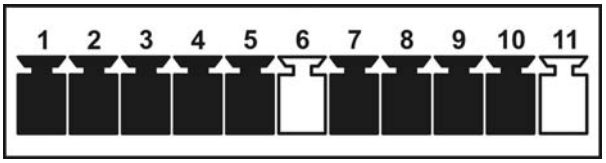
Connectors 1 through 7 on the Rear Panel are Phoenix-type connectors. The control can be configured several different ways in order to match your application needs depending on how these connectors are wired. Configuration is achieved by using connectors with factory-installed jumpers, and by fabricating your own I/O cables.

These connectors use screw-terminal wire connections. No soldering is required which makes configuring, or re-configuring, the connectors a simple task that can be done in just a few minutes.

As shipped, the laser marker includes one 11-pin, one 10-pin, one 9-pin one 8 pin, and one each 4 and 2 pin external I/O Phoenix-type connectors.

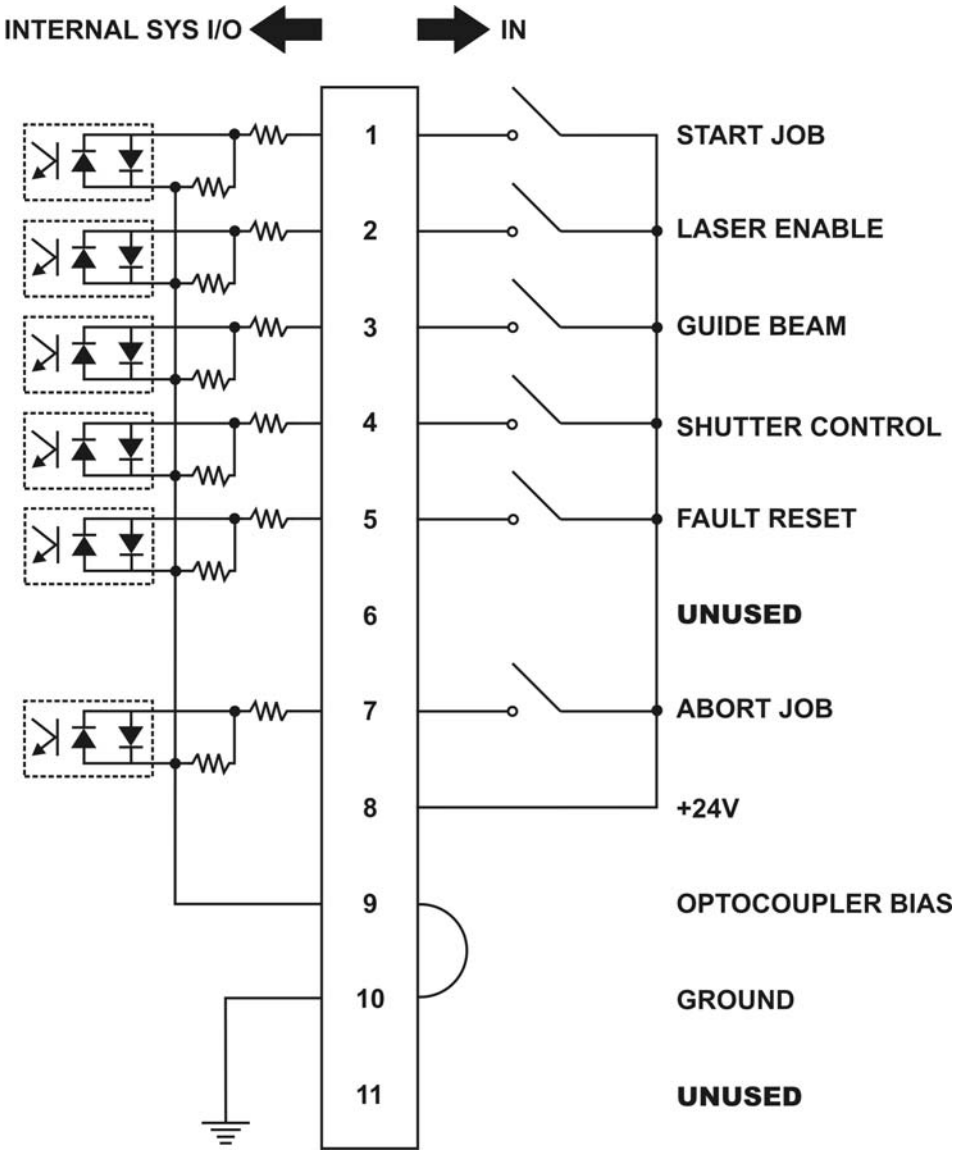
Ensure that all connections are properly strain relieved to prevent pulling out of the connectors.

1. SYSTEM I/O IN Connector

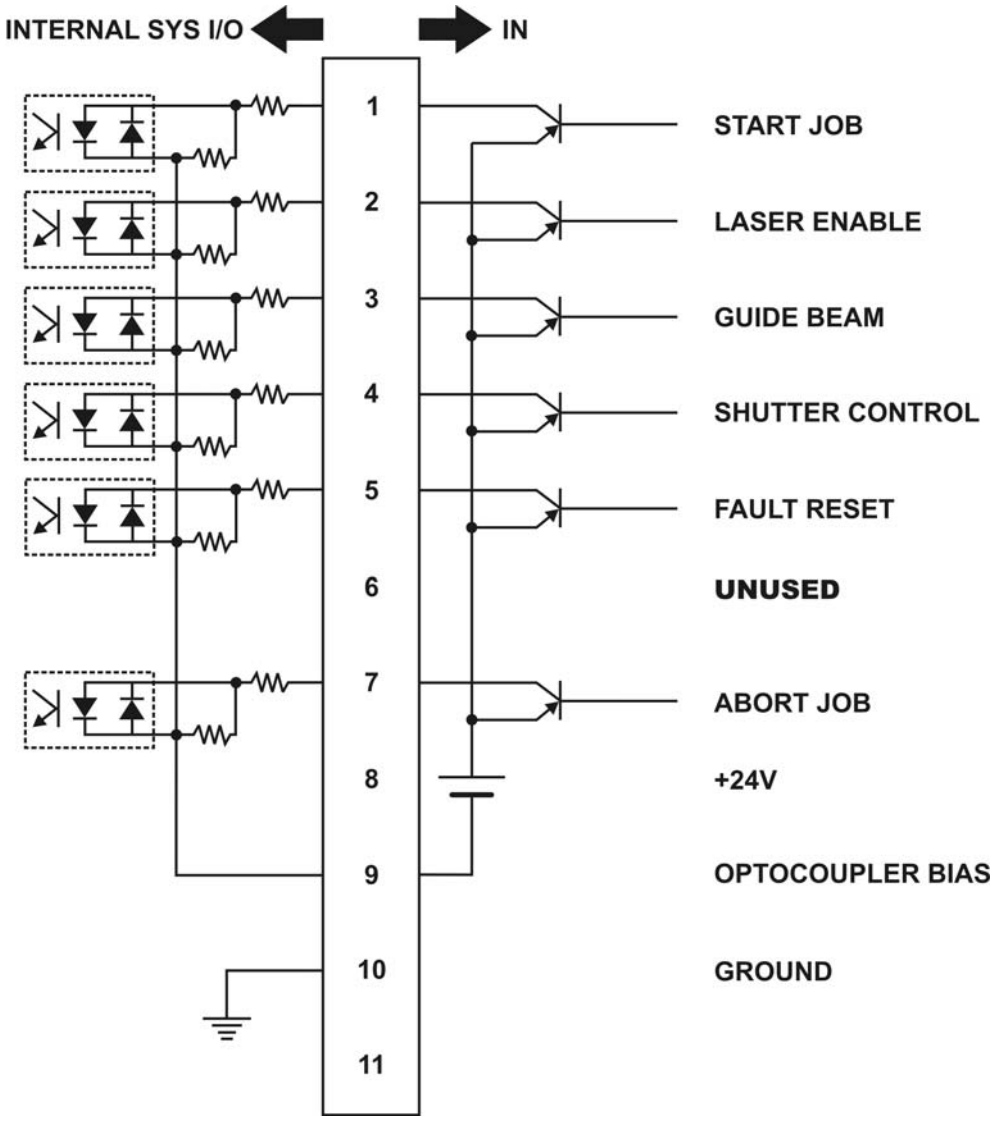


Pin #	Signal	IN/OUT	Description	Note
1	Start Job	IN	**Start Job.** Begins the job sequence.	Photo coupler input
2	Laser Enable	IN	**Laser Enable.** Enables the fiber laser marker. No laser emission is possible without triggering this input.	Photo coupler input
3	Guide Beam	IN	**Guide Beam Control.** Allows external control of visible guide beam	Photo coupler input
4	External Shutter Control	IN	**External Shutter Control.** Allows external control of the safety shutter. This input must be triggered for the shutter to open and permit laser emission.	Photo coupler input
5	Fault Reset	IN	**Fault Reset**. Resets any fault conditions.	Photo coupler input
6	Unused	—	*Unused*	—
7	Abort Job	IN	**Abort Job.** Aborts the current job.	Photo coupler input
8	+24 VDC	—	**+24V**	—
9	DI Common	—	**Optocoupler Bias Input**	—
10	0V	—	**GROUND**	—
11	Unused	—	*Unused*	—

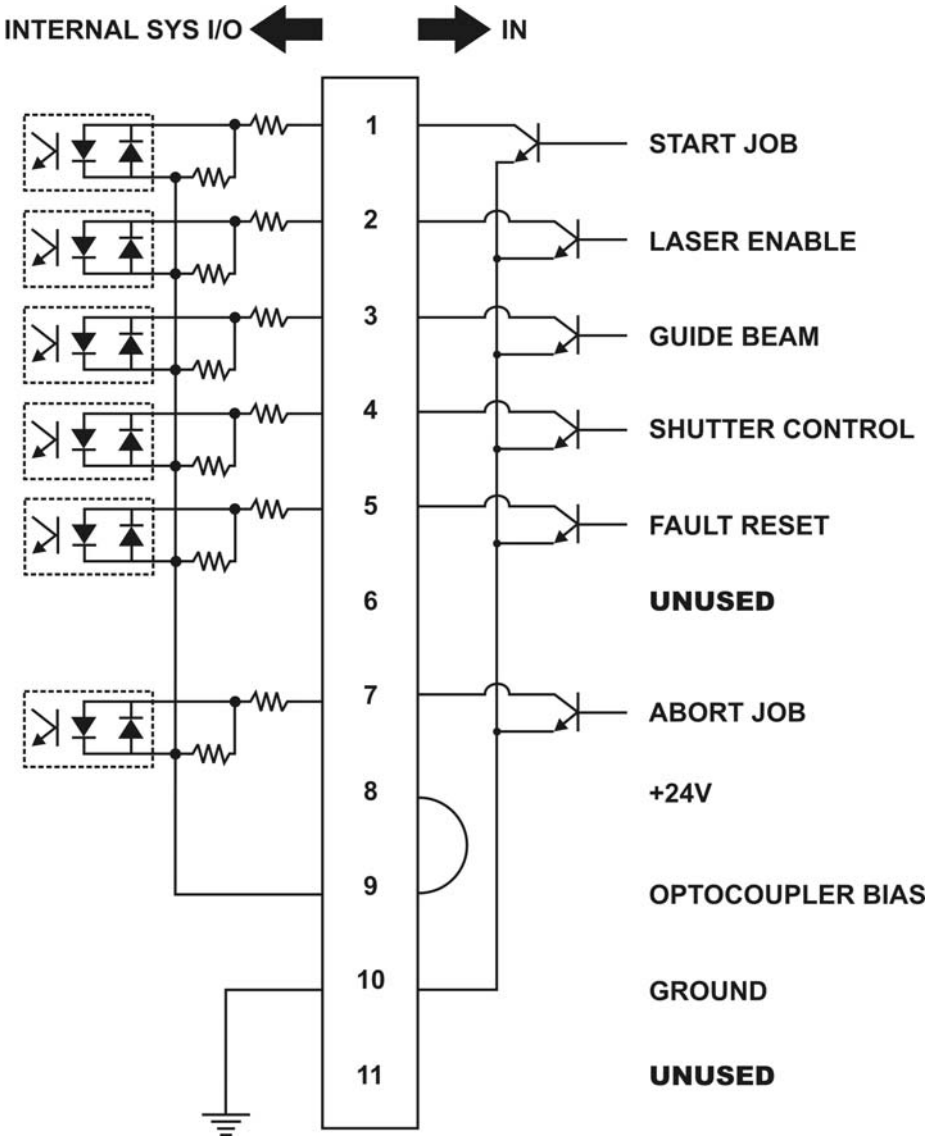
System I/O In • Dry Contact Output



System I/O In • +24 VDC Power Source Output

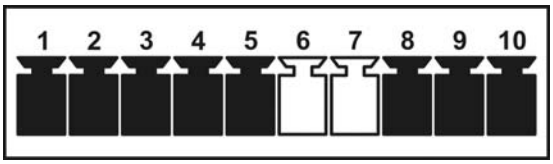


System I/O In • Open Collector



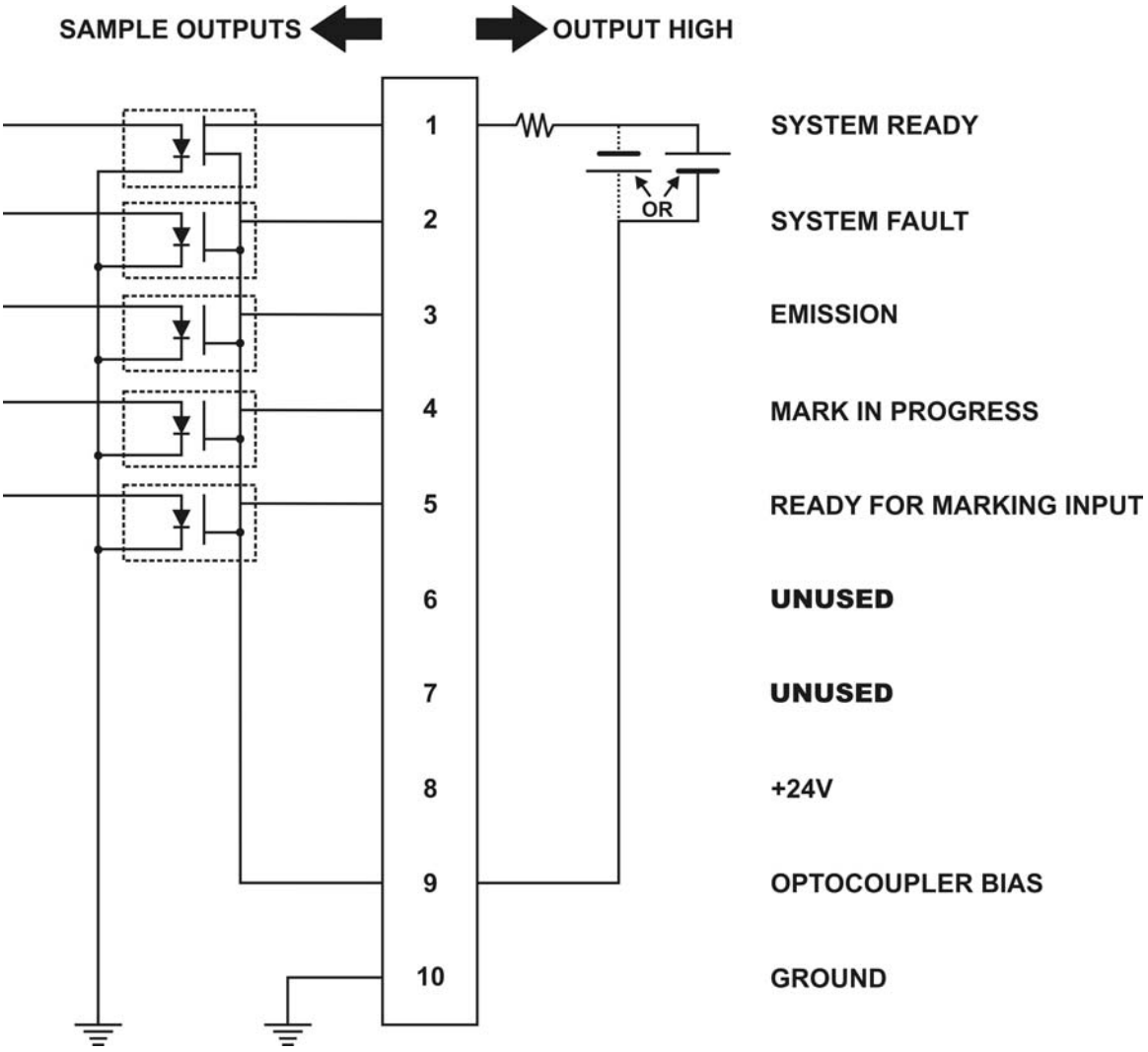
2. SYSTEM I/O OUT Connector

All outputs are configured with photo-MOS output. In order top properly use these outputs, they will need to be biased by an external resistor. In the examples below, a 10kΩ resistor is used to keep the output from floating. The Outputs can either sink or source depending on how the optocoupler bias jumper and resistors are configured.

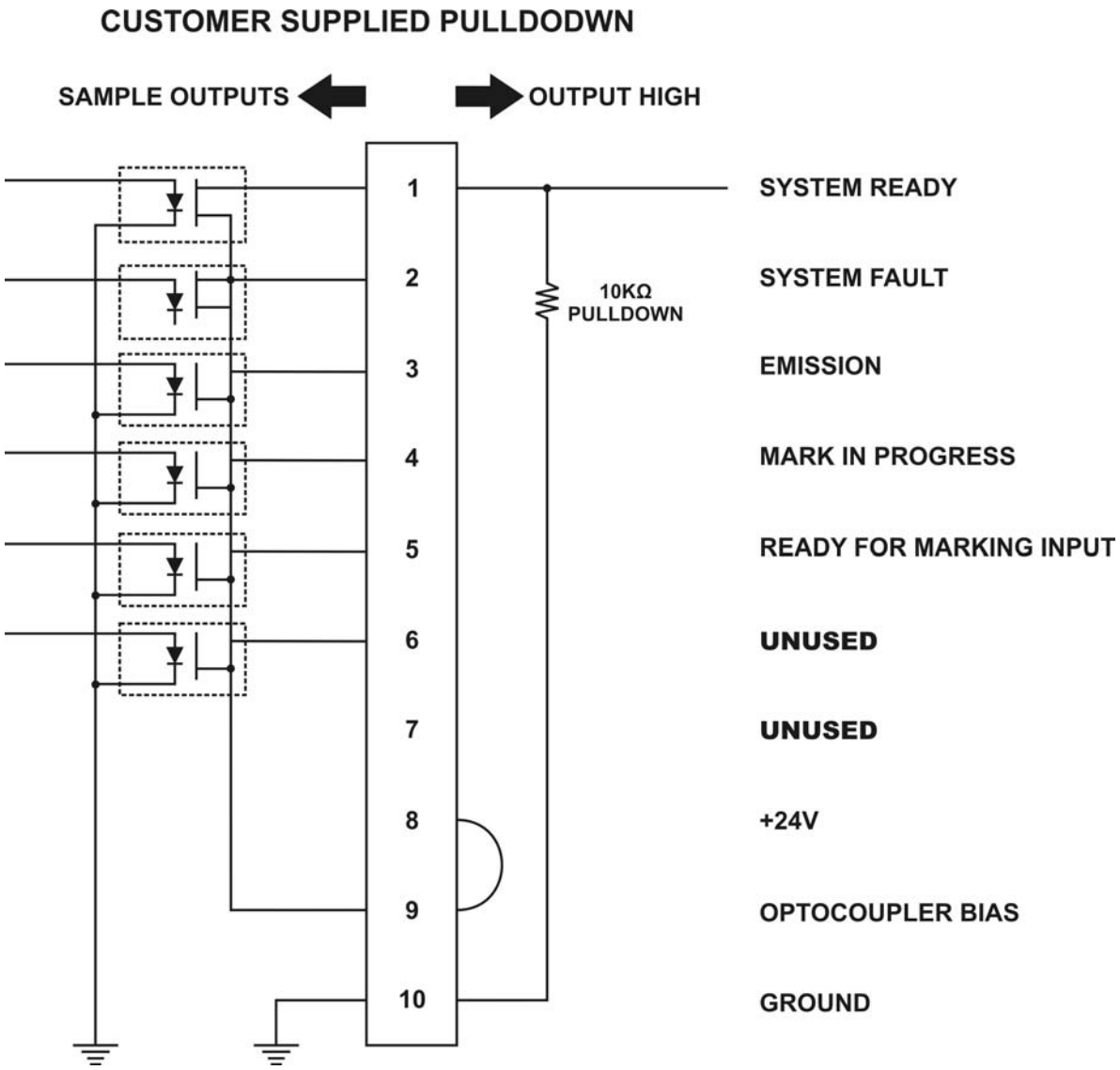


Pin #	Signal	IN/OUT	Description	Note
1	System Ready	OUT	**System Ready.** The system is ready for laser processing. No faults are present.	Photo MOS output
2	System Fault	OUT	**System Fault.** A fault condition exists that must be cleared.	Photo MOS output
3	Job Busy	OUT	**Laser Emission In Progress (Job Busy).** A laser marking job is currently being executed. This output will be active from the time the job is executed until it is completed.	Photo MOS output
4	Mark in Progress	OUT	**Mark In Progress**. The laser is currently marking a segment. This output will trigger on and off rapidly as the laser marks the various components of a job.	Photo MOS output
5	Ready for Mark	OUT	**Ready for Mark Input.** A job has been loaded and is ready to execute.	Photo MOS output
6	Unused	—	*Unused*	—
7	Unused	—	*Unused*	—
8	+24 VDC Output	—	**+24V**	—
9	DO Common	—	**Optocoupler Bias Input**	—
10	0V	—	**GROUND**	—

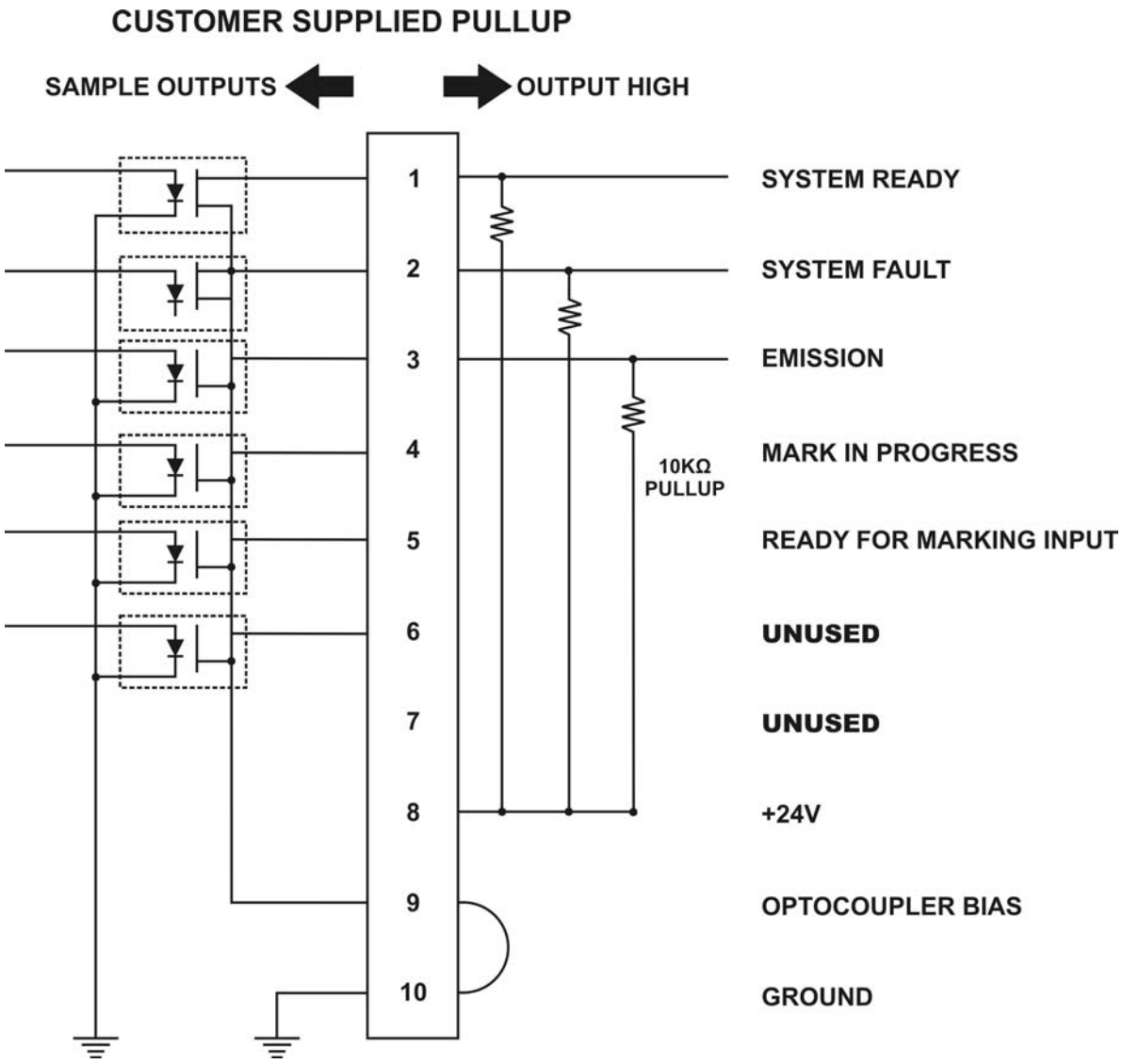
System I/O Out • External Power Supply



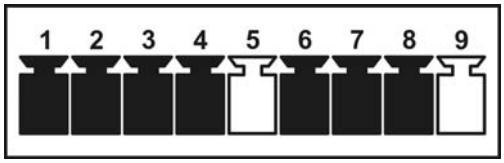
System I/O Out • Pulldown Output



System I/O Out • Pullup Output

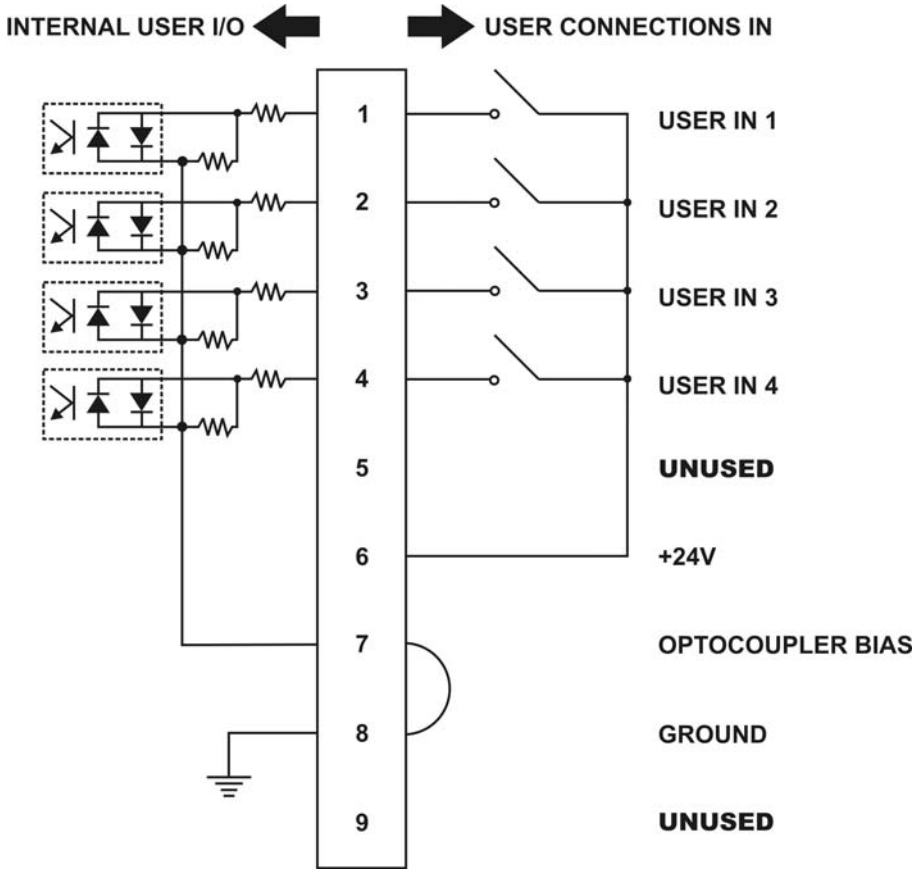


3. USER I/O IN Connector

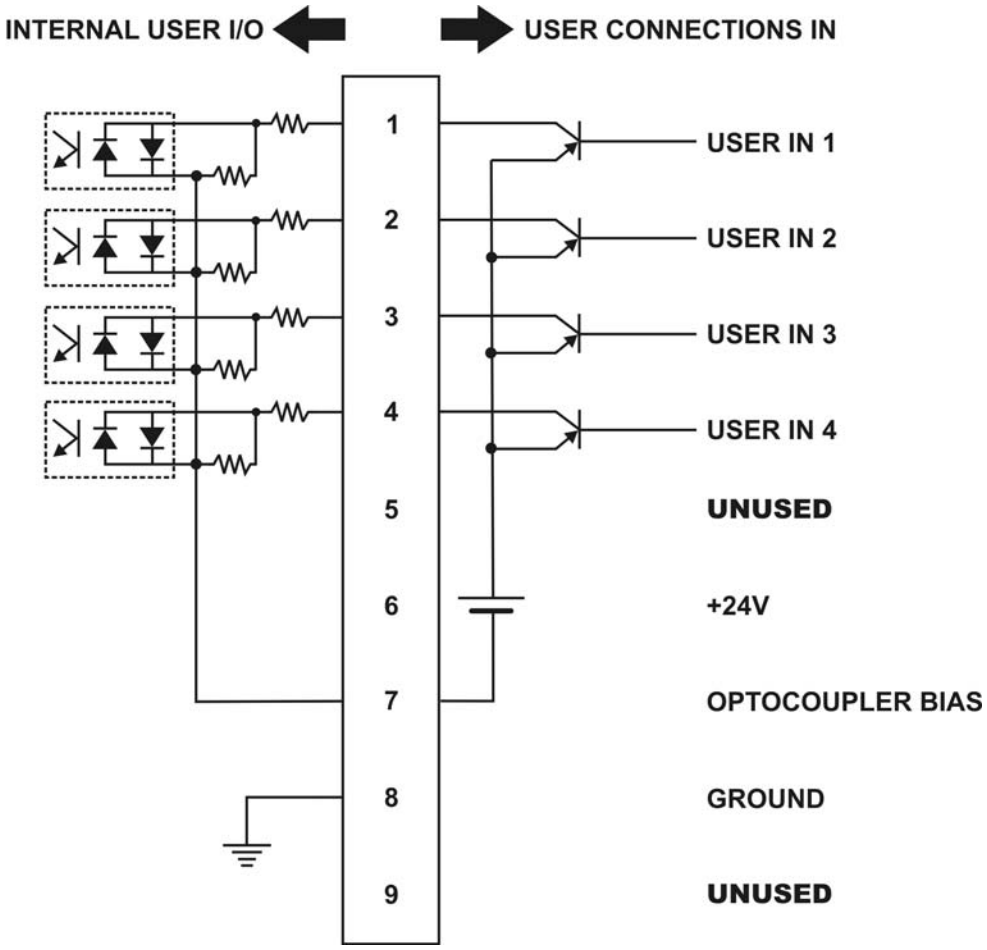


Pin #	Signal	IN/OUT	Description	Note
1	User Input 1	IN	**User Input 1**	Photo coupler input
2	User Input 2	IN	**User Input 2**	Photo coupler input
3	User Input 3	IN	**User Input 3**	Photo coupler input
4	User Input 4	IN	**User Input 4**	Photo coupler input
5	Unused	—	*Unused*	—
6	+24 VDC	—	**+24V**	—
7	DI Common	—	**Optocoupler Bias Input**	—
8	0V	—	**GROUND**	—
9	Unused	—	*Unused*	—

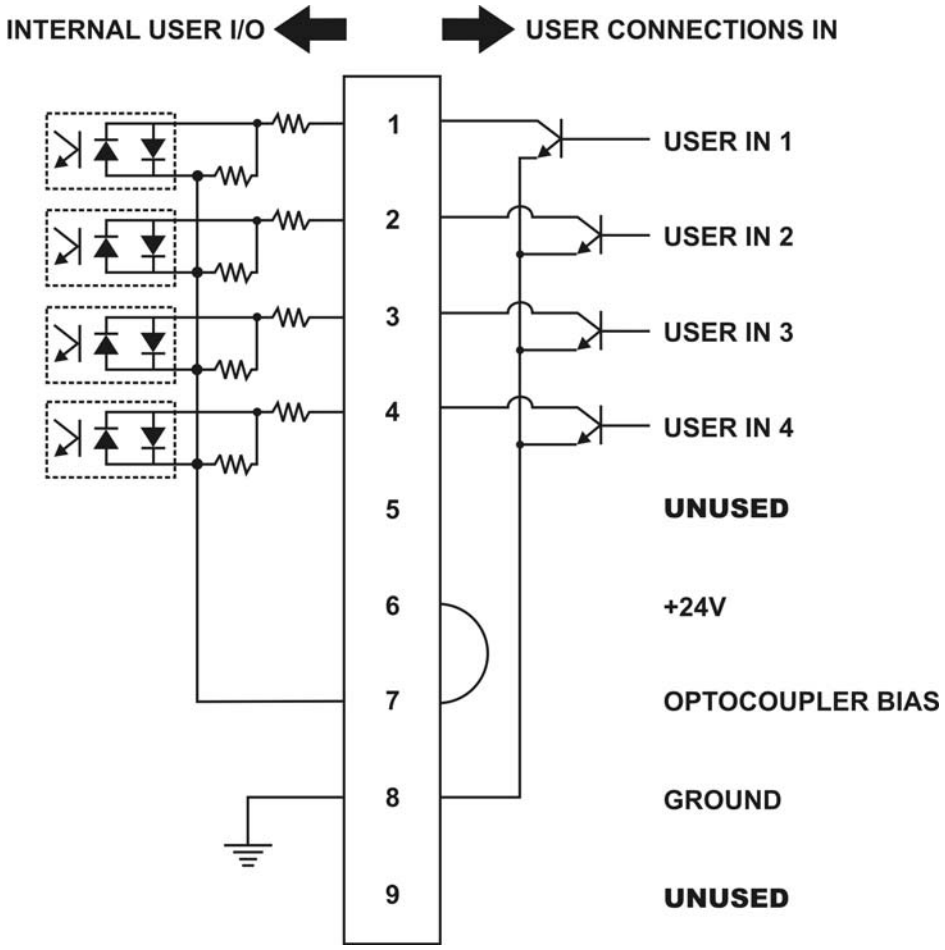
User I/O In • Dry Contact



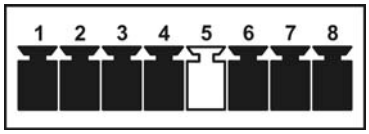
User I/O In • +24 VDC Source



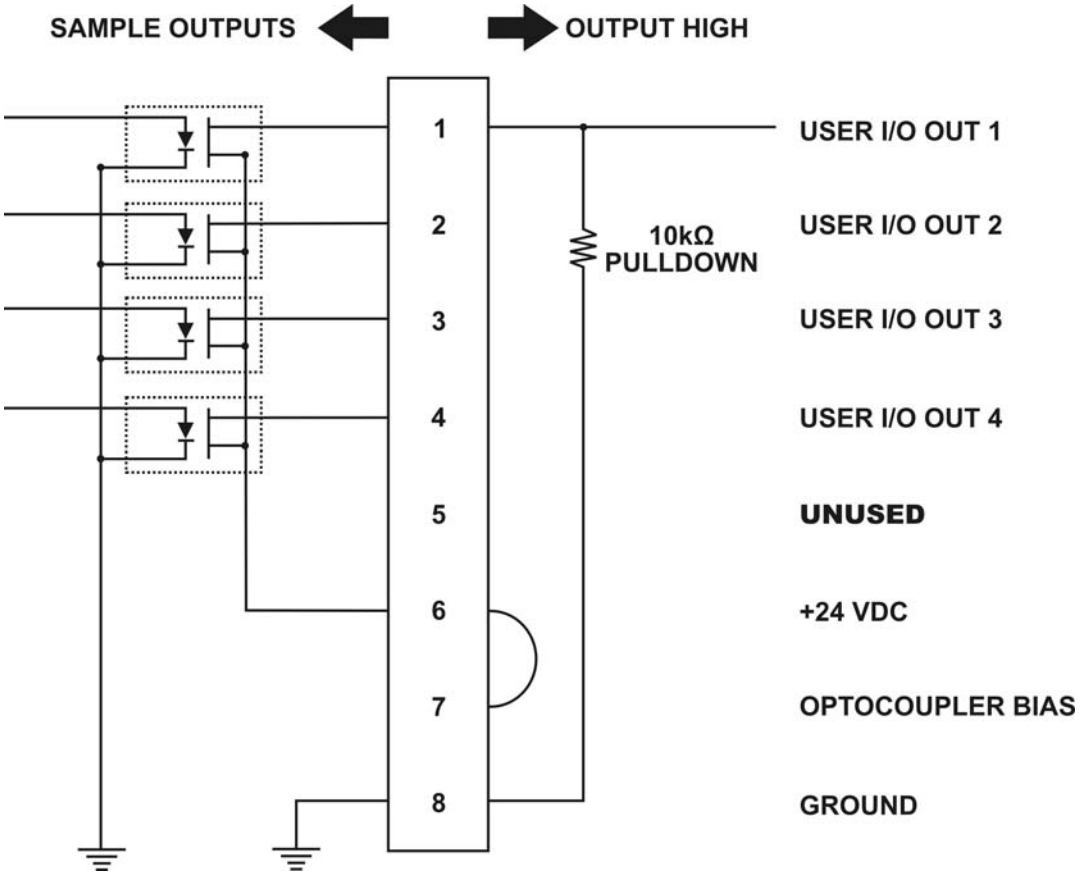
User I/O In • Open Collector



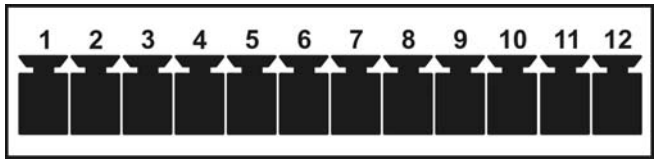
4. USER I/O OUT Connector



Pin #	Signal	IN/OUT	Description	Note
1	User Output 1	OUT	**User Output 1**	Photo coupler input
2	User Output 2	OUT	**User Output 2**	Photo coupler input
3	User Output 3	OUT	**User Output 3**	Photo coupler input
4	User Output 4	OUT	**User Output 4**	Photo coupler input
5	*Unused*	—	*Unused*	—
6	+24 VDC Output	—	**+24V**	—
7	DO Common	—	**Optocoupler Bias Input**	—
8	0V	—	**GROUND**	—

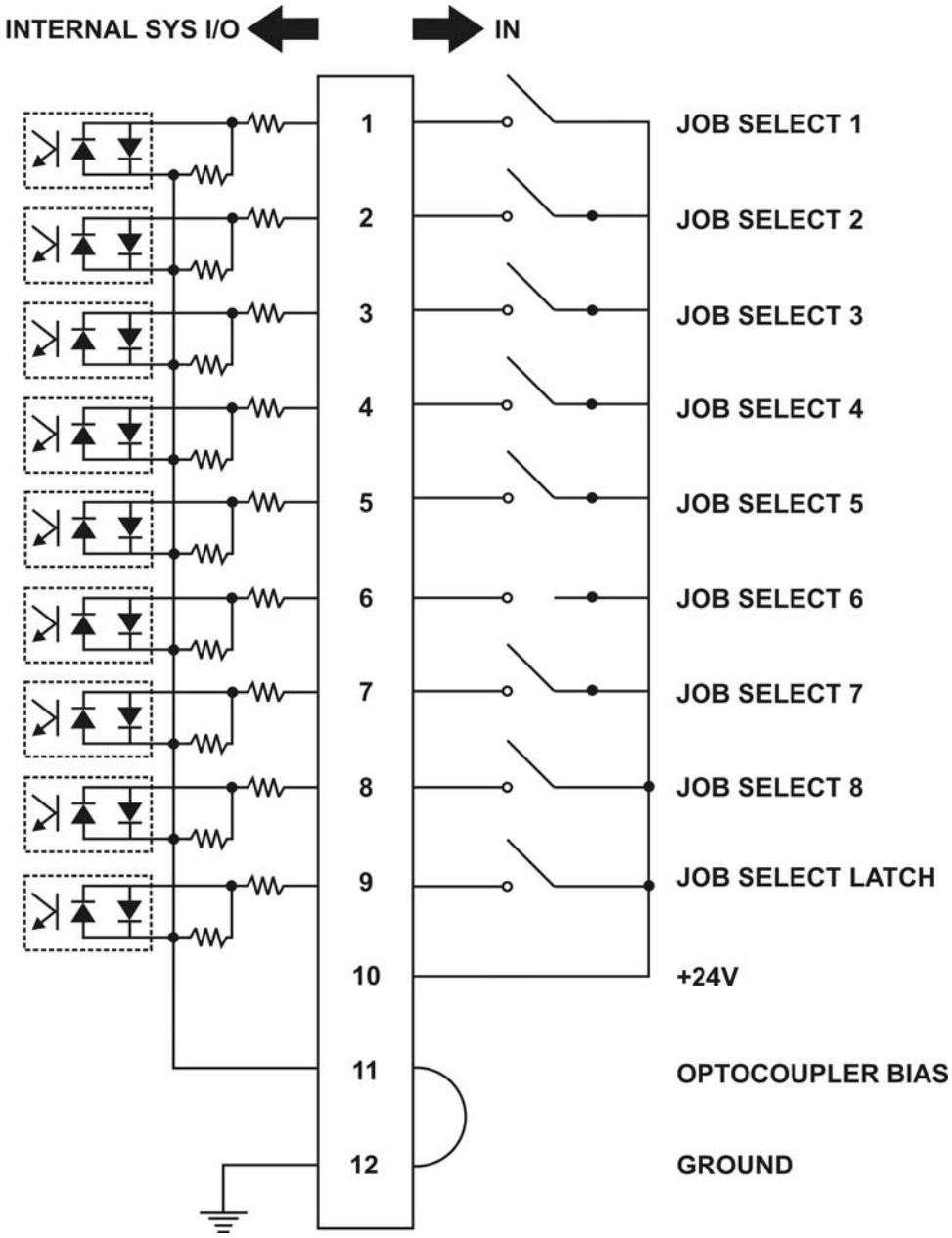


5. JOB SELECT Connector

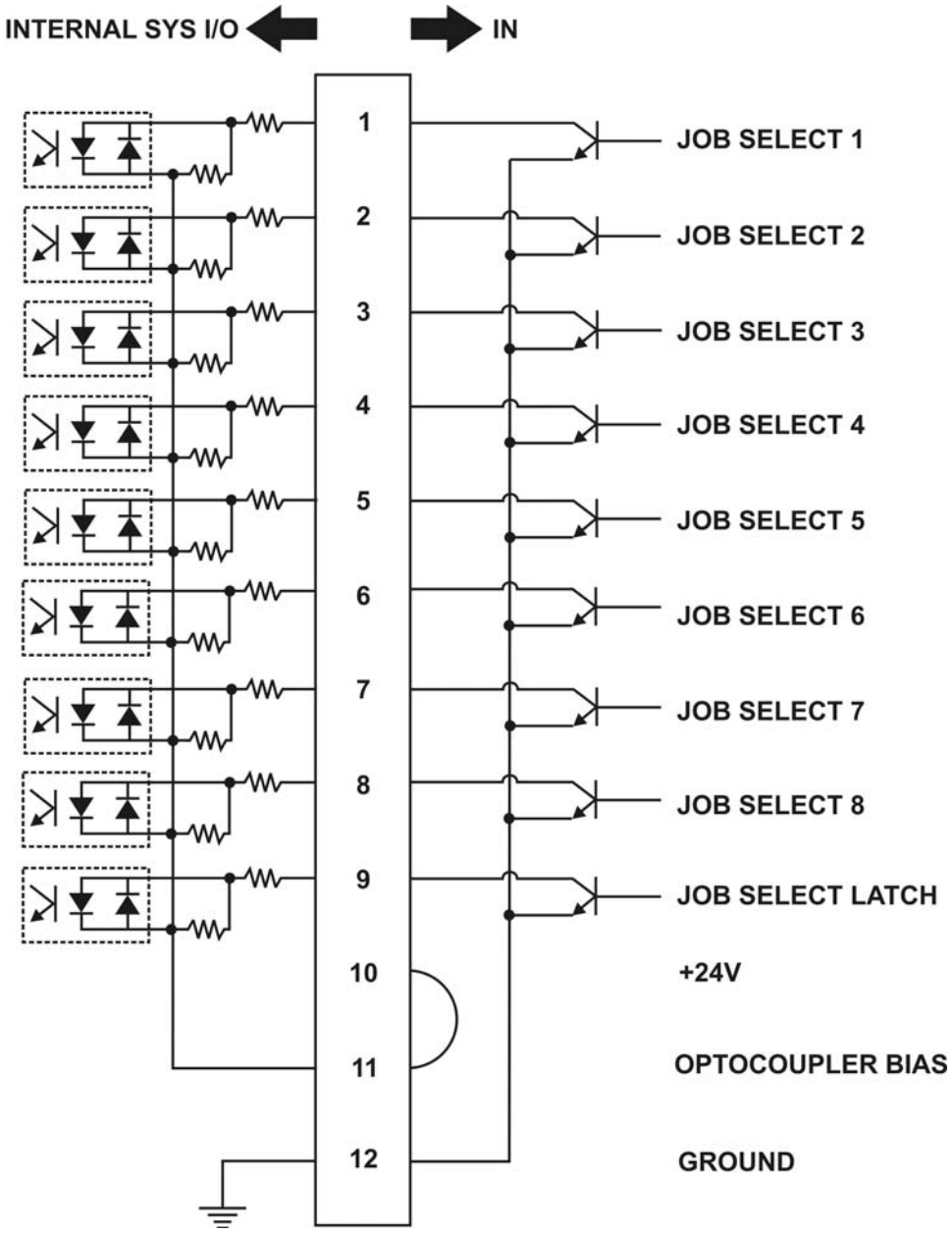


Pin #	Signal	IN/OUT	Description	Note
1	Job Select Bit 1	IN	**Job Select Bit 1**	Photo coupler input
2	Job Select Bit 2	IN	**Job Select Bit 2**	Photo coupler input
3	Job Select Bit 3	IN	**Job Select Bit 3**	Photo coupler input
4	Job Select Bit 4	IN	**Job Select Bit 4**	Photo coupler input
5	Job Select Bit 5	IN	**Job Select Bit 5**	Photo coupler input
6	Job Select Bit 6	IN	**Job Select Bit 6**	Photo coupler input
7	Job Select Bit 7	IN	**Job Select Bit 7**	Photo coupler input
8	Job Select Bit 8	IN	**Job Select Bit 8**	Photo coupler input
9	Job Select Bit Latch	IN	**Job Select Latch**	Used to latch bit combinations for job loads; Photo coupler input
10	+24V DC Output	—	**+24V**	
11	DI COMMON	IN	**Optocoupler Bias Input**	
12	0V	—	**GROUND**	

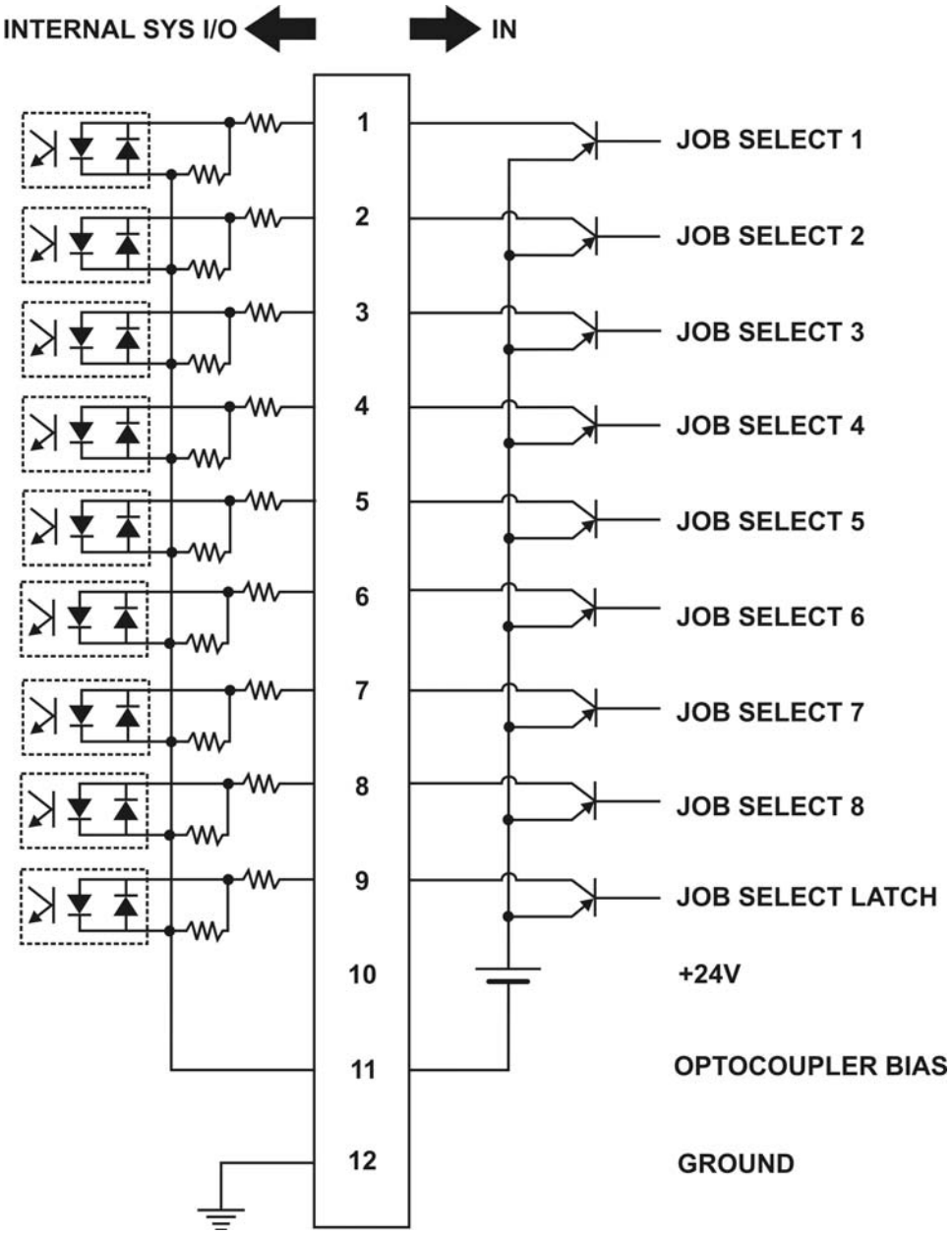
Job Select • Dry Connector



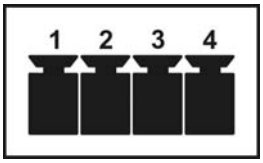
Job Select • Open Collector



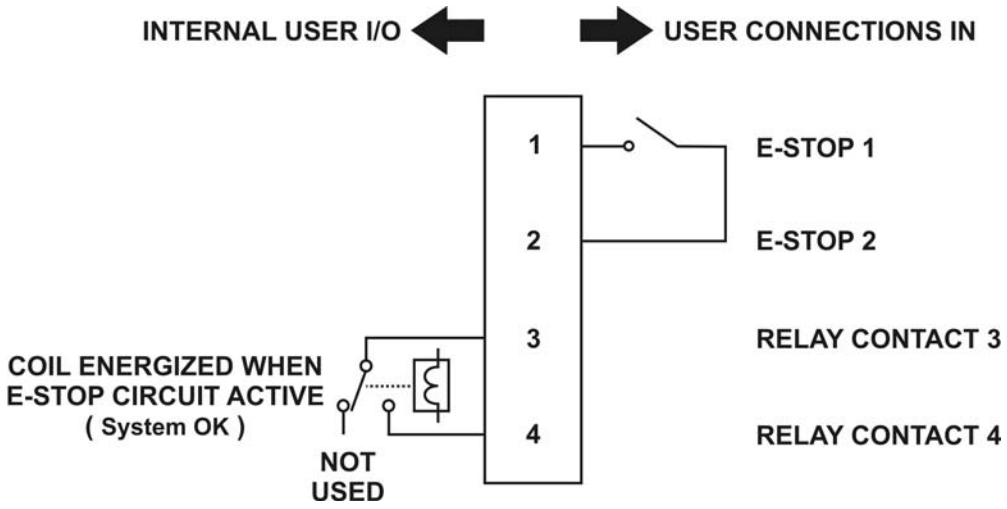
Job Select • +24V Power Source Output



6. E-STOP Connector



Pin #	Signal	IN/OUT	Description	Note
1	E-Stop Input 1	IN	**E-Stop 1**	Contact input
2	E-Stop Input 2	IN	**E-Stop 2**	Contact input
3	E-Stop Output 3	OUT	**Relay Contact 3**	Contact output
4	E-Stop Output 4	OUT	**Relay Contact 4**	Contact output

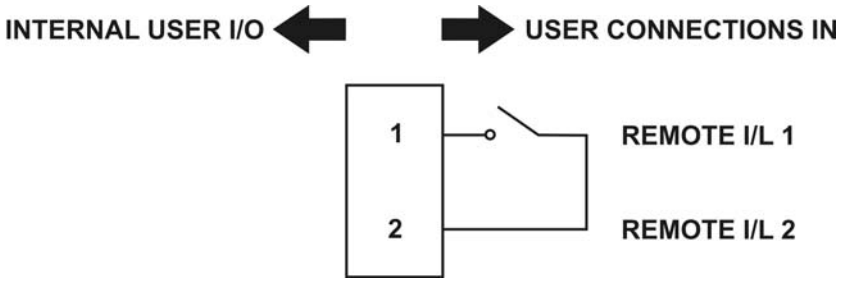


NOTES:

- Pins **1** and **2** ***must*** be connected together by a **DRY CONTACT ONLY** to close the **Emergency Stop** circuit and enable laser operation.
- Pins **3** and **4** are connected to a relay which is closed during normal operation and open if the laser marker is in the **E-Stop** state. Wire this in series with auxiliary equipment or automation system **E-Stop** circuitry to allow the laser marker **E-Stop** to affect outside equipment. The relay supports up to 8A at 30VDC.

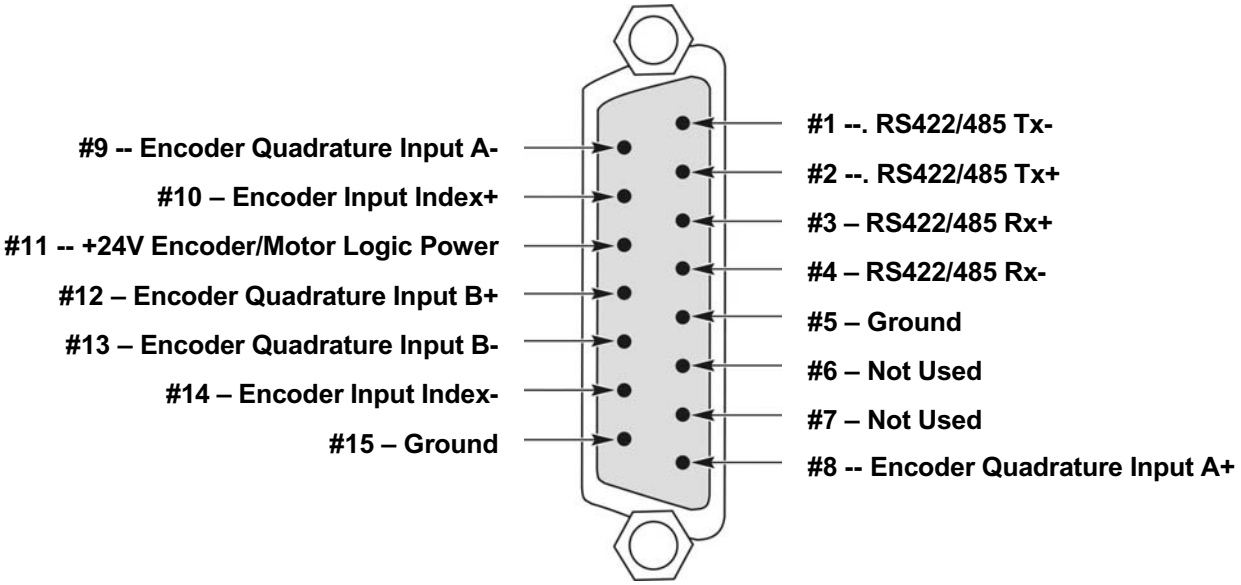
7. REMOTE I/L Connector

Pin #	Signal	IN/OUT	Description	Note
1	Remote Interlock 1	IN	**Remote I/L 1**	Remote I/L Circuit Supply
2	Remote Interlock 2	IN	**Remote I/L 2**	Remote I/L Circuit Return



NOTE: Remote I/L pins **1** and **2** ***must*** be connected by a **DRY CONTACT ONLY** to close the interlock circuit. If the interlock circuit is open, the safety shutter can ***not*** open and the laser can ***not*** be enabled.

8. COM3 MOTORS Connector



NOTE: The maximum current for the **+24V Encoder/Motor Logic Power Supply** (Pin #11) is 800mA.

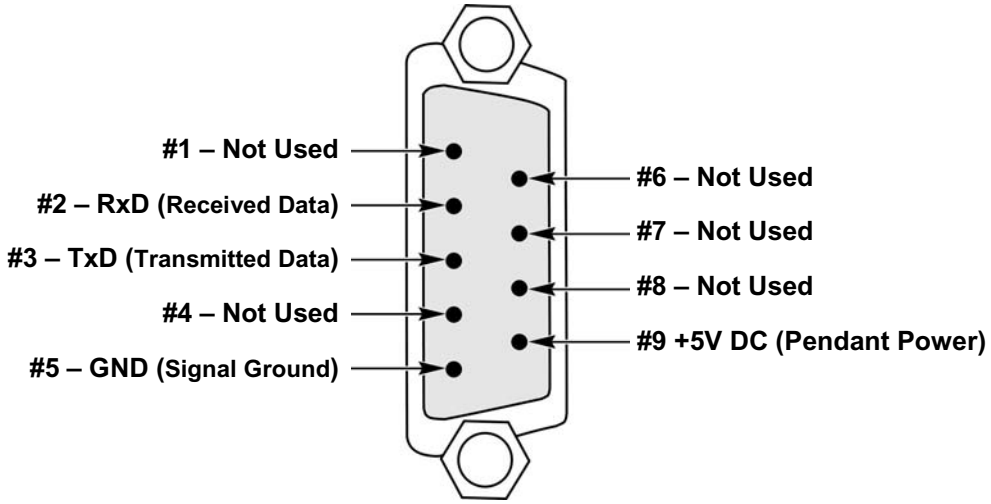
9. LASER HEAD Connector

Connects to the Laser Head.

10. SCANNER Connector

Connects to the Laser Head.

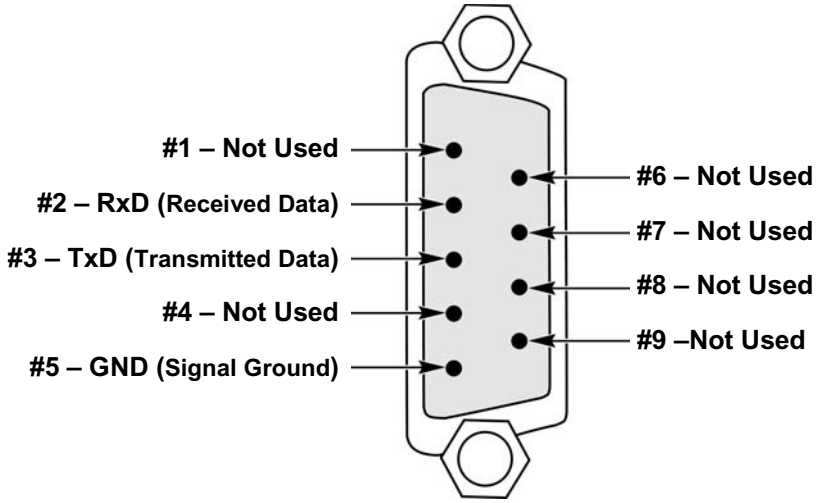
11. COM2 Connector



NOTES:

- The unit communicates via a simplified 3-wire RS232 implementation with **Hardware Flow Control disabled.** Use pins 2, 3, and 5 for **Received Data**, **Transmitted Data**, and **Ground** respectively. Pin 9 provides **+5V** to power a remote control pendant.
- Use a Null-Modem type cable (cross cable) when communicating with a PC.

12. COM1 Connector



NOTES:

- The unit communicates via a simplified 3-wire RS232 implementation with **Hardware Flow Control disabled.** Use pins 2, 3, and 5 for **Received Data**, **Transmitted Data**, and **Ground** respectively.
- Use a Null-Modem type cable (cross cable) when communicating with a PC.

13. ETHERNET Connector

Standard LAN connector. Use for connecting to a Remote API or *Winlase* software.

14. USB Connectors

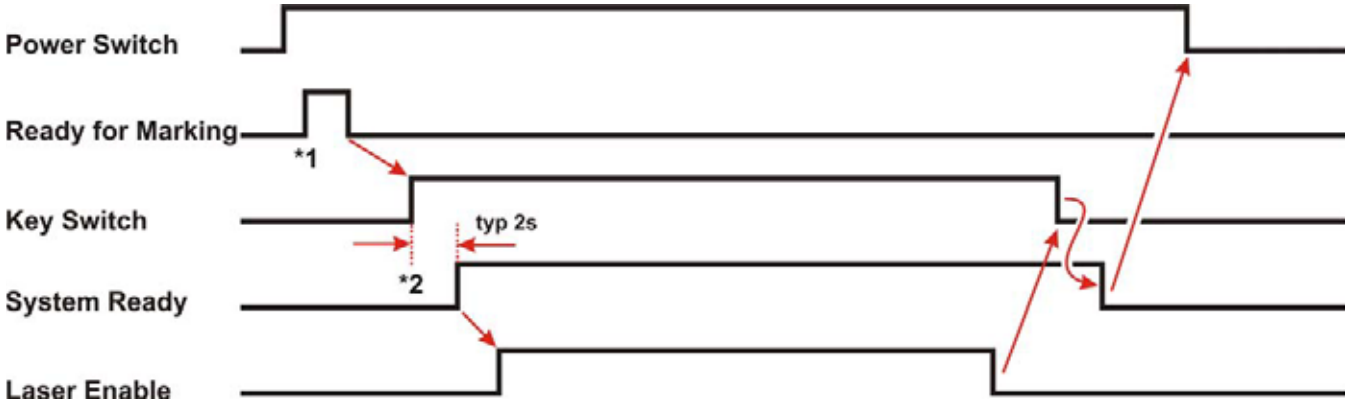
Two Type B USB Ports are available on the back of the unit for adding additional storage space using USB Flash Drives, or “thumb drives”. Not all flash drives are supported. One suggested brand and model is the *PNY Optima Pro Attache 2GB*. Do ***not*** connect any USB device other than a USB flash drive to these ports.

15. MAINTENANCE Connector

Reserved for factory diagnostic use.

Section II. Timing Diagrams

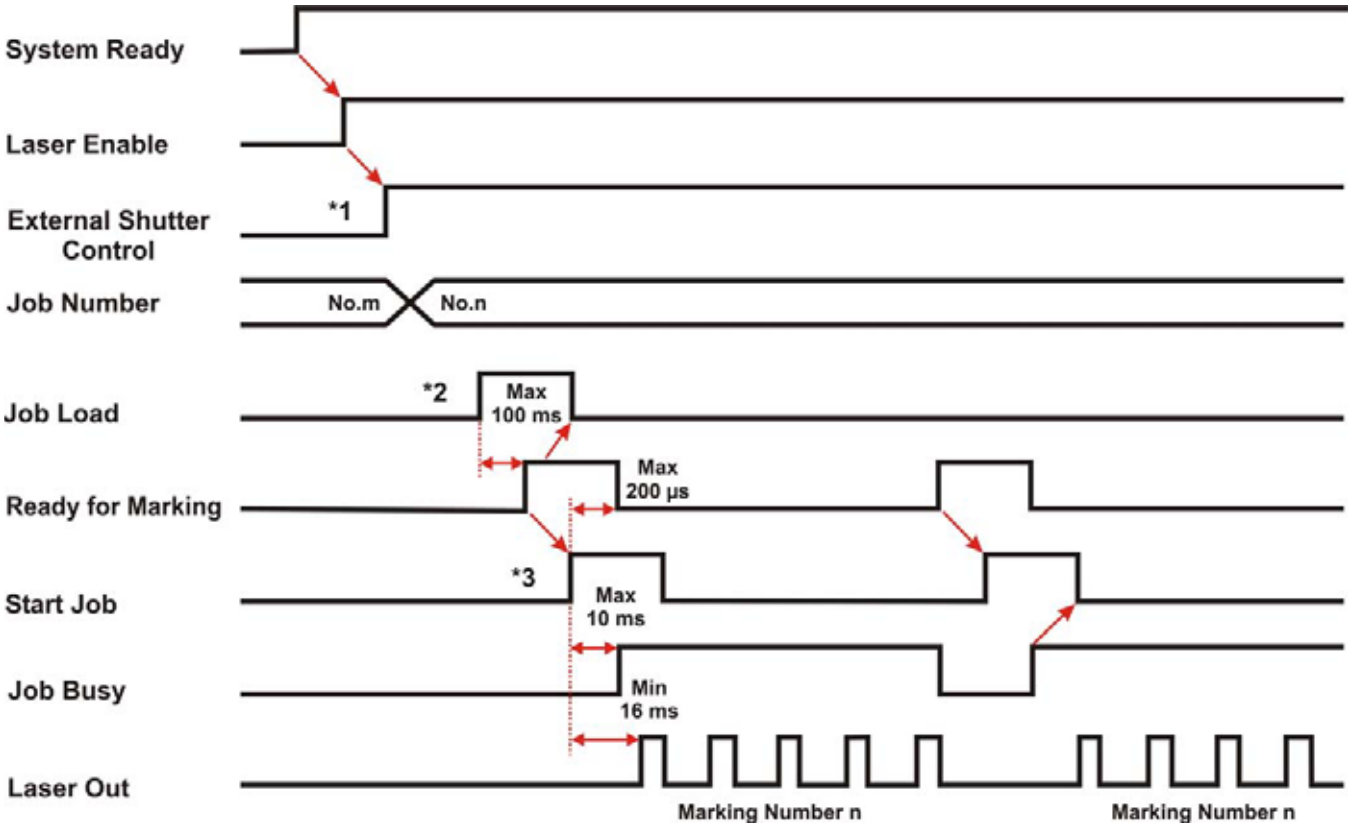
Power ON / OFF



*1 After turning the **Power Switch** ON, the **Ready for Mark** signal turns ON for several seconds as part of the self-check routine. It can take up to 30 seconds to boot the marker control card. Once remote API or Winlase streaming access is available it is possible to proceed

*2 The **System Ready** turns ON two seconds after the Key switch turns ON

Normal Operation (Local Execution Mode)

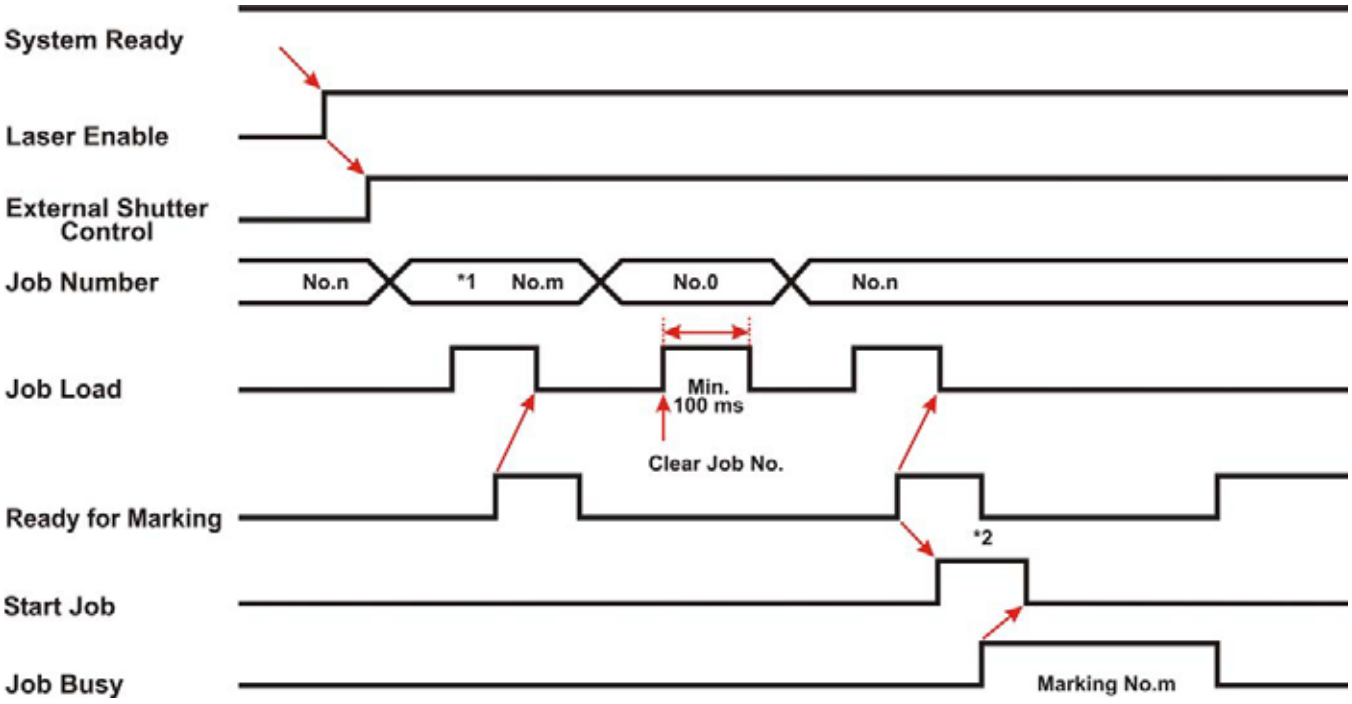


*1 It is necessary to turn the **External Shutter Control** ON for laser marking. When the **External Shutter Control** is OFF, the laser does not fire because the safety shutter is closed. ***You must wait 100ms after opening the shutter before sending a start job command.***

*2 If cache mode is not enabled in the Winlase I/O Job Load mode, it may be necessary to wait several seconds for **Job Load** to turn ON.

*3 The **Start Mark** should be input once the **System Ready** is ON, the **External Shutter** control has been ON for at least 100ms, the **Laser Enable** is ON, and the **Ready for Mark** is ON. If the **Start Mark** turns ON and **Laser Enable** is OFF, the laser does not fire. **Start Mark** can be configured to trigger on rising edge, sinking edge, high level, or low level. The system will fault if **Start Mark** is sent within 100ms of **External Shutter**.

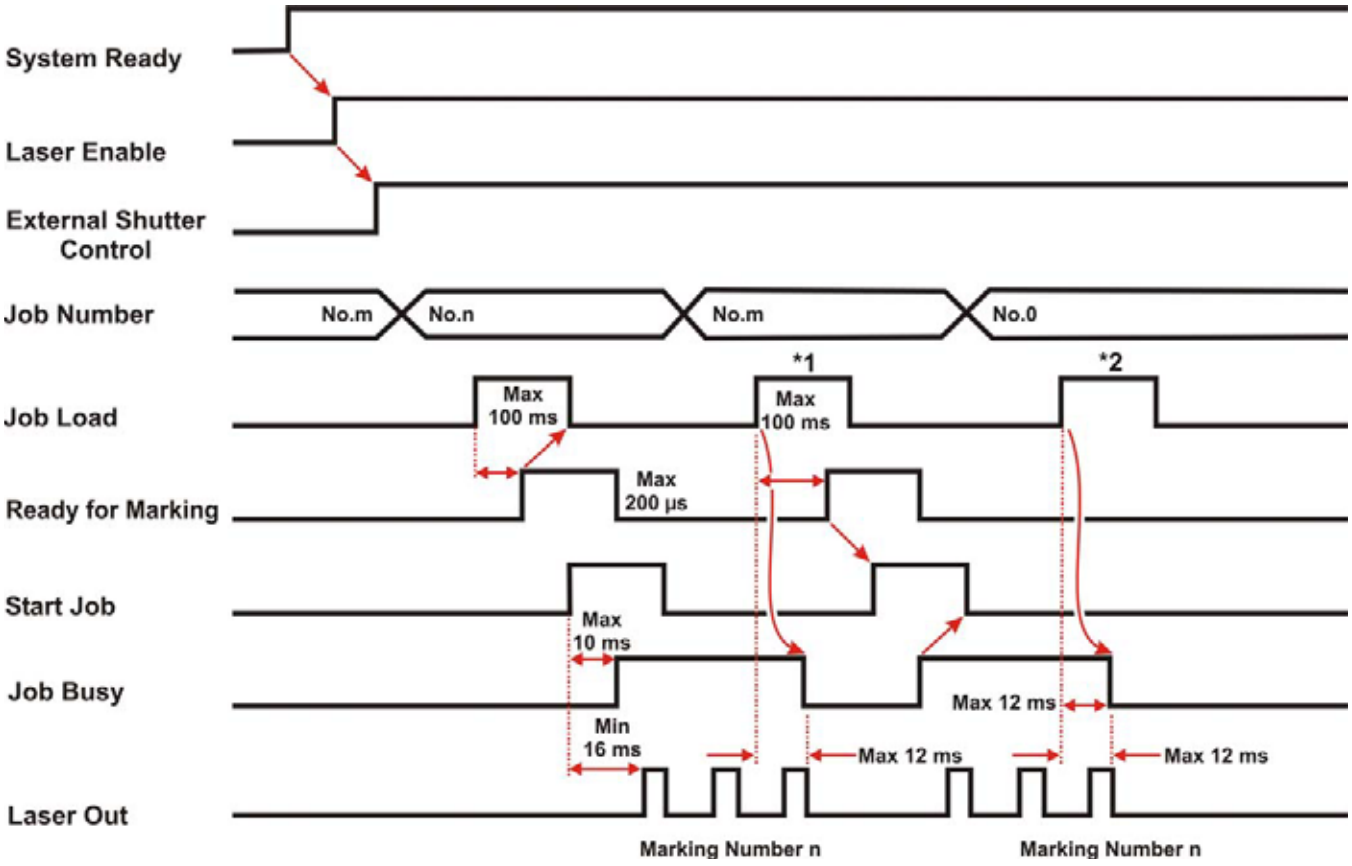
Job Select (Local Execution Mode)



*1 **Job Number** is an 8-bit binary number equivalent to integers between 1 and 255. If you set the binary number to zero, indicating **Job No.0**, and toggle the **Job Load** bit ON, the **Job Number** is cleared. (**Ready for Mark** turns OFF or remains OFF). In addition, if you set a Job No. that has not been downloaded into the unit, and toggle ON the Job Load, **Ready for Mark** will turn OFF or remain OFF.

*2 Toggle **Start Mark** ON after **Ready for Mark** turns ON to start the cycle.

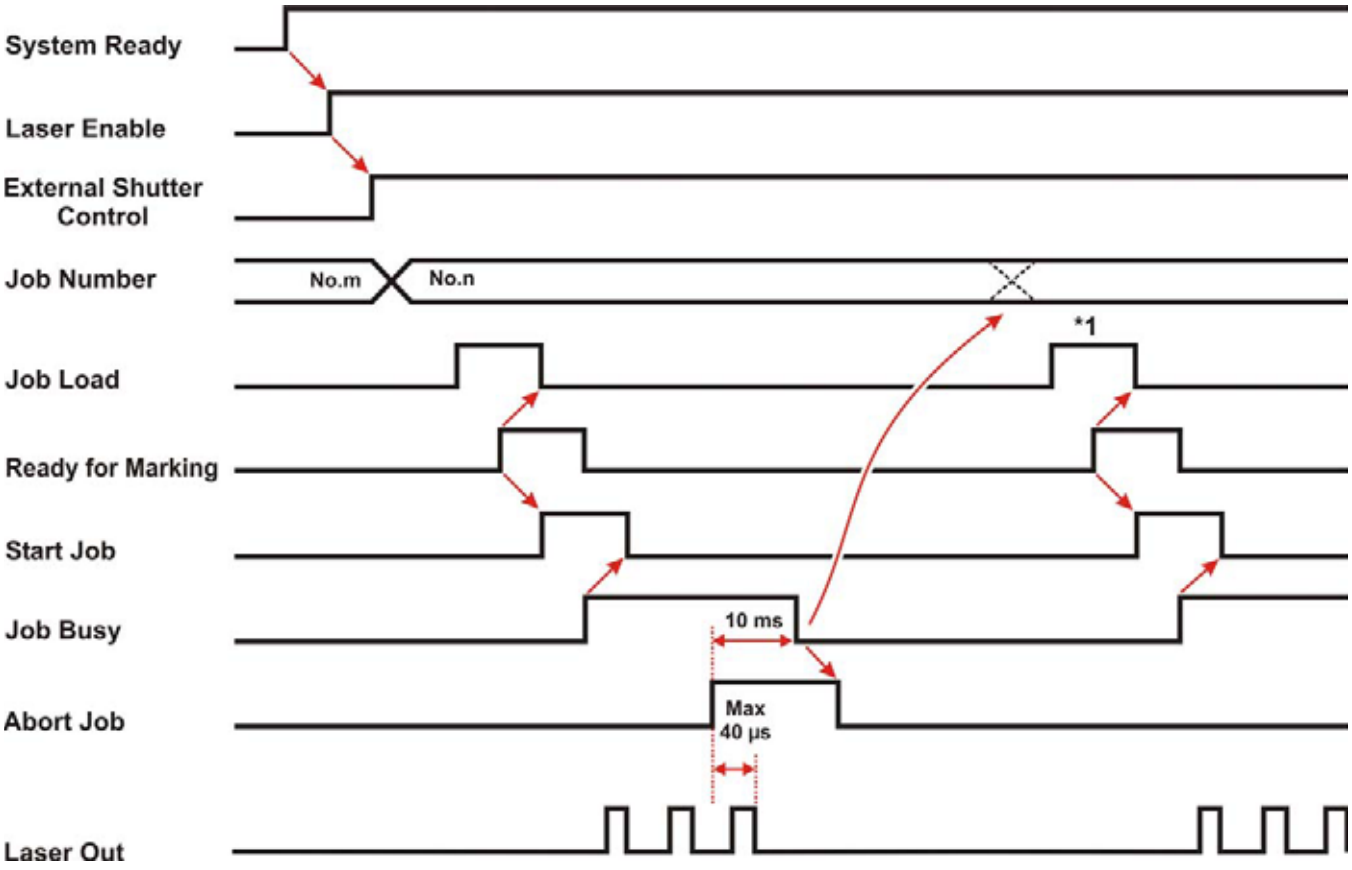
Job Select (During Marking)



*1 When you set a **Job Number** that exists in unit memory and toggle **Job Load** ON during a mark cycle, marking execution stops. In addition, the Job No. changes to the selected one. After the **Ready for Mark** turns ON, then when the **Start Mark** turns ON, the marking execute with the selected **Job Number**.

*2 When you set the **Job No. 0**, and toggle ON the **Job Load**, the marking execution stops. Then the **Job Number** is cleared. (However the **Ready for Mark** does not turn ON). In addition, if you select a **Job Number** that has not been previously downloaded into the unit, and toggle the **Job Load** ON, the **Job Number** is cleared too and **Ready for Mark** will turn OFF or remain OFF. If you want to start marking again, you have to select a valid job number and toggle the **Job Load** ON.

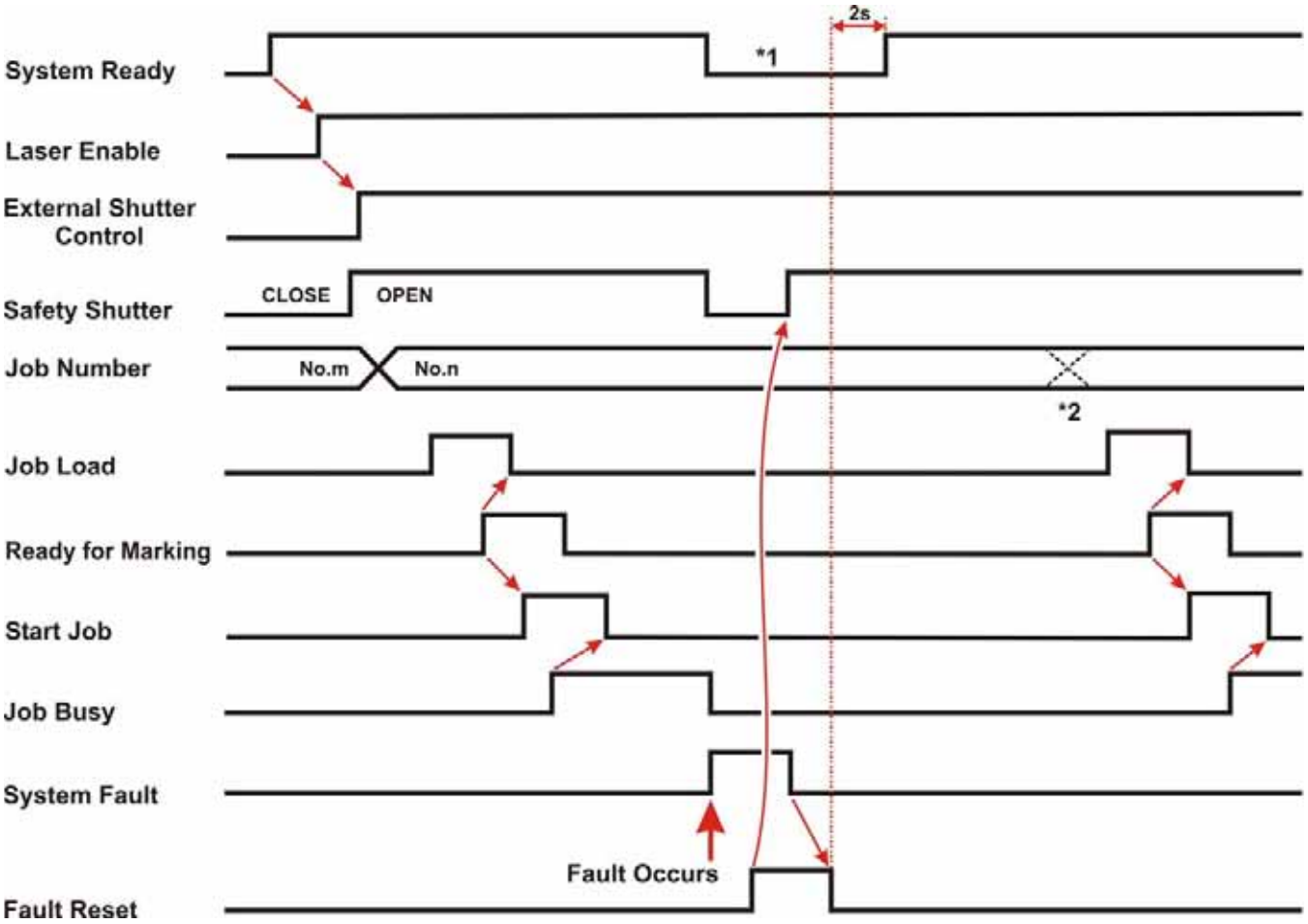
JOB ABORT – Requires Winlase Interlock Configuration



*1 After you input the **Abort Job**, if you want to start marking again, you have to set the **Job Number** and toggle the **Job Load** ON. In addition, if **Abort Job** turns ON not during marking execution, you still need to set the **Job Number** and toggle the **Job Load** ON

*2 After inputting **Abort Job**, the next marking process starts from the beginning of the job. (It does not continue the previous marking job.)

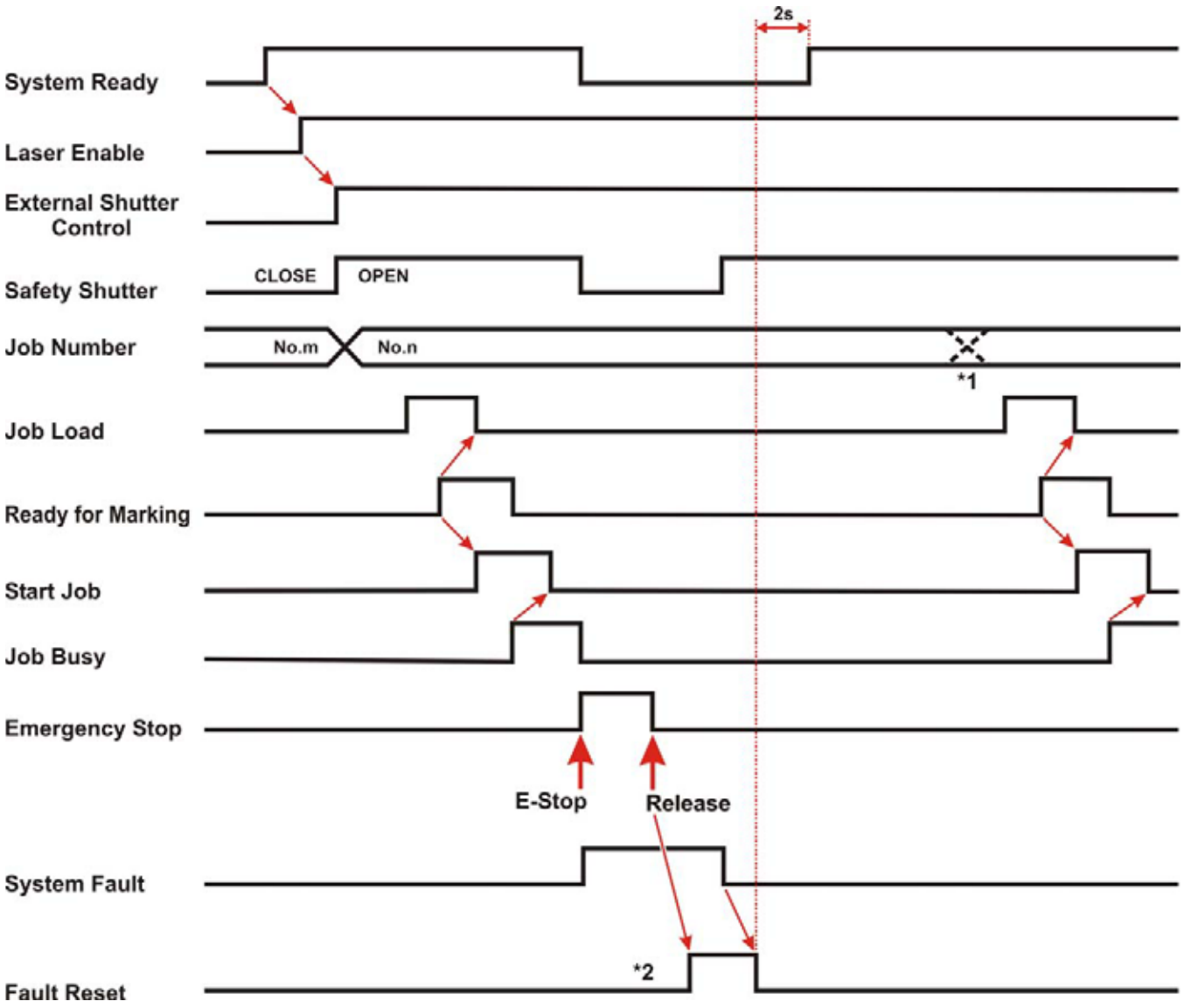
SYSTEM FAULT



*1 When the unit is in a fault condition, the **System Fault** is turned ON and the **System Ready** is turned OFF. Then when the **System Fault** is cleared using **Fault Reset,** (and Fault Reset is OFF), **System Ready** is turned ON 2 seconds later.

*2 After inputting **Fault Reset**, if you want to start marking again, you have to again set the **Job Number** and toggle **Job Load** ON.

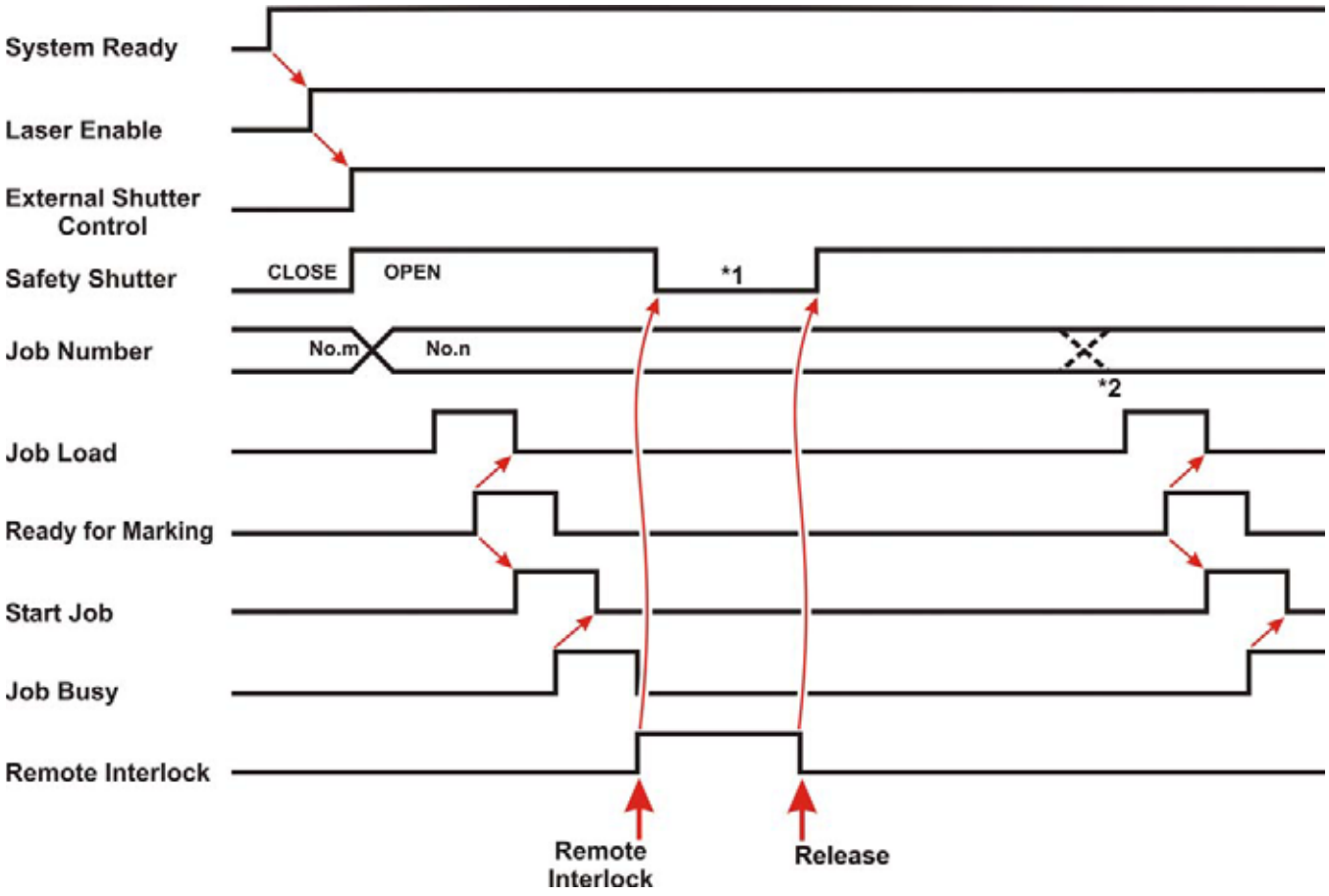
EMERGENCY STOP



*1 After the Fault Reset is turned ON, in order to restart a job you need to set the job number and to turn the **Job Load** ON.

*2 After the Emergency Stop is released and the **Fault Reset** is toggled ON, or the key switch is cycled OFF then ON, the **System Fault** turns OFF. Once the **Fault Reset** is turned OFF the **System Ready** turns ON 2 seconds later. It is necessary to re-load the job by resetting the **Job Number** and toggling **Job Load** ON briefly to continue.

REMOTE INTERLOCK



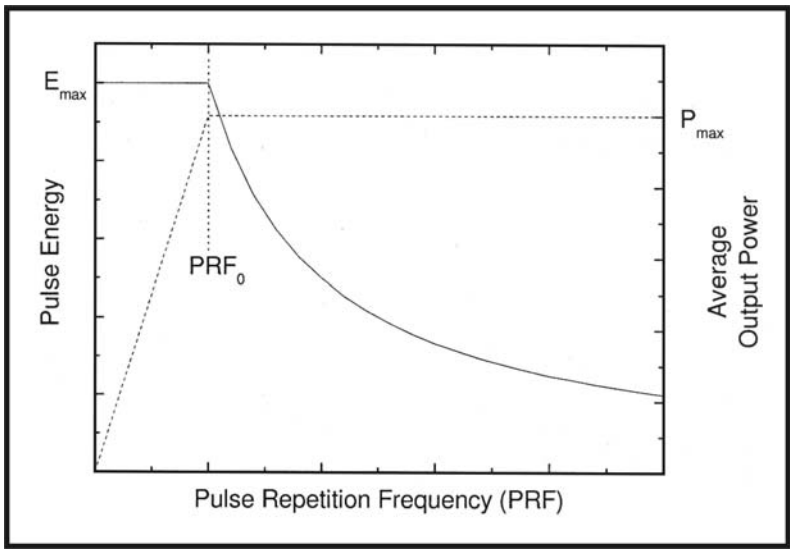
*1 The **Safety Shutter** closes while the **Remote Interlock** is open.

*2 After the **Remote Interlock** is closed, set the **Job No.** and toggle the **Job Load** to ON.

APPENDIX C
SPI PULSED FIBER LASER REFERENCE MATERIAL

Section I: Pulsed Laser Characteristics

Switching Frequency (PRF_0)



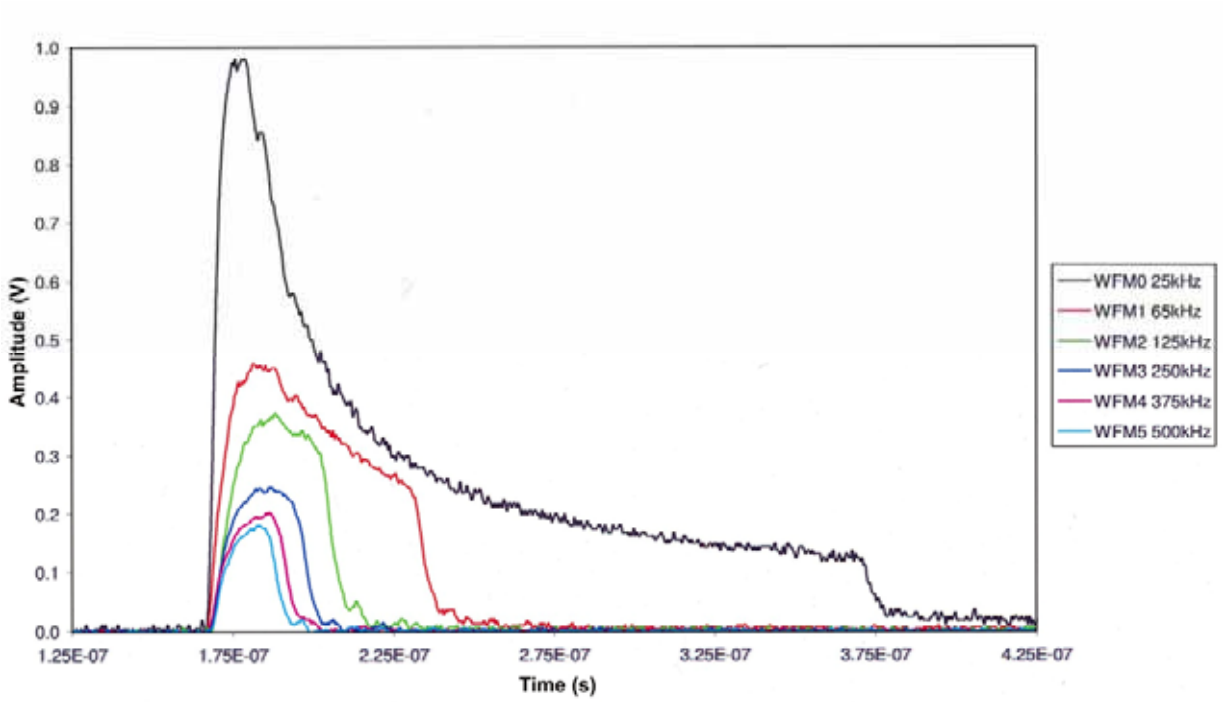
APPENDIX C: SPI PULSED FIBER LASER REFERENCE MATERIAL

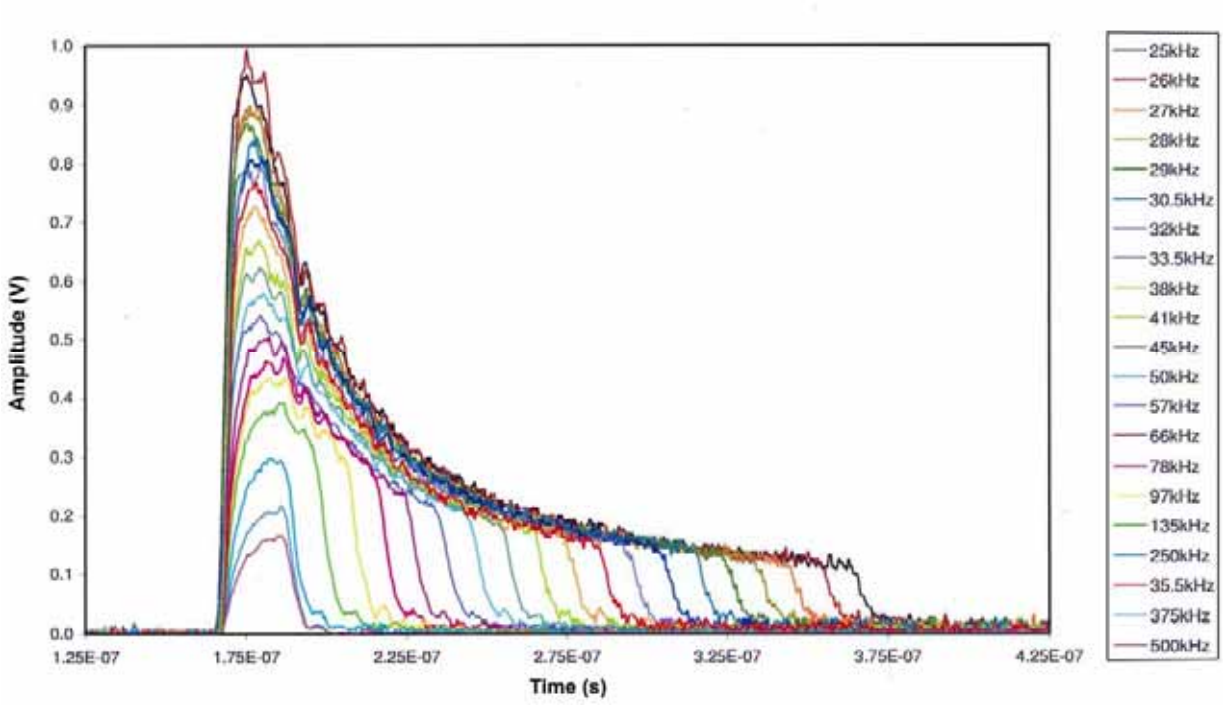
Waveform Reference Table

Waveform Number	20 W HP Lasers Electrical Duration (Approximate ns)	20W HP Lasers PRF_0 (KHZ) / Emax (mJ)	35 W Lasers Electrical Duration (Approximate ns)	35W Lasers PRF_0 (KHZ) / Emax (mJ)
0	200	25 / 0.8	250	30 / 1.33
1	65	65 / 0.31	130	47 / 0.85
2	30	125 / 0.16	60	76 / 0.53
3	15	250 / 0.08	30	145 / 0.28
4	12	375 / 0.053	20	230 / 0.17
5	9	500 / 0.04	9	250 / 0.16
6	9	500 / 0.04	9	250 / 0.16
7	9	500 / 0.04	9	250 / 0.16
8	9	500 / 0.04	9	250 / 0.16
9	9	500 / 0.04	9	250 / 0.16
10	9	500 / 0.04	9	250 / 0.16
11	200	25 / 0.8	250	30 / 1.33
12	190	26 / 0.77	230	32 / 1.25
13	180	27 / 0.74	220	34 / 1.18
14	170	28 / 0.71	200	38 / 1.05
15	160	29 / 0.69	170	41 / 0.98
16	150	30.5 / 0.66	150	43.5 / 0.92
17	140	32/ 0.63	130	47 / 0.85
18	130	33.5 / 0.6	110	51 / 0.78
19	120	35.5 / 0.56	100	54 / 0.74
20	110	38 / 0.53	90	58 / 0.69
21	100	41 / 0.49	80	62 / 0.65
22	90	45 / 0.44	70	69 / 0.58
23	80	50 / 0.4	60	76 / 0.53
24	70	57 / 0.35	50	89 / 0.45
25	60	66 / 0.3	40	108 / 0.37
26	50	78 / 0.26	30	145 / 0.28
27	40	97 / 0.21	20	230 / 0.17
28	30	135 / 0.15	9	250 / 0.16
29	20	250 / 0.08	9	250 / 0.16
30 — 63	20	250 / 0.08	9	250 / 0.16

Waveform Number	20 W SM Lasers Electrical Duration (Approximate ns)	20W SM Lasers PRF_0 (KHZ) / Emax (mJ)
0	220	35 / 0.57
1	120	55 / 0.36
2	55	90 / 0.22
3	25	170 / 0.12
4	18	270 / 0.074
5	15	290 / 0.068
6	Duplicate of 5	Duplicate of 5
7	Duplicate of 5	Duplicate of 5
8	Duplicate of 5	Duplicate of 5
9	Duplicate of 5	Duplicate of 5
10	Duplicate of 5	Duplicate of 5
11	220	35 / 0.57
12	205	37 / 0.54
13	200	39 / 0.51
14	190	44 / 0.45
15	160	48 / 0.42
16	140	51 / 0.39
17	120	55 / 0.36
18	100	60 / 0.33
19	95	63 / 0.32
20	85	68 / 0.29
21	75	72 / 0.28
22	65	80 / 0.25
23	55	90 / 0.22
24	45	105 / 0.19
25	35	125 / 0.16
26	25	170 / 0.12
27	18	270 / 0.074
28	15	290 / 0.068
29 — 63	Duplicate of 28	Duplicate of 28

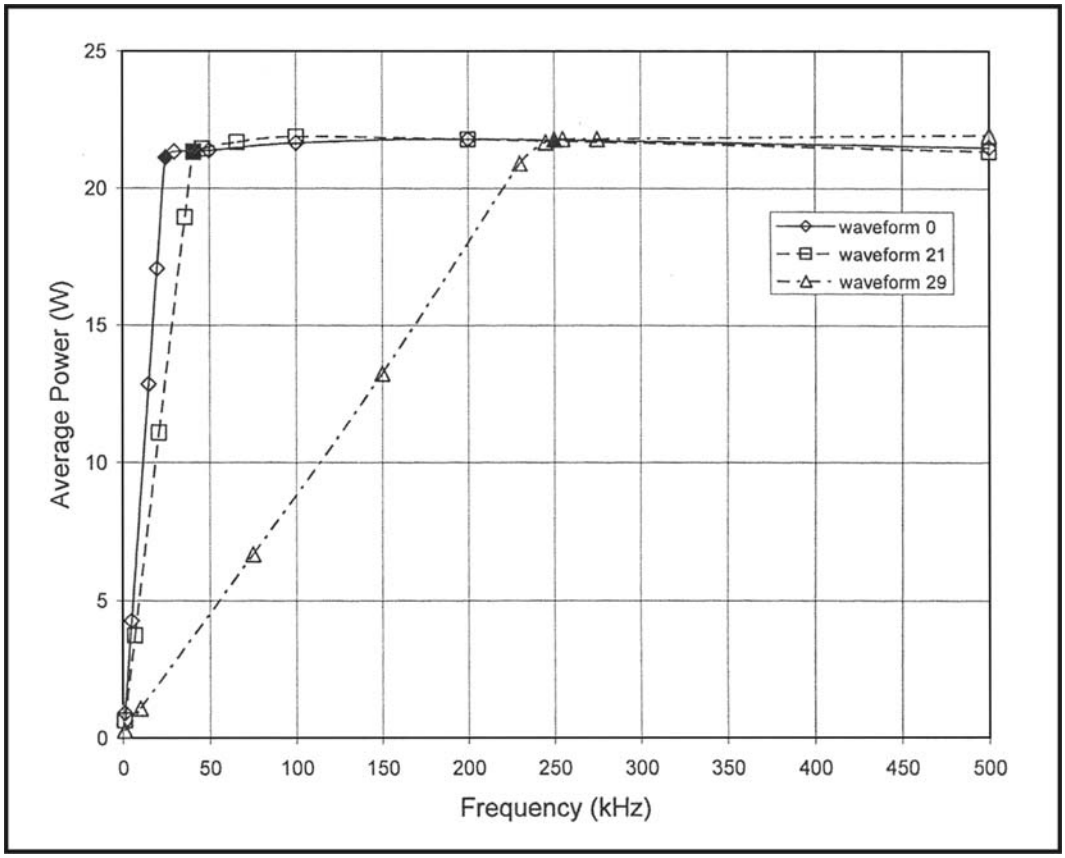
Example of 20W HP Laser Module Optical Pulse Shapes





TOP: **Waveforms 0 – 5 at PRF_0 for each waveform.**
BOTTOM: **Waveforms 11 – 29 at PRF_0 for each waveform.**

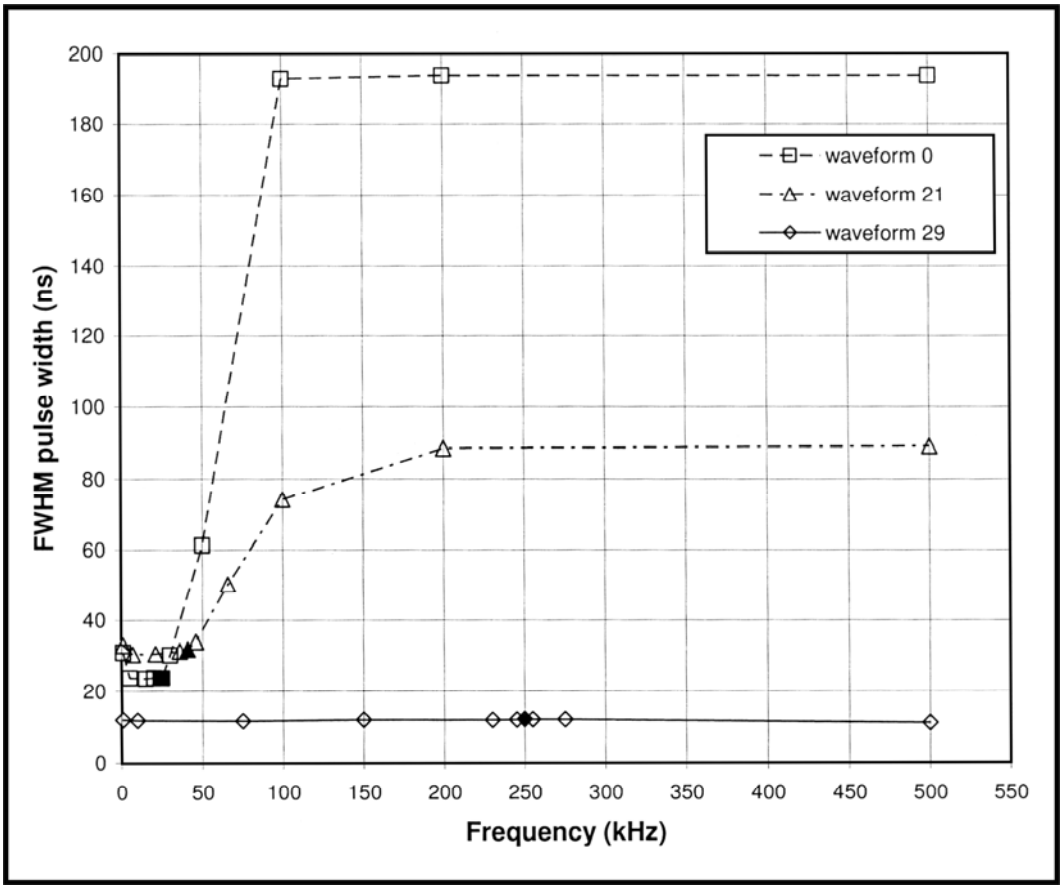
Example of 20W HP Laser Module
Average Power vs. PRF Characteristics



NOTE: Solid points indicate PRF_0.

Waveform 0: PRF_0 = 25 kHz
Waveform 21: PRF_0 = 41 kHz
Waveform 29: PRF_0 = 250 kHz

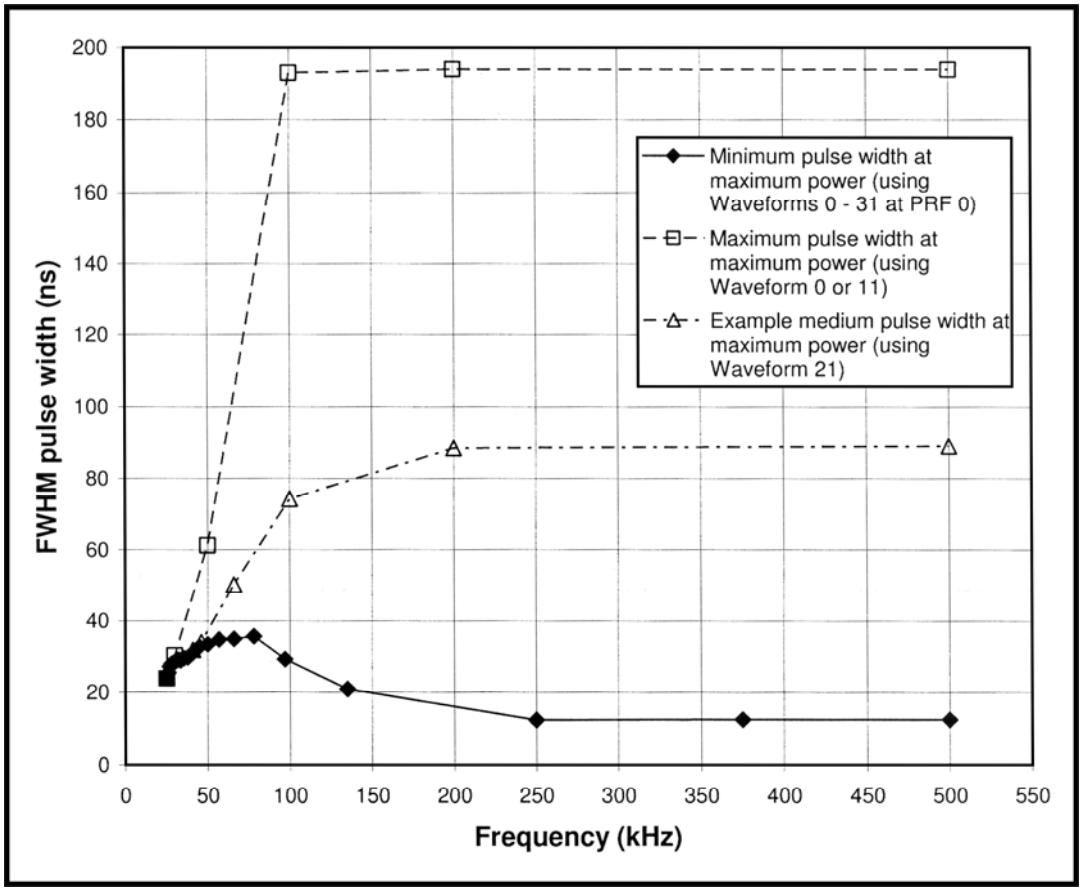
Example of 20W HP Laser Module FWHM Pulse Width vs. PRF Characteristics



NOTE: Solid points indicate PRF_0.

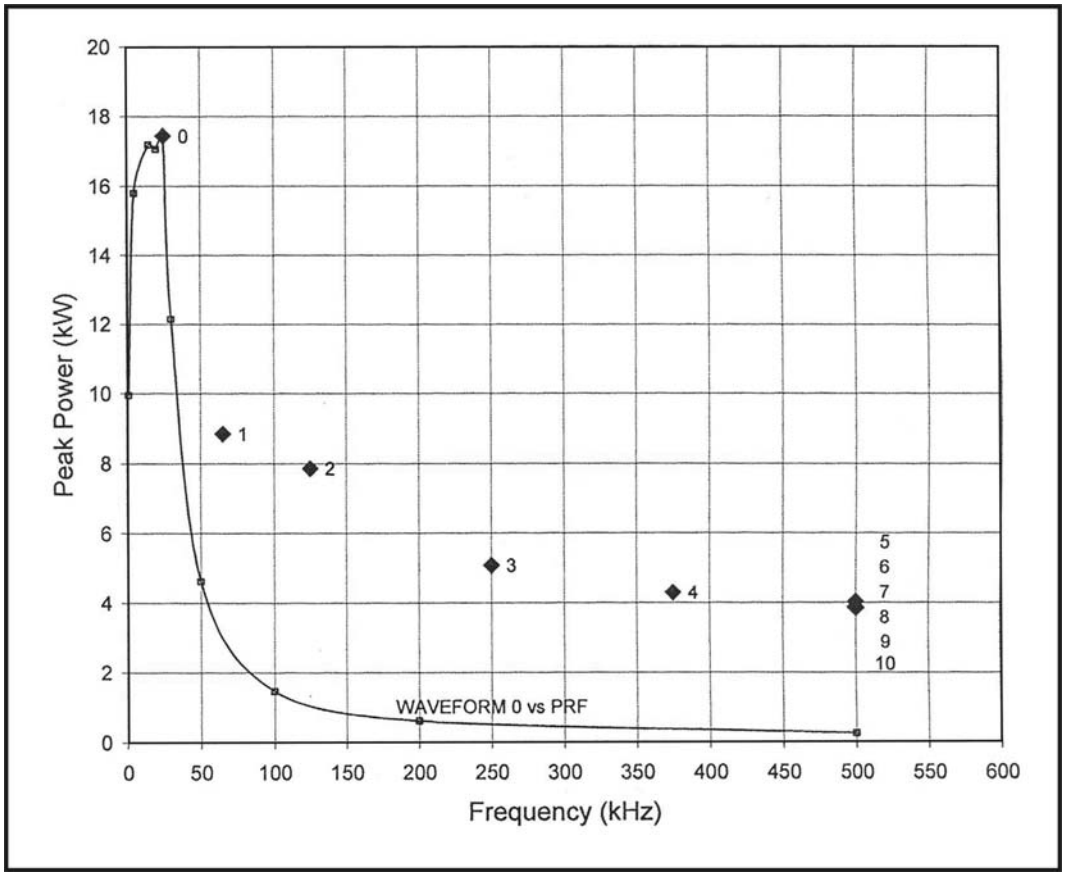
Waveform 0: PRF_0 = 25 kHz
Waveform 21: PRF_0 = 41 kHz
Waveform 29: PRF_0 = 250 kHz

Range Of Achievable FWHM Pulse Width vs. PRF at Maximum Output Power Using Waveforms at, or Above Their Specified PRF_0 Frequency



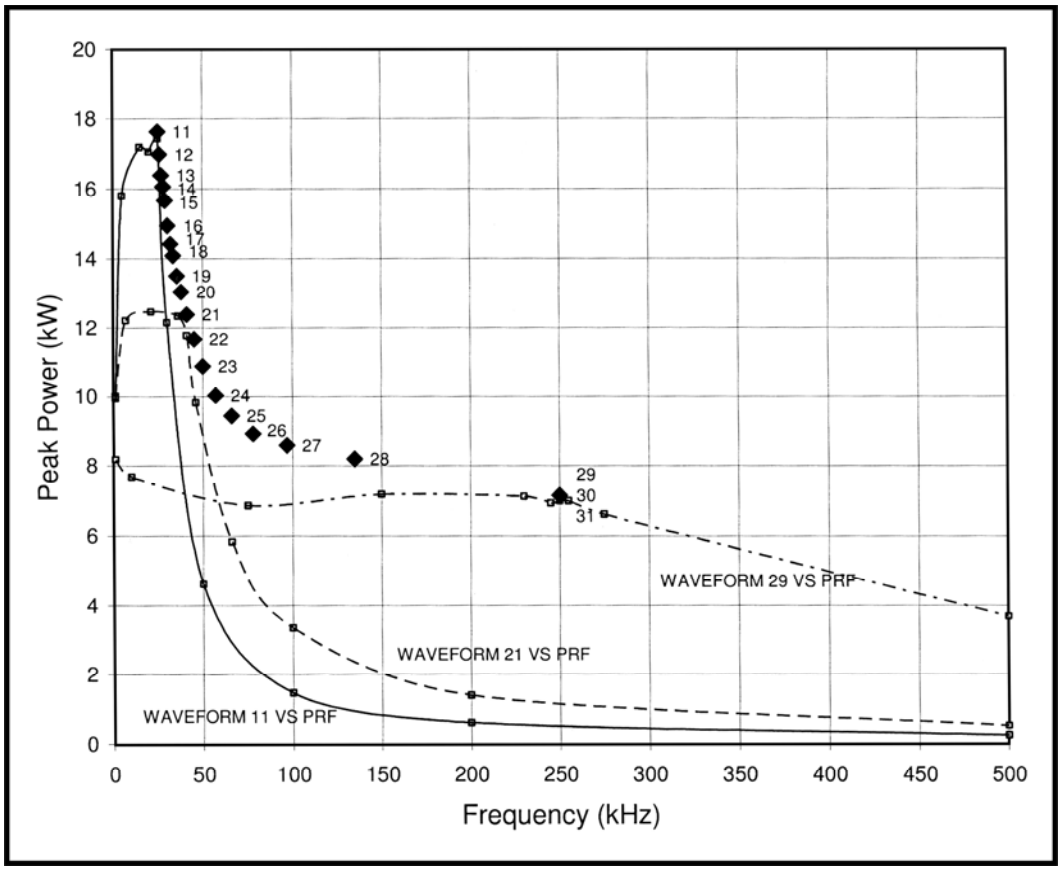
NOTE: This graph does not show operation at pulse rates below PRF_0 for any waveform.

Example of 20W HP Laser Module
Peak Power vs. PRF Characteristics (waveforms 0 – 10)



NOTE: Solid points indicate PRF_0.

Example of HP 20W Laser Module
Peak Power vs. PRF Characteristics (waveforms 11 – 31)



NOTE: Solid points indicate PRF_0.

APPENDIX D
EMBEDDED CONTROLLER REMOTE COMMAND API

Section I. Remote Command API

Remote API Commands (Numerical Listing)

Value/constant	Description
// Control commands //	
1	Abort
2	TakeHostControl
3	ReleaseHostControl
4	GetHostControlStatus
5	GetHostInControl
6	EnableBroadcasting
7	LoadHardwareDefaults
8	HardwareReset
9	GetRemoteIP
10	GetKFactor
14	SetPerformanceGlobals
15	ResetPerformanceGlobals
16	OpenCOMPort
17	CloseCOMPort
18	COMWriteLine
19	GoToZ
20	GoToXYZ
21	SetMOTFEncoderRate
22	SetMemBuffer
23	GetMemBuffer
24	GetAvailableRAM
27	COMWriteChar
29	SetUserOutBit
30	GetUserInWord

Value/constant	Description
31	GetAllIOWords
32	SetUserOutInitWord
33	GetUserOutInitWord
34	SampleMOTFEncoderCount
35	ClearMOTFEncoderCount
36	GetMOTFEncoderCount
37	Echo
38	GetLensFileList

Value/constant	Description
// Object commands //	
100	SetObjectString
101	SetObjectWaitIO
102	TransformObject
103	GetObjectRect
104	GetObjectCenter
105	GetObjectType
106	EnableObject
107	GetObjectString
108	GetObjectName
109	SetObjectUserData
110	GetObjectUserData
111	ResetObject
112	ResetUserTransform
113	TransformObjectByName
// Job commands //	
200	ClearJobList
201	MakeJobActive
202	RemoveJob
203	GetFlashJobFileList
204	GetUSBJobFileList
205	LoadFlashJob
206	LoadUSBJob
207	ExecuteJobOnce

Value/constant	Description
208	ExecuteJobContinuous
209	GetJobStatus
210	GetLastError
211	GetObjectCount
214	GetJobExecutionStatus
215	SetExternalStartMode
216	GetExternalStartMode
218	GetActiveJob
// Administration //	
500	SetAdminPIN
501	GetAdminPIN
502	SetDHCPMode
503	GetDHCPMode
504	SetLocalGateway
505	GetLocalGateway
506	SetLocalIP
507	GetLocalIP
508	SetNodeFriendlyName
509	GetNodeFriendlyName
510	SetSubnetMask
511	GetSubnetMask
512	SetUserPIN
513	GetUserPIN
514	SetCOMPortSpeed
515	GetCOMPortSpeed
516	SetCOMPortAssignments
517	GetCOMPortAssignments
518	SetLocalTime
519	GetLocalTime

Remote API Commands - Detail (Alphabetical Order)

A detail of the Remote API Commands are list alphabetically below by function name.

Abort	
Purpose:	Stops the execution of a job.
Implementation:	"1"
Parameters:	None
Returns:	An API Response Code
Comments:	Immediately stops the execution of a running job and resets the *JobRunning* status to Idle.
Also see:	N/A

ClearJobList	
Purpose:	Removes all loaded jobs from memory.
Implementation:	"200"
Parameters:	None
Returns:	An API Response code
Comments:	The *ActiveJob* is cleared when this call completes.
Also See:	**RemoveJob**, **LoadUSBJob**, **LoadFlashJob**.

ClearMOTFEncoderCount	
Purpose:	Clears the MOTF encoder to zero.
Implementation:	"35'
Parameters:	None
Returns:	An API response code.
Comments:	The ActiveJob must be *Idle* for this command to succeed. To determine ActiveJob status, use the *GetJobStatus* command.
Also see:	**SampleMOTFEncoderCount, GtMOTFEncoderCount, GetJobStatus**

CloseCOMPort	
Purpose:	Closes the specified COM port on the Marker.
Implementation:	"17"
Parameters:	None
Returns:	An API Response code.
Comments:	Client must call **OpenCOMPort** with the appropriate port number before making this call.
Also see:	**OpenCOMPort**, **COMWriteLine**, **COMWriteChar**.

COMWriteChar	
Purpose:	Writes a single character to the COM port opened with the **OpenCOMPort** command.
Implementation:	"27,timeout,char"
	Timeout: the time in milliseconds to wait for a response to the character sent to the COM port. If no response is expected, use the value 0 (zero).
	Char: The character to send to the COM port.
Returns:	*Response:* The response received from the COM connected device. If the port times out, *PortTimeout* is returned. If *portnum* is invalied, *WrongPortNumber* is returned. If a value of 0 (zero) is specified for timeout, *Success* is returned.
Comments:	Client must call *OpenCOMPort* before making this call. When using *COMWriteChar*, a single character is sent to the COM connected device, without an appended line feed. The the Marker will then wait for a single character response from the COM connected device. If the device responds with more than one character, only the first character in the response is returned. If no character is detected, the command will timeout.
Also See:	**OpenCOMPort, CloseCOMPort, COMWriteLine.**

COMWriteLine	
Purpose:	Writes a string of characters to the COM port opened with the *OpenCOMPort* command.
Implementation:	"18,timeout,string."
	timeout: The time in milliseconds to wait for a response to the string sent to the COM port. If no response is expected, use the value 0 (zero).
	string: The string of characters to send to the COM port. The Marker will append a line feed to character to the end of the string.
Returns:	*responsecode, response:* The response received from the COM connected device prepended by an API Response code. If the port times out, *PortTimeout* is returned. If *portnum* is invalid, *WrongPortNumber* is returned. If a value of 0 (zero) is specified for timeout, Success is returned.
Comments:	Client must call the *OpenCOMPort* before making this call. When using *COMWriteLine*, the string sent to the device is appended with a line feed. The the Marker will then wait for a response from the COM connected device, which must have an appended line feed. If no line feed is detected, the command will timeout.
Also see:	**OpenCOMPort**, **CloseCOMPort, COMWriteChar.**

Echo	
Purpose:	Echos a string.
Implementation:	"37,string"
Parameters:	*String*: A string to send that will encode back to the Remote API: Valid range: [3000 characters]
Returns:	*String*: The string value sent.
Comments:	This command can be used to verify communications, and for *keepalive* purposes.
Also see:	N/A

EnableBroadcasting	
Purpose:	Enables or disables the broadcast of messages by the Host.
Implementation:	"6,state"
Parameters:	state: The state of the Broadcast engine: 0 = idle 1 = broadcasting messages Valid range: [0,1]
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. Some high speed operations, especially when using the RS-232 port, may be briefly interrupted during a broadcast cycle. In these cases, the client can disable broadcasting. Note that when broadcasts are disabled, the Marker will no longer be visible on the local network, but will continue to have full API functionality.
Also see:	N/A

EnableObject	
Purpose:	Enable or disable execution of the specified object.
Implementation:	"106,objectindex,state"
Parameters:	Objectindex: The zero-based inex of the object to change. Valid rage: [0,1]
	state: The enabled state of the object: 0 = The object will not execute when the job is executed. 1 = The object will execute when the job is executed (default). Valid range: [0,1]
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. Some high speed operations, especially when using the RS-232 port, may be briefly interrupted during a broadcast cycle. In these cases, the client can disable broadcasting. Note that when broadcasts are disabled, the Marker will no longer be visible on the local network, but will continue to have full API functionality.
Also see:	**TakeHostControl, GetJobStatus.**

ExecuteJobContinuous	
Purpose:	Starts the execution of the Active job and will execute the job in an infinite loop.
Implementation:	"208,cacheobjects"
Parameters:	Cacheobjects: Flag indicating whether to cache objects in the command FIFO. 0 = do not cache objects, but wait for the *StartMark* signal. 1 = cache objects immediately. Valid range: [0,1]
Returns:	An API Response code

ExecuteJobContinuous	
Comments:	Client must call *TakeHostControl* before making this call. The *ActiveJob* must be set to a job currently loaded in RAM with a call to *MakeJobActive.* The the Marker will wait to start marking until a hardware *StartMark* signal is received, and will continue repeating this process until *Abort* is called. The *cacheobjects* flag affects how the list of marking objects is processed and sent to the command FIFO. If *chacheobjects = 0*, the system will wait for the *StartMark* signal before sending the objects in the list to the command FIFO. This option is useful when using External Control, etc. If *cacheobjects = 1*, all objects in the job are commited to the command FIFO (up to the memory capacity of the FIFO) as soon as the *ExecuteJobContinuous* commenad is received, or the previous cycle has completed. Thi option is useful when very high speed repetitive performance is required. This command returns a response immediately. To determine if the **ActiveJob** is loading, use the *GetJobStatus* command. To determine of the ActiveJob is executing, use the *GetJobExecutionStatus* command.
Also See:	**TakeHostControl**, **MakeJobActive**, **GetJobStatus, GetJobExecutionStatus**, **Abort**, **LoadFlashJob**, **LoadUSBJob**

ExecuteJobOnce	
Purpose:	Starts the execution of the ActiveJob, and runs it once without repeat.
Implementation:	"207,cacheobjects"
Parameters:	Cacheobjects: Flag indicating whether to cache objects in the command FIFO. 0 = do not cache objects, but wait for the StartMark signal. 1 = cache objects immediately. Valid range: [0,1]
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. The *ActiveJob* must be set to a job currently loaded in RAM with a call to *MakeJobActive.* The job can be stopped by calling *Abort* at any time. The *cacheobjects* flag affects how the list of marking objects is processed and sent to the command FIFO. If *cacheobjects = 0*, the system will wait for the *StartMark* signal before sending the objects in the list to the command FIFO. This option is useful when using External Control, etc. If *cacheobjects = 1*, all objects in the job are committed to the command FIFO (up to the memory capacity of the FIFO) as soon as the *ExecuteJobOnce* command is received. This option is useful when very high speed repetitive performance is required. This command returns a response immediately. To determine of the ActiveJob is loading, use the *GetJobStatus* command. To determine if the ActiveJob is executing, use the *GetJobExecutionStatus* command.
Also See:	**TakeHostControl**, **GetJobStatus**, **GetJobExecutionStatus**, **Abort**, **LoadFlashJob**, **LoadUSBJob**

GetActiveJob	
Purpose:	Gets the name and index value of the currently ActiveJob.
Implementation:	"218"
Parameters:	None
Returns:	*0,value,name* if there was no error, where: *value* = the I/O Job Selection index of the job *name* = the name of the job or *errorcode* if there was an error. Errorcode is an API response code.
Comments:	Before interacting with a job, it must be made active with the *MakeJobActive* command.
Also see:	**LoadFlashJob, LoadUSBJob, MakeJobActive**

GetAdminPIN	
Purpose:	Gets the current Administration PIN.
Implementation:	"501"
Parameters:	None
Returns:	*adminpin:* A string representing the AdminPIN.
Comments:	The *AdminPIN* is used with the Pendant interface, and provides password protection for Administrative functions.
Also see:	**SetAdminPIN**, **GetUserPIN**, **SetUserPIN**

GetAllIOWords	
Purpose:	Gets the state of all the digital inputs and outputs as two WORDS.
Implementation:	"31"
Parameters:	None
Returns:	*StandardWORD WxtendedWord:* An 18-bit WORD representing the Standard inputs and outputs followed by a 32-bit WORD representing the Extended I/O card inputs and outputs. The two WORDS are separated by a comma. *StandardWORD* bit definitions: Bit 0: User In 1 — Bit 10: User Out 1 Bit 1: User In 2 — Bit 11: User Out 2 Bit 2: User In 3 — Bit 12: User Out 3 Bit 3: User In 4 — Bit 13: User Out 4 Bit 4: Start Mark — Bit 14: Mark In Progress Bit 5: Job Load — Bit 15: Job Busy Bit 6: Interlock 1 — Bit 16: System Error Bit 7: Interlock 2 — Bit 17: Ready Bit 8: Interlock 3 Bit 9: Interlock 4 *ExtendedWORD* bit definitions: Bit 0: User In 5 — Bit 16: User Out 5 Bit 1: User In 6 — Bit 17: User Out 6 Bit 2: User In 7 — Bit 18: User Out 7 Bit 3: User In 8 — Bit 19: User Out 8 Bit 4: User In 9 — Bit 20: User Out 9 Bit 5: User In 10 — Bit 21: User Out 10 Bit 6: User In 11 — Bit 22: User Out 11 Bit 7: User In 12 — Bit 23: User Out 12 Bit 8: User In 13 — Bit 24: User Out 13 Bit 9: User In 14 — Bit 25: User Out 14 Bit 10: User In 15 — Bit 26: User Out 15 Bit 11: User In 16 — Bit 27: User Out 16 Bit 12: User In 17 — Bit 28: User Out 17 Bit 13: User In 18 — Bit 29: User Out 18 Bit 14: User In 19 — Bit 30: User Out 19 Bit 15: User In 20 — Bit 31: User Out 20 **NOTE:** Laser model numbers beginning with **8-75** support User In bits 1-4 ***only***. Model numbers beginning with **8-77** support User In bits 1-12. Bits 5-12 are defined in Appendix B as "Job Select Bit 1-8" All units support User Out bits 1-4 only.
Comments:	The ExtendWORD is valid only with the optional Extended I/O card connected. Bit 0 is defined as the LSB of the WORD. A bit value of 0 indicates the input is LOW, a value of 1 indicates the input is HIGH.
Also see:	**SetUserOutBit, GetUserInWord**

GetAvailableRAM	
Purpose:	Gets the amount of available RAM from the Windows operating system.
Numerical implementation:	"24"
Parameters:	None
Returns:	*memorycount:* The amount of available RAM as reported by the *Windows CE* operating system.
Comments:	None
Also see:	N/A

GetCOMPPortAssignments	
Purpose:	Gets the COM port to be used for the available interface.
Implementation:	"517"
Parameters:	None
Returns:	Assignments: Comma delimited string representing the assignments. The string is in three sections: Section1 = Pendant assignment Section2 = RemoteAPI assignment Section3 = Motion control assignment. Example: "1,2,3" would indicate that the Pendant is assigned to COM1, the Remote API to COM2, and Motion Control to COM3.
Comments:	Client must call *TakeHostControl* before making this call. There are three COM ports available: COM1, COM2, and COM3.
Also see:	**SetCOMPortAssignments**, **SetCOMPortSpeed, GetCOMPortSpeed**

GetCOMPPortSpeed	
Purpose:	Gets the baud rate of the specified COM port.
Implementation:	"515"
Parameters:	None
Returns:	*speeds:* A comma delimited string representing the speeds assigned to the interfaces. The string is in three sections: Section1 = Pendant port baud rate. Section2 = RemoteAPI port baud rate. Section3 = Motion control port baud rate.
Comments:	Client must call *TakeHostControl* before making this call. There are three COM ports available on the Marker: COM1, COM2, and COM3. Only the baud rate can be configured on the ports. The remaining settings are: 8 data bits No parity 1 stop bit XON/XOFF (Software handshake)
Also see:	**SetCOMPortSpeed**, **SetCOMPortAssignments, GetCOMPortAssignments**

GetDHCPMode	
Purpose:	Gets the current DHCP mode.
Implementation:	"503"
Parameters:	None
Returns:	"StaticIP": the Marker is in fixed IP address mode.
	"Autodetect": the Marker is in Dynamic IP address mode.
Comments:	If the Marker is in DHCP mode at power up, it will attempt to get an IP address from a DHCP server on the local network.
Also See:	**SetDHCPMode**

GetExternalStartMode	
Purpose:	Gets the current ExternalStart mode of the Active Job.
Implementation:	"216"
Parameters:	None
Returns:	Mode: 0 = When port is HIGH 1 = When port is LOW 2 = After transition from LOW > HIGH 3 = After transition from HIGH > LOW
Comments:	Client must call *TakeHostControl* before making this call. The External Start mode controls what type of signal transition on the Start Mark input will trigger the start of job execution.
Also see:	**SetExternalStartMode**

GetFlashJobFileList	
Purpose:	Gets a comma delimited list of all jobs stored in Flash memory
Numerical implementation:	"203"
Parameters:	None
Returns:	*joblist:* A comma delimited list of all jobs stored in Flash memory.
Comments:	The flash memory is located on the Marker.
Also see:	**GetUSBJobFileList**

GetHostControlStatus	
Purpose:	Indicates whether the Client currently has exclusive control of the Marker.
Numerical implementation:	"4"
Parameters:	None
Returns:	*InControl:* Client currently has exclusive control of the Marker.
	NotInControl: Client currently does not have exclusive control of the Marker.
Comments:	None
Also see:	**TakeHostControl**, **ReleaseHostControl**

GetHostInControl	
Purpose:	Gets the current host interface that has exclusive control of the Marker.
Numerical implementation:	"5"
Parameters:	None
Returns:	*HostPendant*: The pendant has exclusive control.
	HostLANStream: The LAN based streaming interface (WinLase LAN) has exclusive control
	HostGUI: A User Interface running on the Marker has exclusive control.
	HostBluetooth: The Bluetooth interface has exclusive control.
	HostLAN: The LAN interface has exclusive control.
	HostRS-232: The RS-232 interface has exclusive control.
	HostLocalIO: The localIO job loading Host has exclusive control.
Comments:	None
Also see:	**TakeHostControl**, **ReleaseHostControl**

GetJobExecutionStatus	
Purpose:	Gets the command execution status of the FIFO.
Numerical implementation:	"214"
Parameters:	None
Returns:	*JobIdle*: The command FIFO is not currently executing commands.
	Busy: TheActiveJob is currently loading / executing.
Comments:	Use this command to check the status of the command FIFO. If the status is *JobIdle*, the command FIFO is not currently executing commands. There may be commands being locaded into the command FIFO, however. To check the loading status, use *GetJobStatus*. If the status is *Busy*, the command FIFO is currently executing commands. It may be possible to load commands into the FIFO while it is executing. To set the *ActiveJob* to a job currently loaded into RAM, call *MakeJobActive*.
Also see:	**GetJobStatus, MakeJobActive**, **LoadUSBJob**, **LoadFlashJob, ExecuteJobOnce, ExecuteJobContinuous.**

GetJobStatus	
Purpose:	Gets the command loading status of the job engine.
Implementation:	"209"
Parameters:	None
Returns:	*JobIdle*: There are no commands waiting to be loaded into the command FIFO.
	Busy: The job ending is currently loading commands into the command FIFO.
Comments:	Use this command to check the status of the job engine, which feeds the command FIFO. If the status is *JobIdle*, the job engine is available for loading new commands into the FIFO. The FIFO may still be actively executing commands, however. To check the execution status, use *GetJobExecutionStatus*. If the status is *Busy*, the job engine is currently loading commands into the command FIFO and is not available. To set the *ActiveJob* to a job currently loaded into RAM, call *MakeJobActive*.
Also see:	**GetJobExecutionStatus, MakeJobActive, LoadUSBJob, LoadFlashJob, ExecuteJobOnce, ExecuteJobContinueous.**

GetKFactor	
Purpose:	Gets the calibration factor of the lens configuration.
Implementation:	‘‘10”
Parameters:	None
Returns:	*kfactor*: The calibration factor of the lens configuration, in field units (bits)/mm.
Comments:	Use this command to discover the conversion between real world units and field units.
Also see:	**SetAlignmentGlobals**

GetLastError	
Purpose:	Gets the last error string from the server. Reading the last error will clear it.
Numerical implementation:	“210”
Parameters:	None
Returns:	*errorstring*: The last server error represented as a string.
Comments:	An application should periodically check *GetLastError*, as some errors may be generated asynchronously after Success was returned from a previous command. For example, after a (potentially) long marking process has been started with a call to *RunJobContinuous*, an error could occur that is generated after the call to *RunJobContinuous* returns.
Also see:	N/A

GetLensFileList	
Purpose:	Gets a comma delimited list of all lens (correction table) files stored in Flash memory
Implementation:	"38"
Parameters:	None
Returns:	*lenslist:* A comma delimited list of all lens (correction table) files stored in Flash.
Comments:	Flash memory is the LEC-1 internal storage memory.
Also see:	**N/A**

GetLocalGateway	
Purpose:	Gets the default local Gateway address for the Marker.
Implementation:	“505”
Parameters:	None
Returns:	*gateway*: The Gateway address in dot notation format (ex. 192.168.42.1)
Comments:	None
Also see:	**SetLocalGateway**

GetLocalIP	
Purpose:	Gets the local IP address for the Marker.
Implementation:	“507”
Parameters:	None
Returns:	*ipaddress*: The IP address in dot notation format (ex. 192.168.42.1)
Comments:	None
Also see:	**SetLocalIP**

GetLocalTime	
Purpose:	Gets the current local time and date.
Implementation:	"519"
Parameters:	None
Returns:	*localtime*: A comma delimited string containing the local time data. The string is in this format: *year,monhy,day,hour,minute,second*
Comments:	None
Also see:	**SetLocalTime**

GetMemBuffer	
Purpose:	Gets the internal memory buffer at the specified index.
Implementation:	"23,bufferindex"
Parameters:	*bufferindex*: The index of the memory buffer to retrieve. Valid range: [1 to 10]
Returns:	*bufferstring*: The contents of the specified memory buffer
Comments:	A string based object (barcodes and text) can be configured to retrieve the next string to mark from an internal memory buffer at execution time. There are 10 buffers available. The string object must be configured to use these memory buffers.
Also see:	**SetMemBuffer**

GetMOTFEncoderCount	
Purpose:	Gets the current value of the MOTF encoder counter.
Implementation:	"36"
Parameters:	None
Returns:	*0,count* if there was no error, or *errorcode* if there was an error. *errorcode* is an API response code.
Comments:	The value count is in units of field bits. The ActiveJob must be *Idle* for this command to succeed. To determine ActiveJob status, use the *GetJobStatus* command.
Also see:	**GetJobStatus, SampleMOTFEncoderCount, ClearMOTFEncoderCount**

GetNodeFriendlyName	
Purpose:	Gets the local IP address for the Marker.
Numerical implementation:	"509"
Parameters:	None
Returns:	*name*: The name of the Marker. This name is used in the Marker broadcasts.
Comments:	None
Also see:	**SetNodeFriendlyName**

GetObjectCenter	
Purpose:	Gets the geometric center of the specified object in field units.
Numerical implementation:	"104,objectindex"
Parameters:	objectindex: The zero-based index of the object. Valid range: 0 to (object count — 1)
Returns:	x,y where: x = the x coordinate of the object center, in bits y = they coordinate of the object center, in bits
Comments:	Client must call *TakeHlostControl* before making this call. The Object at *objectindex* must be a valid marking object type. The ActiveJob must be *Idle* for this command to succeed. To determine ActiveJob status, use the *GetJobStatus* command. The *marking field* is described using a Cartesian coordinate system, with (0,0) at the center of the field, (-32768, -32768) at the bottom left comer, and (32767, 32767) at the top right comer. The field scale factor can be discovered by calling *GetKFactor*.
Also see:	**GctObjectR,Rect, TransformBoject, GetObjectCount, GetObjectType.**

GetObjectCount	
Purpose:	Gets the number of objects in the *ActiveJob*.
Numerical implementation:	“2l1"
Parameters:	None
Returns:	*0,objectcount* if there was no error, or *errorcode* if there was an error. *errorcode* is an API response code.
Comments:	Client must call *TakeHostControl* before making this call. There must be an *ActiveJob.*
Also see:	**GetObjectType**

GetObjectName	
Purpose:	Gets the name of the specified object.
Implementation:	"108,objectindex"
Parameters:	*objectindex:* The zero-based index of the object. Valid range: [0 to (object count - 1)]
Returns:	*O,objectname* if there was no error, or *errorcode* if there was an error. *errocode* is an API response code.
Comments:	Client must call *TakeHostControl* before making this call.
Also see:	N/A

GetObjectString	
Purpose:	Gets the string value of the specified text object.
Implementation:	“107,objectindex”"
Parameters:	*objectindex:* The zero-based index of the object. Valid range: 0 to (object count - 1)
Returns:	*0,objectstring* if there was no error, or *errorcode* if there was an error. *errocode* is an API response code.
Comments:	Client must call *TakeHostControl* before making this call. The specified object must be a DynamicText object.
Also see:	**SetObjectString, GetObjectType**

GetObjectType	
Purpose:	Gets the type of the specified object.
Implementation:	"105,objectindex"
Parameters:	*objectindex:* The zero-based index of the object. Valid range: [0 to (object count - 1)]
Returns:	*0,objecttype* if there was no error, or *errorcode* if there was an error. *errocode* is an API response code. *objecttype* is the type of the specified object: 0 = StaticMark 1 = Dynamic Barcode 2 = DynamicText 3 = Alignment 4 = SerialPort 5 = SmartMotion
Comments:	Client must call *TakeHostControl* before making this call. There must be an *ActiveJob*.
Also see:	**GetObjectCount**

GetObjectRect	
Purpose:	Gets the bounding rectangle for the specified object in field units.
Implementation:	"103,objectindex"
Parameters:	*objectindex:* The zero-based index of the object. Valid range: 0 to (object count - 1)]
Returns:	*left, top, right, bottom* where: *left* = the leftmost bounds of the vectorlist, in bits *top* = the topmost bounds of the vectorlist, in bits *right* = the rightmost bounds of the vectorlist, in bits *bottom* = the bottommost bounds of the vectorlist, in bits
Comments:	Client must call *TakeHostControl* before making this call. The Object at *objectindex* must be a valid marking object type. The *ActiveJob* must be *Idle* for this command to succeed. To determine *ActiveJob* status, use the *GetJobStatus* command. The *marking field* is described using a Cartesian coordinate system, with (0,0) at the center of the field, (-32768, -32768) at the bottom left corner, and (32767, 32767) at the top right corner. The field scale factor can be discovered by calling *GetKFactor*.
Also see:	**GetObjectCenter, TransformObject, GetObjectCount, GetObjectType**

GetObjectUserData	
Purpose:	Gets the specified User Data stored by the specified object.
Implementation:	"110,objectindex,dataindex"
Parameters:	*objectindex:* The zero-based index of the object. Valid range: [0 to (object count - 1)]. dataindex: The zero-based index of the data. Valid range: [0 and 1]
Returns:	*0,datastring* if there was no error, or *errorcode* if there was an error. *errocode* is an API response code.
Comments:	Client must call *TakeHostConirol* before making this call. All objects have two string based data buffers (with index 0 and 1) available for the programmer to use for any reason. The maximum size of the buffers is 256. The data buffers are cleared when the object is first loaded, and are *not* persistent between job loads. The data contained in the buffers is not used by the object.
Also see:	**SetObjectUserData**

GetRemoteIP	
Purpose:	Gets the IP address of the Client that has exclusive control of the Marker.
Implementation:	"9"
Parameters:	None
Returns:	*remotelP:* The remote IP address in dot notation format ex. 192.168.42.1
Comments:	This value is only valid if the Client that has exclusive control is using a TCP/IP connection. All other host interfaces will report "0.0.0.0".
Also see:	N/A

GetSubnetMask	
Purpose:	Gets the Subnet mask of the Marker.
Implementation:	"511"
Parameters:	None
Returns:	*subnetmask:* The Subnet mask in dot notation format ex. 255.255.255.0
Comments:	None
Also see:	**SetSubnetMask**

GetUSBJobFileList	
Purpose:	Gets a comma delimited list of all jobs stored in a USB drive.
Implementation:	"204"
Parameters:	None
Returns:	*joblist: A* comma delimited list of all jobs stored in USB drive. If no USB drive is found, *NoDrive* is returned.
Comments:	Please check with the factory for tested and approved USB devices.
Also see:	**GetFlashJohFileList**

GetUserInWord	
Purpose:	Gets the current User In input port as a 20 bit WORD.
Implementation:	"30"
Parameters:	None
Returns:	*WORD:* A 20 bit WORD representing the state of the User In 1 - User In 20 inputs.
Comments:	Bits *5* - 20 are available only with the optional Extended 1/0 card. The LSB of the WORD indicates User In 1. A bit value of 0 indicates the input is LOW, a value of 1 indicates the input is HIGH.
Also see:	**SetUserOutBit, GetAllIOWords, GetUserOutInitWord**

GetUserOutInitWord	
Purpose:	Gets the User Out WORD used to set the User Out outputs at system start up and after an Abort.
Implementation:	"33"
Parameters:	None
Returns:	*WORD*: A 20 bit WORD representing the state of the User Out ports at startup and after an Abort.
Comments:	Bits 5 – 20 are available only with the optional Extended I/O card. The LSB of the WORD indicates User Out 1. A bit value of 0 indicates the output will be set LOW, a value of 1 indicates the output will be set HIGH.
Also see:	**SetUserOutBit, GetALLIOWords, GetUserInWord**

GetUserPIN	
Purpose:	Gets the current User PIN.
Implementation:	"513"
Parameters:	None
Returns:	*userpin:* A string representing the UserPIN.
Comments:	The *UserPIN* is used with the Pendant interface, and provides password protection for User access functions.
Also see:	**SetUserPIN, GetAdminPIN, SetAdminPiN**

GoToXYZ	
Purpose:	Commands the x, y and z-axes to jump to the specified coordinate, at the specified jump speed, and inserts the specified jump delay after the jump.
Implementation:	"20,xcoordinate,ycoordinate,zcoordinate,jumpdelay"
Parameters:	*xcoordinate:* The coordinate location, in bits, to jump to. Valid range: [-32768 to 32767]
	ycoordinate: The coordinate location, in bits, to jump to. Valid range: [-32768 to 32767]
	zcoordinate: The coordinate location, in bits, to jump to. Valid range: [-32768 to 32767]
	jumpspeed: The speed, in bits/millisecond, at which the jump is executed. Valid range: [1 to 65535]
	jumpdelay: The time, in microseconds, after the jump is executed. Valid range: [1 to 65535]
Returns:	An API Response code
Comments:	***GoToXYZ*** is a Control Command, and the instructions are placed in the FIFO buffer immediately.
Also see:	**GoToZ**

GoToZ	
Purpose:	Commands the z-axis to jump to the specified coordinate, and inserts the specified jump delay after the jump.
Implementation:	" 19,zcoordinate,jumpdelay"
Parameters:	*zcoordinate:* The coordinate location, in bits, to jump to. Valid range: [-32768 to 32767]
	jumpdelay: The time, in microseconds, to delay after the jump is executed. Valid range: [1 to 65535]
Returns:	An API Response code
Comments:	*GoToZ* is a Control Command, and the instructions are placed in the FIFO immediately.
Also see:	**GoToXYZ**

HardwareReset	
Purpose:	Resets the Marker
Implementation:	''8''
Parameters:	None
Returns:	An API Response code
Comments:	After receiving this command, the Marker will perform a soft reset. Any changes made to the IP address parameters will be applied at this time. If you are using a TCP/IP based interface, the socket connection will be closed before the soft reset, and you will have to reconnect after the reset.
Also see:	N/A

LoadFlashJob	
Purpose:	Loads a job from flash memory into RAM.
Implementation:	"205,jobname"
Availability:	LAN, RS-232
Parameters:	*jobname:* The job file name in flash to load. Use the filename with the extension (ex. *Circle.dat*). There can be multiple jobs loaded in RAM simultaneously.
Returns:	An API Response code
Comments:	Before interacting with a job, it must be made active with the *MakeJobActive* command.
Also see:	**GetFlashJobList**, **ExecuteJobContinuous**, **ExecuteJobOnce**, **MakeJobActive**

LoadHardwareDefaults	
Purpose:	Loads the currently configured laser, lens and controller parameters from Flash.
Implementation:	"7"
Parameters:	None
Returns:	An API Response code
Comments:	The *laser*, *lens* and *controller* parameters are stored in non-volatile memory on the Marker. This command will reload the parameter sets from the configuration files in non-volatile memory and may take up to 20 seconds to complete.
Also see:	N/A

LoadUSBJob	
Purpose:	Loads a job from the USB drive into RAM.
Implementation:	"206.jobname"
Parameters:	*jobname*: The job file name on the USB drive to load. Use the filename and extension (ex. *Circle.dat*"). There can be multiple jobs loaded in RAM simultaneously. The job must be located on the USB drive at the following path: \LEC\Jobs.
Returns:	An API Response code
Comments:	Before interacting with a job, it must be made active with the *MakeJobActive* command.
Also see:	**GetUSBJobFileList**, **ExecuteJobContinuous**, **ExecuteJobOnce**, **MakeJobActive**

MakeJobActive	
Purpose:	Sets a job loaded into RAM as the *ActiveJob*.
Implementation:	"201,jobname"
Parameters:	*jobname:* The job file name to make active. Use the filename and extension (ex. *Circle.dat*).
Returns:	An API Response code.
Comments:	Before interacting with a job, it must be made active with the *MakeJobActive* command.
Also see:	**LoadFlashJob**, **LoadUSBJob**

OpenCOMPort	
Purpose:	Opens the specified COM port on the Marker.
Implementation:	"16,portnum,baudrate"
Parameters:	*portnum:* The COM port to open. Only COM2 is supported at this time. Valid range: [1,2,3]
	baudrate: The port baud rate. Valid range: [110, 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 128000, 256000]
Returns:	An API Response code
Comments:	The COM port is opened with the specified baud rate, 8 data data bits, 1 stop bit, no parity, and software flow control.
Also see:	**CloseCOMPort**, **COMWriteLine, COMWriteChar**

ReleaseHostControl	
Purpose:	Release the exclusive control of the Marker back to the *LANStream* host.
Implementation:	"3"
Parameters:	None
Returns:	An API Response code
Comments:	When any Client releases exclusive control, the *LANStream* host is given exclusive control. The *LANStream* host is the streaming interface used when connected to the device with the WinLase LAN.
Also see:	**TakeHostControl**

RemoveJob	
Purpose:	Deletes the Active Job from memory..
Implementation:	"202"
Parameters:	None
Returns:	An API Response code
Comments:	The *ActiveJob* is cleared after this call completes. To set the Active job, call *MakeJobActive.*
Also see:	**LoadFlashJob**, **LoadUSBJob**, **MakeJobActive**

ResetObject	
Purpose:	Restores the object vector lists and object transforms to the state they were in when the object first loaded from the job.
Implementation:	"111,objectindex"
Parameters:	*objectindex:* The zero-based index of the object to change. Valid range: [0 to (object count -1)]
Returns:	An API Response code.
Comments:	Client must call *TakeHostControl* before making this call. The ActiveJob must be Idle for this command to succeed. To determine ActiveJob status, use the *GetJobStatus* command.
Also see:	**TakeHostControl, GetJobStatus, TransformObject, SetObjectString**

ResetPerformanceGlobals	
Purpose:	Resets the current Performance global values to their defaults.
Numerical implementation:	"15"
Parameters:	None
Returns:	An API Response code
Comments:	All marking objects have a global performance data structure applied before marking. Calling *ResetPerformanceGlobals* resets the performance data structure to default values.
Also see:	**SetPerformanceGlobals**

ResetUserTransform	
Purpose:	Resets the *UserTransform* to the Identity matrix.
Implementation:	"112,objectindex"
Parameters:	Objectindex: The zero-based index of the object to change. Valid range: [0 to (object count-1)]
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. The ActiveJob must be *Idle* for this command to succeed. To determine ActiveJob status, use the *GetJobStatus* command.
Also see:	**TransformObject, ResetObject**

SampleMOTFEncoderCount	
Purpose:	Clears the MOTF encoder counter to zero, waits for the sample period, then reads the encoder count.
Implementation:	"34,sampleperiod"
Parameters:	*sampleperiod:* The time, in ms, to wait after clearing the counter to zero before reading the counter value.
Returns:	*0,count* if there was no error, or *errorcode* if there was an error. *Errorcode* is an API response code.
Comments:	The value count is in units of field bits. The ActiveJob must be Idle for this command to succeed. To determine ActiveJob status, use the *GetJobStatus* command.
Also see:	**ClearMOTFEncoderCount, GetMOTFEncoderCount, GetJobStatus**

SetAdminPIN	
Purpose:	Sets the AdminPIN
Implementation:	"500,pin"
Parameters:	*pin:* The numeric based Administration password.
Returns:	An API Response code
Comments:	The *AdminPIN* is used with the Pendant interface, and provides password protection for Administrative functions.
Also see:	**GetAdminPIN**, **SetUserPIN**, **GetUserPIN**

SetCOMPortAssignments	
Purpose:	Sets the COM port to be used for the Pendant interface and for the RS-232 Host.
Implementation:	"516,pendant,remoteapi,motioncontrol"
	Pendant: The COM port to used for the Pendant interface. Valid range: [1,2,3]
	remoteapi: The COM port to used for the RemoteAPI interface. Valid range: [1,2,3]
	motioncontrol: The COM port to used for the Motion Control interface. Valid range: [1,2,3]
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. This setting will not take effect until the board is power cycled, or a *HardwareReset* command is issued. There are three COM ports available on the Marker: COM1, COM2, and COM3. When issuing this command, you must assign a different COM port to each interface, or the command will fail.
Also see:	**GetCOMPortAssignments, SetCOMPortSpeed, GetCOMPortSpeed**

SetCOMPortSpeed	
Purpose:	Sets the baud rate of the specified COM port.
Implementation:	"514,pendantspeed,apispeed,motionspeed"
Parameters:	*pendantspeed:* The desired baud rate of the COM port that the pendant is assigned to. Valid values: [110, 300, 600, 1200, 2400, 9600, 14400, 19200, 38400, 56000, 57600, 115200]
	apispeed: The desired baud rate of the COM port that the Remote API is assigned to. Valid values: [110, 300, 600, 1200, 2400, 9600, 14400, 19200, 38400, 56000, 57600, 115200]
	motionspeed: The desired baud rate of the COM port that the Motion Controller is assigned to. Valid values: [110, 300, 600, 1200, 2400, 9600, 14400, 19200, 38400, 56000, 57600, 115200]
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. This setting will not take effect until the board power is cycled, or a *HardwareReset* command is issued. There are three COM ports available on the Marker: COM1, COM2, and COM3. Only the baud rate can be configured on the ports. The remaining settings are: 8 data bits No parity 1 stop bit XON/XOFF (Software handshake)
Also see:	**GetCOMPortSpeed**, **SetCOMPortAssignments, GetCOMPortAssignments**

SetDHCPMode	
Purpose:	Sets the current DHCP mode.
Implementation:	"502,mode"
Parameters:	mode: *Static* or *Autodetect*
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. This setting will not take effect until the board is power cycled, or a *HardwareReset* command is issued. If the Marker is in DHCP mode at power up, it will attempt to get an IP address from a DHCP server on the local network.
Also see:	**TakeHostControl**, **HardwareReset**, **GetDHCPMode**

SetExternalStartMode	
Purpose:	Sets the current ExternalStart mode of the ActiveJob.
Implementation:	"215, mode"
Parameters:	Mode: 0 = When port is HIGH 1 = When port is LOW 2 = After transition from LOW > HIGH 3 = After transition from HIGH > LOW
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. The ActiveJob must be *Idle* for this command to succeed. To determine ActiveJob status, use the *GetJobStatus* command. The External Start mode controls what type of signal transition on the Start Mark input will trigger the start of job execution.
Also see:	**GetExternalStartMode**

SetLocalGateway	
Purpose:	Sets the default local Gateway address for the Marker.
Implementation:	"504,gateway"
Parameters:	*gateway:* The Gateway address in dot notation format (ex. 192.168.42.1)
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. This setting will not take effect until the board power is cycled, or a *HardwareReset* command is issued.
Also see:	**TakeHostControl**, **HardwareReset**, **GetLocalGateway**

SetLocalIP	
Purpose:	Set the IP address of the Marker
Implementation:	"506,ipaddress"
Parameters:	*ipaddress:* The local IP address, in dot notation format (ex. 192.168.42.1)
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. This setting will not take effect until the board power is cycled, or a *HardwareReset* command is issued.
Also see:	**TakeHostControl**, **HardwareReset**

SetLocalTime	
Purpose:	Sets the current local time and date.
Implementation:	"518,year,month,day,hour,minute,second"
Parameters:	*year*: Specifies the current year.
	month: Specifies the current month. January = 1, February = 2, and so on.
	day: Specifies the day of the month.
	hour: Specifies the current hour in 24 hour time format.
	minute: Specifies the current minute.
	second: Specifies the current second.
Returns:	An API Response code
Comments:	None
Also see:	**GetLocalTime**

SetMemBuffer	
Purpose:	Sets the internal memory buffer at the specified index.
Numerical implementation:	"22,bufferindex,newstring"
Parameters:	*bufferindex*: The index of the memory buffer to set. Valid range: [1 to 10]
	newstring: The string to save in the specified memory buffer. Valid size: [1 to 2999 characters]
Returns:	An API Response code
Comments:	A string based object (barcodes and text) can be configured to retrieve the next string to mark from an internal memory buffer at execution time. There are 10 buffers available.
Also see:	**GetMemBuffer**

SetMOTFEncoderRate	
Purpose:	Set the current MOTF encoder rate in the *ActiveJob*, which will take effect the next time a job is executed.
Implementation:	"21,<rate>"
Parameters:	*rate*: The encoder rate, in bits/count. When the Marker is configured for simulated encoder mode, a fixed encoder pulse rate of 1 MHz is used, so the encode rate is the distance the part moves (in field bits) through the field during 1 clock cycle (1/1,000,000 second). When using an actual encoder, the encoder rate equals the distance the part moves (in field bits) through the field during one encoder pulse. Example with simulated encoder mode: Line speed = 100 mm/s KFactor = 570 bits/mm Encoder pulse rate 1,000,000 counts/s ((Line speed) x (KFactor)) / 1,000,000 = 0.057 counts/bit Valid range: [-32768.0 to 32767.0] Default: [0] (no MOTF)
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call, and the *ActiveJob* must be a job that has MOTF enabled. The Marker must have a valid Advanced (MOTF) license.
Also see:	**TakeHostControl**, **MakeJobActive**, **GetKFactor**

SetNodeFriendlyName	
Purpose:	Set the name of the Marker
Implementation:	"508,name"
Parameters:	*name*: The name of the Marker. This name is used in the Marker broadcasts.
Returns:	An API Response code.
Comments:	Client must call *TakeHostControl* before making this call.
Also see:	**TakeHostControl**

SetObjectString	
Purpose:	Set the string value of a string based marking object contained in the *ActiveJob*.
Implementation:	"100,objectindex,newstring"
Parameters:	*objectindex:* The zero-based index of the object to change. Valid size: [1 to 2999 characters]
	newstring: The new string value. *Valid size:* [1 to 2999 characters] Special Conditions: *DataMatrix*: To embed control characters in the string, use the tilde (~) character before the control code. To embed an actual ~ character, use two tilde characters in a row (~~). To imbed an ASCII 0 character, use ~@ instead of the ASCII 0.
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. The Object at *objectindex* must be a valid string based object, such as a text or barcode object. The *ActiveJob* must be *Idle* for this command to succeed. To determine *ActiveJob* status, use the *GetJobStatus* command.
Also see:	**TakeHostControl**, **GetJobStatus**

SetObjectUserData	
Purpose:	Set the specified User Data stored by the specified object.
Implementation:	"109,objectindex,dataindex,datastring"
Parameters:	*objectindex:* The zero-based index of the object to change. Valid range: [0 to (object count – 1)].
	datatindex: The zero-based index of the data. Valid range: [0 and 1]
	datastring: The string to store in the specified data buffer. Valid size: [1 to 255 characters]
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. All objects have two string-based data buffers (with index 0 and 1) available for the programmer to use for any reason. The maximum size of the buffers is 256. The data buffers are cleared when the object is first loaded, and are ***not*** persistent between job loads. The data contained in the buffers is not used by the object.
Also see:	**GetObjectUserData**

SetPerformanceGlobals	
Purpose:	Sets the current Performance global data structure values.
Implementation:	"l4,markspeed,laserpower,pulsewidth,period,orientation,xoffset,yoffset,zoffset"
Parameters:	*markspeed*: Scale factor for the mark speed. Valid range: [0.500 to 1.500] Default: [1.000]
	laserpower: Scale factor for the laser power. Valid range: [0.800 to 1.200] Default: [1.000]
	pulsewidth: Scale factor for the laser modulation signal pulse width. Valid range: [0.500 to 1.500] Default: [1.000]
	period: Scale factor for the laser modulation signal period. Valid range: [0.500 to 1.500] Default: [1.000]
	orientation: Orientation of the scan head to the marking field. Valid values: [0, 90, 180, 270] Default: [0]
	xoffset: Offset of the x-coordinate of all marking objects, in bits. Valid values: [-32768 to 32767] Default: [0]
	yoffset: Offset of the y-coordinate of all marking objects, in bits. Valid values: [-32768 to 32767] Default: [0]
	zoffset: Offset of the z-coordinate of all marking objects, in bits. Valid values: [-32768 to 32767] Default: [0]
Returns:	An API Response code
Comments:	All marking objects have a global performance "adjustment" applied before marking. Use the *SetPerformanceGlobals* function to set the scalar values that are applied to the specified parameters. Not all parameters need to be set at the same time. To leave the current setting of a specific parameter unchanged, use the string "NOP" for that parameter. For example, to only set the pulsewidth scalar to 1.2, use the following command: *SetPerformanceGlobals,NOP,NOP,1.2,NOP,NOP,0,0,0.* Note that the x, y and z offsets must be provided.
Also see:	**ResetPerformanceGlobals**

SetSubnetMask	
Purpose:	Sets the Subnet mask of the Marker.
Implementation:	"510,subnetmask"
Parameters:	*subnetmask:* The Subnet mask in dot notation format (ex. 255.255.255.0)
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. This setting will not take effect until the board power is cycled, or a *HardwareReset* command is issued.
Also see:	**TakeHostControl**, **HardwareReset**, **SetSubnetMask**

SetUserOutBit	
Purpose:	Sets the specified User Out output..
Implementation:	"29,bitnum,value"
Parameters:	*bitnum:* The UserOut output to set, which ranges from User Out 1 – User Out 20. Valid range: [1 to 20].
	value: The logic level to set the output to. Valid range: [0 or 1].
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call and the *ActiveJob* must be *Idle* for this command to succeed. To determine *ActiveJob* status, use the *GetJobStatus* command. **NOTE:** Laser model numbers beginning with **8-75** support User In bits 1-4 ***only***. Model numbers beginning with **8-77** support User In bits 1-12. Bits 5-12 are defined in Appendix B as "Job Select Bit 1-8" All units support User Out bits 1-4 only. When value is 0, the output will be set LOW, when value is 1 the output will be set HIGH.
Also see:	**TakeHostControl**, **GetJobStatus, GetUserInWord, GetALLIOWords**

SetUserOutInitWord	
Purpose:	Set the User Out output WORD that will be used at system start up and after an *Abort*.
Implementation:	"32,word"
Parameters:	*word:* The User Out WORD. Valid range: [1 to 1048576].
Returns:	An API Response code
Comments:	**NOTE:** Laser model numbers beginning with **8-75** support User In bits 1-4 ***only***. Model numbers beginning with **8-77** support User In bits 1-12. Bits 5-12 are defined in Appendix B as "Job Select Bit 1-8" All units support User Out bits 1-4 only.
Also see:	**TakeHostControl**, **GetUserInWord, GetAllIOWords, SetUserOutBit**

SetUserPIN	
Purpose:	Sets the current User PIN.
Numerical implementation:	"512,userpin"
Parameters:	*userpin*: A string representing the UserPIN.
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. This setting will not take effect until the board power is cycled, or a *HardwareReset* command is issued. The *UserPIN* is used with the Pendant interface, and provides password protection for User access functions.
Also see:	**TakeHostControl**, **HardwareReset**, **SetUserPIN**, **GetAdminPIN**, **SetAdminPIN**

TakeHostControl	
Purpose:	Request exclusive control of the Marker.
Numerical implementation:	"2"
Parameters:	None
Returns:	An API Response code
Comments:	A Client cannot gain exclusive control of the Marker if it is busy processing a job. Use *GetJobStatus* to determine if there is a job currently being processed.
Also see:	**ReleaseHostControl**, **GetJobStatus**

TransformObject	
Purpose:	Applies rotation, scaling and offset to the specified object.
Implementation:	"l02,objectindex,rotation,rotationcenterx,rotationcentery,xscale,yscale,xoffset,yoffset"
Parameters:	*objectindex*: The zero-based index of the object to translate. Valid range: [0 to object count - 1].
	rotation: The relative amount to rotate the object, in degrees. Valid range: [-360.0 to 360.0].
	rotationcenterx: The coordinate position, in field units (bits), representing the center of rotation in the x-axis. This value can be a coordinate position that is outside the normal marking field.. Valid range: [-2,1147,483,648 to 2,1147,483,647].
	rotationcenterx: The coordinate position, in field units (bits), representing the center of rotation in the y-axis. This value can be a coordinate position that is outside the normal marking field.. Valid range: [-2,1147,483,648 to 2,1147,483,647].
	xscale: Amount to scale object in the x-axis. Valid values: [> 0].
	yscale: Amount to scale object in the x-axis. Valid values: [> 0].
	xoffset: The amount to move the object in the x-axis in filed units (bits). Valid range: [-2,1147,483,648 to 2,1147,483,647].
	yoffset: Offset of the y-coordinate of all marking objects, in bits. Valid range: [-2,1147,483,648 to 2,1147,483,647].
Returns:	An API Response code

TransformObject	
Comments:	Client must call *TakeHostControl* before making this call. The Object at *objectindex* must be a valid marking object type. The *ActiveJob* must be *Idle* for this command to succeed. To determine ActiveJob status, use the *GetJobStatus* command. The transformations are applied in the following order: 1. Rotation. 2. Scaling 3. Offset Subsequent calls to *TransformObject* are cumulative. To restore the object to its original location, the job containing the object must be reloaded. In order to maximize performance, the Remote API does not check whether an object's vector list is within the legal marking field. **Therefore, it is the responsibility of the programmer** to insure that after an object has been transformed, it is within the legal marking field boundaries. Undefined results can be expected if an attempt is made to execute an object with a vector list outside the legal marking field. The bounds of an object can be discovered by calling *GetObjectRect*. The marking field is described using a Cartesian coordinate system, with (0,0) at the center of the field (-32768, -32768) at the bottom left corner, and (32767, 32767) at the top right corner. The field scale factor can be discovered by calling *GetKFactor*.
Also see:	**GetKFactor, GetJobStatus, TakeHostControl, GetObjectType, GetObjectRect, GetObjectCenter.**

TransformObjectByName	
Purpose:	Applies rotation, scaling, offset and (optionally) a new string value to the specified named string-based object and concatenates the transform values to the current *UserTransform*.
Numerical implementation:	"113,objectname,newstring,rotation,rotationcenterx,rotationcentery,xscale,yscale,xoffset, yoffset"
Parameters:	objectname: The name of the object. Valid number of characters: [1 to 255]
	newstring: The new string value. There cannot be embedded commas in the string. To pass strings with embedded commas, see *SetObjectString*. Special conditions: *DataMatrix*: To imbed control characters in the string, use the tilde (~) character before the control code. To imbed an actual ~character, use two tilde characters in a row (~ ~). To imbed an ASCII 0 character, use ~@ instead of the ASCII 0.
	rotation: The relative amount to rotate the object, in degrees. A positive rotation value results in a clockwise rotation. Valid range: [-360.0 to 360.0]
	rotationcenterx: The coordinate position, in field units (bits), representing the center of rotation in the x-axis. This value can be a coordinate position that is outside the normal marking field. Valid range: [-2,147,483,648 to 2,147,483,647]
	rotationcentery: The coordinate position, in field units (bits), representing the center of rotation in the y-axis. This value can be a coordinate position that is outside the normal marking field. Valid range: [-2,147,483,648 to 2,147,483,647]
	xscale: Amount to scale object in the x-axis. Valid range: [> 0]

TransformObjectByName	
	yscale: Amount to scale object in the y-axis. Valid range: [> 0]
	xoffset: The amount to move the object in the x-axis, in field units (bits). Valid range: [-2,147,483,648 to 2,147,483,647]
	yoffset: The amount to move the object in the y-axis, in field units (bits). Valid range: [-2,147,483,648 to 2,147,483,647]
Returns:	An API Response code
Comments:	Client must call *TakeHostControl* before making this call. The Object at *objectindex* must be a valid marking object type. The ActiveJob must be *Idle* for this command to succeed. To determine ActiveJob status, use the *GetJobStatus* command. The transformations are applied in the following order: 1. String is changed 2. Rotation 3. Sealing 4. Offset Both the Object *outline* and the Object *fill* are transformed with this call. When calling *SetObjectString*, or if the object will change its string value at run-time because of serialization, AutoDat, etc., the Object is processed in the following order: 1. New Outline vectors are generated 2. The JobTransform is applied (from the Job) 3. The Object is Justified 4. New Fill vectors are generated 5. The *UserTransform* is applied Subsequent calls to *TransformObject* are cumulative, as each transform is concantenated to the current *UserTransform*. To clear all transforms (setting the *UserTransform* to the Identity matrix), call *ResetUserTransform*. In order to maximize performance, the Remote API does not check whether an object's vectorlist is within the legal marking field. Therefore, ***it is the responsibility of the programmer*** to insure that after an object has been transformed, it is within the legal marking field boundaries. Undefined results can be expected if an attempt is made to execute an object with a vectorlist outside the legal marking field. The bounds of an object can be discovered by calling *GetObjectRect*. The *marking field* is described using a Cartesian coordinate system, with (0,0) at the center of the field. (-32768, -32768) at the bottom left corner, and 32767, -32767 at the top right corner. The field scale factor can be discovered by calling *GetKFactor*.
Also see:	**GekKFactor, GetJobStatus, TakeHostControl, GetObjectType, GetObjectRect, GetObjectCenter, ResetObjectTransform, SetObjString, TransformObject**

API Response Codes

The following are possible responses returned by the API. In certain cases, the response message may be an error message rather than the expected *Success* or return variable(s).

Value/constant	Description
0 = Success	The operation completed successfully
1 = Idle	The job engine is idle
2 = Busy	The job engine is currently executing a job
3 = NoJob	The specified job was not found
4 = InControl	The requesting client has exclusive control of the Host
5 = NotInControl	The requesting client does not have exclusive control of the Host
6 = LicenseUnavailable	A valid license was not found
7 = LicenseAccessDenied	The current license does not allow the requested feature
8 = BadCommand	The API command was not recognized
9 = BadArg	A specified argument was invalid
10 = ArgOutOfRange	A specified argument was out of range
11 = Unknown TimeZone	The specified time zone cannot be found
12 = BadConversion	Error while converting between multibyte and Unicode characters
13 = RegistryError	A Windows CE Registry read or write operation failed
14 = TimeZoneFileError	A TimeZoneFile operation failed
100 = NoFilesFound	No files were found at the specified path
101 = NoDrive	No drive was found
102 = JobOutOfMemory	Out of memory exception
103 = TooManyObjects	Internal error, consult factory
104 = NoObject	The specified object does not exist
105 = JobException	An internal job exception
106 = NotInHostControl	Operation cannot be performed if the client is not in control
107 = WrongHostType	Operation cannot be performed with this host type
108 = ErrorJobBusy	Operation cannot be performed while a job is executing
109 = NoActiveJob	There is no Active Job
110 = ErrorSoftware	Internal error, consult factory
111 = LoadFail	A job load failed
112 = NoObjects	Job file version not compatible with current firmware
113 = WriteFail	Internal error writing file
114 = JobFileFormat	Job file format error
115 = FileException	Internal error while processing file
116 = UnknownObject	Unknown object type
117 = UnknownType	Unknown type
118 = NotSupported	Operation not supported
119 = NotAvailable	Resource not available
120 = FPGADataFail	Internal FPGA data format failure
121 = FileNotFound	The file specified was not found

Value/constant	Description
200 = NoProperties	Object does not contain any properties
201 = ObjectException	Internal object exception
202 = Abort	Operation was aborted
203 = NoFontResource	The font specified in the object was not found
204 = NoOverride	Internal object error
205 = ExternalEnableDenied	Operation denied by External control
206 = CannotCreatePort	A port setting (baudrate, stopbits, etc.) is invalid
207 = CannotOpenPort	Error while attempting to open COM port
208 = PortNotOpen	Port must be open to execute command
209 = PortTimeout	A port operation timed out
210 = WrongPortNumber	Invalid port number
211 = WrongObjectType	Operation is not supported by this object type
212 = AxisNotConfigured	A motion control axis referenced in an object has not been configured
213 = TextBufferOverrun	Too many characters in buffer (MAX 3000 including command opcode)
214 = InvalidBCStringValue	The barcode string to encode contains invalid characters
215 = InvalidBCStringLength	The length of the barcode string to encode is invalid
216 = InvalidBCParameters	A barcode parameter is invalid
217 = BCTooLarge	The resulting barcode image is too large to process
218 = NoVectors	An object was saved to the job with both Mark Outline and Mark Fill disabled

Example Program

An example program is provided to illustrate how to initiate a session with the LEC-1, load a job stored locally on the controller, run the job once, and then close the session.

C# Example

This is an example written for the LAN interface. Error checking of the value returned from Socket.ReadLine() has been omitted for clarity.

```
//Connect to the LEC-1 (with specific platform function call)
Socket.Connect();
string result Socket.ReadLine(); //Welcome banner sent by LEC-1

//Take exclusive control of the LEC-
Socket.WriteLine("TakeHostcontrol");
result = Socket.ReadLine();

//Load a job
Socket.WriteLine("LoadFlashJob,testjob.dat");
result = Socket.ReadLine();

//Make the job active
Socket.WriteLine("MakeJobActive,testjob.dat");
result = Socket.ReadLine();

//Run the job once
Socket.WriteLine("ExecuteJobOnce");
result = Socket.ReadLine();

//Wait while job is running
do
{
        //Check job status
        Socket.WriteLine("GetJobStatus");
        result = Socket.ReadLine();
        Sleep(100);
}
while (result != "Idle");

//Close session
Socket.WriteLine("ReleaseHostControl");
result = Socket.ReadLine();

//Disconnect (with specific platform function call)
Socket.Close();
```

Section II. RS-232 and TCP/IP Commands to *Winlase* Software (Not Remote API)

The interface provided for RS-232 and TCP/IP is textual; commands are sent over either port as ASCII text strings. The commands listed below are in support of the standard RS-232 communications.

RS-232 and TCP/IP Interface

Command Syntax

For the sake of clarity, the responses listed below each command have been listed with their descriptive error codes. In practice, the responses are returned with numerical error codes. For example; when the Host is in control of WinLase, the Interface Request Status for **STATUS,IN_HOST_MODE** will be **STATUS,512**.

Command Set

The following list describes all of the **Remote Interface** commands and their intended use, and is presented in alphabetical order by command name. A description of the command parameters follows each command.

HOME		(**NOTE:** WinLase ***must*** be under **Host** control)
Purpose:	Commands all of the motor control axes to return to their Home position. This command is valid only if WinLase has been configured with a compatible motor controller.	
Responses:	ACK	Acknowledged
	ERROR,NOT_IN_HOST_MODE	WinLase must be in host mode
	ERROR,NO_MOTOR_CONTROLLER	Motor controller board not found
	ERROR,MOTOR_HOME	There was an error during the homing process

MODIFY,buffer,##,******		(**NOTE:** WinLase does not need to be in **Host** control)
Purpose:	To store the string '*****' in the internal string buffer at index ##. ## must be between 1 and 10. Text objects within the job must have their "Source" set to *Get String from Memory buffer* to use the buffer contents. Calling this will clear the previous value stored in the buffer.	
Responses:	ACK	Acknowledged
	ERROR,UNKNOWN_VERB	First word in command line not recognized
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized
	ERROR,NO_SUCH_BUFFER	Buffer ## out of range
	ERROR,INVALID_TEXT	The length of the ***** string is zero, or does not contain markable characters

MODIFY,field,##,******	(**NOTE:** WinLase ***must*** be under **Host** control)	
Purpose:	To modify a field of text or barcode where ## is the number of the field (object) to be modified, and "*****"is the new text string. The field ## corresponds to the position the object has in the Object List within the job (i.e. the first object in the Object List would have an index value of 1). If the index values are not known at run time, use **MODIFY**, buffer instead. The marker must be OFFLINE. If the field does not exist, an error is returned.	
Responses:	ACK	Acknowledged
	ERROR,NOT_IN_HOST_MODE	WinLase must be in host mode
	ERROR,UNKNOWN_VERB	First word in command line not recognized
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized
	ERROR,UNKNOWN_QUALIFIER	The ## field was not an integer value.
	ERROR,NO_SUCH_FIELD	The ## field index is larger than the total number of objects loaded.
	ERROR,UNKNOWN_QUALIFIER	The length of the ***** string is zero, or does not contain markable characters.

MODIFY,position,##,xoffset,yoffset	(**NOTE:** WinLase ***must*** be under **Host** control)	
Purpose:	To change the position of a marking object, where ## is the number of the object to be modified, *xoffset* is the amount to move the object in bits along the x-axis, and *yoffset* is the amount to move the object in bits along the y-axis. The field ## corresponds to the position the object has in the Object List within the job (i.e. the first object in the Object List would have an index value of 1).The marker must be OFFLINE. If the field does not exist, an error is returned.	
Responses:	ACK	Acknowledged
	ERROR,NOT_IN_HOST_MODE	WinLase must be in host mode
	ERROR,UNKNOWN_VERB	First word in command line not recognized
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized
	ERROR,UNKNOWN_QUALIFIER	The ## field was not an integer value.
	ERROR,NO_SUCH_FIELD	The ## field index is larger than the total number of objects loaded.

MODIFY,rotation,##,angle	(**NOTE:** WinLase ***must*** be under **Host** control)	
Purpose:	To rotate a marking object about it's center, where ## is the number of the field (object) to be modified, and angle is the amount to rotate the object, in units of 0.010 degrees. A positive angle value rotates the object clockwise. The object ## corresponds to the position the object has in the Object List within the job (i.e. the first object in the Object List would have an index value of 1).The marker must be OFFLINE. If the field does not exist, an error is returned.	
Responses:	ACK	Acknowledged
	ERROR,NOT_IN_HOST_MODE	WinLase must be in host mode
	ERROR,UNKNOWN_VERB	First word in command line not recognized
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized
	ERROR,UNKNOWN_QUALIFIER	The ## field was not an integer value.
	ERROR,NO_SUCH_FIELD	The ## field index is larger than the total number of objects loaded.

MODIFY,rotationex,##,angle,xcenter,ycenter	(**NOTE:** WinLase ***must*** be under **Host** control)	
Purpose:	To rotate a marking object about an arbitrary center of rotation, where ## is the number of the object to be modified, *angle* is the amount to rotate the object, in units of 0.010 degrees, *xcenter* is the x-axis center of rotation, in bits and *ycenter* is the y-axis center of rotation in bits. A positive angle value rotates the object clockwise. The object ## corresponds to the position the object has in the Object List within the job (i.e. the first object in the Object List would have an index value of 1). The marker must be OFFLINE. If the field does not exist, an error is returned.	
Responses:	ACK	Acknowledged
	ERROR,NOT_IN_HOST_MODE	WinLase must be in host mode
	ERROR,UNKNOWN_VERB	First word in command line not recognized
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized
	ERROR,UNKNOWN_QUALIFIER	The ## field was not an integer value.
	ERROR,NO_SUCH_FIELD	The ## field index is larger than the total number of objects loaded.

MODIFY,scale,##,xscale,yscale	(**NOTE:** WinLase ***must*** be under **Host** control)	
Purpose:	To scale a marking object from it's center, where ## is the number of the object to be modified, *xscale* is the amount to scale the object in the x-axis, in percent, and *yscale* is the amount to scale the object in the y-axis, in percent. For example, to decrease the size of an object to half it's current size, use the value 50.00 (%) for both the *xscale* and *yscale* values. The object ## corresponds to the position the object has in the Object List within the job (i.e. the first object in the Object List would have an index value of 1). The marker must be OFFLINE. If the field does not exist, an error is returned.	
Responses:	ACK	Acknowledged
	ERROR,NOT_IN_HOST_MODE	WinLase must be in host mode
	ERROR,UNKNOWN_VERB	First word in command line not recognized
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized
	ERROR,UNKNOWN_QUALIFIER	The ## field was not an integer value.
	ERROR,NO_SUCH_FIELD	The ## field index is larger than the total number of objects loaded.

OFFLINE	(**NOTE:** WinLase ***must*** be under **Host** control)	
Purpose:	Commands the laser to immediately stop marking, and returns the laser to the MARKER OFFLINE state	
Responses:	ACK	Acknowledged
	ERROR,NOT_IN_HOST_MODE	WinLase must be in host mode
	ERROR,ALREADY_OFFLINE	Marker is already offline

ONLINE		(**NOTE:** WinLase ***must*** be under **Host** control)
Purpose:	Commands the marker to start the marking process. System will immediately start polling external start port, and enter MARKER_ONLINE state. This call automatically sets the external start flag to true, and sets the repeat mode to repeat indefinitely.	
Responses:	ACK	Acknowledged
	ERROR,NOT_IN_HOST_MODE	WinLase must be in host mode
	ERROR,ALREADY_ONLINE	Marker is marking or waiting for external start signal.
	ERROR,NO_JOB_LOADED	No job loaded.
	ERROR,INTERLOCKS_OPEN	An interlock port on the interlock I/O card is open.
	ERROR,NO_SCANCARD	There is no scan head card installed in machine
	ERROR,NO_HARDLOCK	No Hardlock detected
	ERROR,NO_IOCARD	No I/O card installed in the computer.
	ERROR,STEP_REPEAT_INVALID	The values saved in the job for step and repeat will result in an invalid object position.
	ERROR,TEXT_SOURCE _INVALID	A text object was saved with a Source value incompatible with the host interface.
	ERROR,TEXTMERGE_INVALID	There was an error while processing a TextMerge file.
	ERROR,OBJECT_OUT_OF_BOUNDS	There is an object in the job that is outside the legal marking field.

OPEN,file,######		(**NOTE:** WinLase ***must*** be under **Host** control)
Purpose:	To open a file where "####" is a text string describing the file to be opened, and must be a fully qualified UNF file path. If the file cannot be found, or is corrupt, an error code is returned.	
Responses:	ACK	Acknowledged
	ERROR,NOT_IN_HOST_MODE	WinLase must be in host mode
	ERROR,FILE_NOT_FOUND	The file was not found at the indicated path location, or there was an error while opening the file.
	ERROR,UNKNOWN_VERB	First word in command line not recognized.
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized
	ERROR, UNKNOWN_QUALIFIER	The file path was less than 3 characters in length.

REQUEST,data,bits_per_mm,c#,h#		(**NOTE:** WinLase does not need to be in **Host** control)
Purpose:	Request WinLase to return the current value for the scan field bits/mm, where *c#* is the zero based card index and *h#* is the zero based scan head index. An example string would be: "REQUEST,data,bits_per_mm,0,0". The bits/mm value is the ratio of a point coordinate in bits and the actual position of the point in millimeters.	
Responses:	DATA,x	x is the bits/mm value.
	ERROR,UNKNOWN_VERB	First word in command line not recognized
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized

REQUEST,data,cyclecount	(**NOTE:** WinLase does not need to be in **Host** control)	
Purpose:	Request WinLase to return the current cycle count. The cycle count indicates the number of full cycles, including step and repeat.	
Responses:	DATA,x	x is the current cycle count.
	ERROR,UNKNOWN_VERB	First word in command line not recognized.
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized.

REQUEST,data,partcount	(**NOTE:** WinLase does not need to be in **Host** control)	
Purpose:	Request WinLase to return the current part count. The part count is the individual marks within a cycle.	
Responses:	DATA,x	x is the current part count.
	ERROR,UNKNOWN_VERB	First word in command line not recognized
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized

REQUEST,data,cycletime	(**NOTE:** WinLase does not need to be in **Host** control)	
Purpose:	Request WinLase to return the current cycle time. The cycle time is defined as the elapsed time to do all marks within a single cycle.	
Responses:	DATA,x	x is the current cycle time.
	ERROR,UNKNOWN_VERB	First word in command line not recognized
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized.

REQUEST,data,parttime	(**NOTE:** WinLase does not need to be in **Host** control)	
Purpose:	Request WinLase to return the current part mark time. The part time is defined as the elapsed time to do a single mark within an overall cycle.	
Responses:	DATA,x	x is the current part time.
	ERROR,UNKNOWN_VERB	First word in command line not recognized
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized.

REQUEST,data,jobname	(**NOTE:** WinLase does not need to be in **Host** control)	
Purpose:	Request WinLase to return the name of the currently loaded job.	
Responses:	DATA,*filepath*	*filepath* is the fully qualified path to the currently loaded job file.
	ERROR,UNKNOWN_VERB	First word in command line not recognized
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized.

REQUEST,data,version		(**NOTE:** WinLase does not need to be in **Host** control)
Purpose:	Request WinLase to return its Version number.	
Responses:	DATA,x	x is the Version number
	ERROR,UNKNOWN_VERB	First word in command line not recognized
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized

REQUEST,data,user		(**NOTE:** WinLase does not need to be in **Host** control)
Purpose:	Request WinLase to return the User currently logged on to the current Windows NT/2000 session.	
Responses:	DATA,*username*	*username* is the currently logged on user.
	ERROR,UNKNOWN_VERB	First word in command line not recognized.
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized.

REQUEST,field, ##		(**NOTE:** WinLase ***must*** be under **Host** control)
Purpose:	To request data from a field of text, barcode or graphic where ## is the number of the field to be queried. The marker must be OFFLINE. If the field does not exist, an error is returned.	
Responses:	DATA,*field#*,*objecttype*,*data*	field# is the field # of the object.
		objecttype is object type defined in Appendix B data is the string value for text and barcodes and the graphic file path for a graphic object.
	ERROR,NOT_IN_HOST_MODE	WinLase must be in host mode.
	ERROR,UNKNOWN_VERB	First word in command line not recognized.
	ERROR,UNKNOWN_NOUN	Second word in command line not recognized.
	ERROR,UNKNOWN_QUALIFIER	The ## field was not an integer value.
	ERROR,NO_SUCH_FIELD	The ## field index is larger than the total number of objects loaded.

REQUEST,status,interface		(**NOTE:** WinLase does not need to be in **Host** control)
Purpose:	Returns the current status of the Host interface	
Responses:	STATUS,IN_HOST_MODE	Host is in control of WinLase.
	STATUS,HOST_NOT_READY	Not available for host command.
	STATUS,HOST_READY	Available for host command.

REQUEST,status,marker		(**NOTE:** WinLase does not need to be in **Host** control)
Purpose:	Returns the current status of the laser marker and WinLase software.	
Responses:	STATUS,HOST_NOT_READY	Cannot get status of marker because host is not available for host command.
	STATUS,INTERLOCKS_OPEN	An interlock port on the interlock I/O card is open.
	STATUS,MARKER_ONLINE	Marker is marking or waiting for external start signal.
	STATUS,MARKER_OFFLINE	Job is loaded and marker is ready to accept ONLINE command or MODIFY command.
	STATUS,NO_JOB_LOADED	No job loaded.
	STATUS,ERROR_PROCESS	There was an error while in the ONLINE mode. This error will be cleared after it is read once, and if all OK, the next response will be STATUS,MARKER_OFFLINE.

RUN		(**NOTE:** WinLase ***must*** be under **Host** control)
Purpose:	Commands the marker to start the marking process. System will immediately execute the currently loaded job and enter the MARKER_ONLINE state. This call does not automatically set the external start flag to true, and ***does not*** set the repeat mode to repeat indefinitely. The current job settings will be used for these two parameters.	
Responses:	ACK	Acknowledged.
	ERROR,NOT_IN_HOST_MODE	WinLase must be in Host mode.
	ERROR,ALREADY_ONLINE	Marker is marking or waiting for external start signal.
	ERROR,NO_JOB_LOADED	No job loaded.
	ERROR,INTERLOCKS_OPEN	An interlock port on the interlock I/O card is open.
	ERROR,NO_SCANCARD	There is no scan head card installed in machine.
	ERROR,NO_HARDLOCK	No Hardlock detected.
	ERROR,NO_IOCARD	No I/O card installed in computer.
	ERROR,STEP_REPEAT_INVALID	The values saved in the job for step and repeat will result in an invalid object position.
	ERROR,TEXT_SOURCE_INVALID	A text object was saved with a Source value incompatible with the host interface.
	ERROR,TEXTMERGE_INVALID	There was an error while processing a TextMerge file.
	ERROR,OBJECT_OUT_OF_BOUNDS	There is an object in the job that is outside the legal marking field.

SET,control,host		(**NOTE:** WinLase does not need to be in **Host** control)
Purpose:	Puts WinLase into external control mode. All user input at the console is disabled.	
Responses:	ACK	Acknowledged.
	ERROR,ALREADY_IN_HOST_MODE	Host is already in Host mode.
	ERROR,HOST_NOT_READY	Host cannot go into host mode because the *Allow Host Control* check box in WinLase is cleared.

SET,control,local		(**NOTE:** WinLase ***must*** be under **Host** control)
Purpose:	Releases WinLase from the external control mode. Enables user input at the console.	
Responses:	ACK	Acknowledged.
	ERROR,ALREADY_IN_LOCAL_MODE	Host is already in local mode.
	ERROR,MARKER_ONLINE	Marker is marking or waiting for external start signal.

Example Program

An example program is provided to illustrate how to initiate a session with WinLase, manipulate an object in the loaded job, run the job, and then close the session.

C++ Example

The following pseudo-code uses an application defined function called **SendToSocket()**, which represents a method of outputting text from either the RS-232 or TCP/IP ports, and receiving a response as it's return value.

```
//Aquire WinLase
SendToSocket ("SET,control,host");
//Make sure we have control
if(SendToSocket ("REQUEST,status,interface")!=" STATUS,512 )
        return ERROR;
//Load a job
SendToSocket ("OPEN,file,c:\\test\\job\\test.wlj");
//Make sure job has loaded properly
if(SendToSocket ("REQUEST,status,marker")!=" STATUS,2300 )
        return ERROR;
//Change the text in the object at index position 2
SendToSocket ("MODIFY,field,2,"Hello World");
//Put WinLase into the ONLINE mode, waiting for START PROCESS to toggle
SendToSocket ("ONLINE");
//Verify we are in ONLINE MODE
if(SendToSocket ("REQUEST,status,marker")!=" STATUS,2301)
        return ERROR;
//Mark some parts

//Stop polling the STARTPROCESS input
SendToSocket ("OFFLINE");
//Release WinLase
SendToSocket ("SET,control,local");
```