



**Windows  
Sockets**

## **I-Mark WinSock Programming**

---

**Technical Brochure**



## Controlling I-Mark through WinSock TCP/IP

### The Basics...

I-Mark has a feature which is turned on by default that enables it to listen for ASCII text string commands over TCP/IP. The default listening port is **5555**. Each ASCII Command should be structured in the following format. **Command [Argument] <value> EOS**

The command and arguments can be found in the Commands section of this document. If any arguments are required in the command, they must be sent along with it as one string. Likewise, if the command and argument are providing a value for the controller to use, the entire string must be sent as one. Each string should be terminated with the **EOS** character or **End of String** (ASCII Dec 10 Hex 0x0A). If the command and all of its arguments are accepted, then the controller will respond with the **%** character. If the command sent has invalid arguments or values or is simply not recognized, I-Mark will respond with the **?** character.

### Contents

Controlling I-Mark through WinSock TCP/IP .....	2
The Basics.....	2
The Commands.....	3
The Commands Explained.....	4
DIN Function:.....	4
DOUT Function:.....	5
DISABLE Function:.....	6
ENABLE Function:.....	7
HOME Function:.....	8
IGLOBAL Function:.....	9
RESET Function:.....	11

# Marking Machines



## The Commands...

Most of these you won't be using but they're there just in case you decide to use them.

Command	Arguments	Value Format	Example
DIN	(Axis Name, ALL)		Returns 8 bit byte of the Inputs
DOUT	Axis, Port, Bits:Value	1:1 = Output 1 On  1:0 = Output 1 Off	DOUT X,0,1:0,2:1,3:0
DISABLE	Axis names (X, Y)		DISABLE X Y
ENABLE	Axis names (X, Y)		ENABLE X Y
HOME	Axis names (X, Y)		HOME X Y
IGLOBAL	(Register Number) =value	Value is 32bit Int	IGLOBAL (500)=88456864 IGLOBAL(500) will return value
RESET			Will Reset the controller



## The Commands Explained...

### DIN Function:

This command is used in this format to return the value of the Digital Inputs on the controller and their state. This is useful to determine if an input wired into the controller has been pressed. If use it is recommended that this command be implemented on a timer event to poll the controller for status.

**Syntax:**

<return> = DIN(<Axis>, <Port>)

<return> = DIN(<Axis>, <Port>, <Bit>)

Argument	Type	Description
<Axis>	Axis	The axis (or axes) on which to execute the command. ALWAYS = X
<Port>	32-bit signed integer	The port from which to read the value. SHOULD ALWAYS = 1
<Bit>	32-bit signed integer	The bit to read. 0-7
RETURNS	32-bit signed integer	The value of the digital input. 0 = Off 1 = On

**Example:**

DIN (X,ALL) 'This will return the 8 bit value of the X-Axis DIN

DIN (X, 1, 1) 'This will return the state of the individual bit 1 on the Controller.  
'NOTE: Always use port 1 for I-Mark.

# Marking Machines



## DOUT Function:

This command is used in this format to set the value of the Digital Outputs on the controller and their state.

### Syntax:

```
DOUT <Axis>, <Port>, <Value>
```

```
DOUT <Axis>, <Port>, <Bits>:<Values> [ , <Bits>:<Values> ... ]
```

Argument	Type	Description
<Axis>	Axis	The axis (or axes) on which to execute the command.
<Port>	32-bit signed integer	The port from which to read the value. SHOULD ALWAYS = 1.
<Value>	32-bit signed integer	The value to set the port to.
<Bits>	32-bit signed integer	The value of the digital input. 0 = Off 1 = On.
<Values>	32-bit signed integer	The values to set the bits to.

### Example:

DOUT X, 1, 7 Bits 2,1,0 = 1) on X axis	'Write the value of 7 to the port (Bit 3 = 0, Bits 2,1,0 = 1) on X axis
DOUT X, 1, 1:0	'Write the value of 0 to bit 1 on the X Axis.



# Marking Machines



## ENABLE Function:

This command is used to Enable the Axis or Axes and restore power to the motors. This command cannot be executed if the marker is already in motion.

Syntax: `ENABLE <Axis> [ <Axis> . . . ]`

Argument	Type	Description
<Axis>	Axis	The axis (or axes) on which to execute the command.

Example:

```
ENABLE X Y           'Enables the Axis X and Y
```



# Marking Machines



## IGLOBAL Function:

This command is used in this read or write values from/into the Global Integer Registers of I-Mark. This will be the primary function for controlling I-Mark and Writing Placeholder Data into I-Mark to be printed. It can also be used to verify that the correct data has been transmitted or as a check to ensure that no data is already present in the registers.

Syntax:	IGLOBAL( <RegNumber>)=<Value>
	<return> = IGLOBAL(<RegNumber>)

Argument	Type	Description
<RegNumber>	32-bit signed integer	The variable (register) number that is set.
<Value>	32-bit signed integer	The value to store.
RETURNS	32-bit signed integer	The value stored.

**NOTE:** There are specific registers which **MUST** be used for I-Mark and none other. Writing to an incorrect register may interfere with the normal operation of the marking system.



## Register Map

RegNumber	Type	Description
500	32-bit signed integer	This control register used to command I-Mark. (See EtherNet/IP Docs)
550	32-bit signed integer	The status register where I-Mark reports its current status.
300-319	32-bit signed integer	Register range for PLACEHOLDER 0, each register stores 4 chars of ascii data
320-339	32-bit signed integer	Register range for PLACEHOLDER 1, each register stores 4 chars of ascii data
340-359	32-bit signed integer	Register range for PLACEHOLDER 2, each register stores 4 chars of ascii data
360-379	32-bit signed integer	Register range for PLACEHOLDER 3, each register stores 4 chars of ascii data
380-399	32-bit signed integer	Register range for PLACEHOLDER 4, each register stores 4 chars of ascii data
400-419	32-bit signed integer	Register range for PLACEHOLDER 5, each register stores 4 chars of ascii data
420-439	32-bit signed integer	Register range for PLACEHOLDER 6, each register stores 4 chars of ascii data
440-459	32-bit signed integer	Register range for PLACEHOLDER 7, each register stores 4 chars of ascii data
460-479	32-bit signed integer	Register range for PLACEHOLDER 8, each register stores 4 chars of ascii data
480-499	32-bit signed integer	Register range for PLACEHOLDER 9, each register stores 4 chars of ascii data

# Marking Machines



Example:

IGLOBAL(500)=1	'Writes the value of 1 into the register, I-Mark interprets this command as a Mark command. See EtherNet/IP documentation for Control Values.
IGLOBAL(500)=2	'Writes the value of 2 into the register, I-Mark interprets this command as a Reset command
IGLOBAL(300)= 1094861636	'Writes the value into the register, I-Mark will interpret this as ASCII "ABCD"
IGLOBAL(550)	Returns the value of I-Mark status register. See EtherNet/IP documentation for breakdown of status value.

## RESET Function:

This command is used to reset the I-Mark controller. This does not clear any of the register's values or NonVolatile memories but it will cause the hardware to restart.

Syntax: RESET

Example:

RESET	'Resets the I-Mark controller