mecco®

marking & traceability

etherMark™

ACC-B-001.1

COMMUNICATION GUIDE

www.mecco.com

Table of Contents

Purpose

This document describes the EtherMark™ protocol available as an option with MeccoMark® laser marking systems and Couth dot peen marking systems.

Scope

This manual provides the information required to implement the EtherMark control protocol in a production environment. It includes technical specifications, recommended network setup, register list, commands, states, and example programs. This protocol supports PLCs using the Ethernet/IP standard (support for PROFINET coming soon).

PLC programming is beyond the scope of this manual. While examples of PLC programming are provided here, an operator should obtain specific training on PLC logic before attempting to program a system in a production line.

Introduction

EtherMark is a network protocol developed by Mecco specifically to enable MeccoMark® and Couth marking systems to be controlled by programmable logic controllers (PLCs) using industry standard APIs (application programming interfaces). Using EtherMark enables your product marking systems to be integrated into a factory's existing Ethernet/IP automation network. Marking jobs can be programmed using standard PLC logic, eliminating the need for operators to learn a new programming language.

EtherMark is implemented as a combination of hardware and software that provides full ODVA compliance to the Ethernet/IP object model interface. Jobs can be directly loaded from the PLC, eliminating the need for a PC on the factory floor.

Technical Specifications

Connection: 10/100 Mbps Ethernet standard RJ45 port, to network and PLC

Compliancy: ODVA compliant EIP/CIP object model interface

IP Address: Manually set fixed IP configured by user

Firmware: Flash firmware upgrade via Ethernet or USB

Control:

- Supervisory control of LEC Laser Marker
- Supervisory control of Couth Dot Peen Marker

Barcoding:

- Laser: Capable of 3000 characters in a 2D barcode with string concatenation
- Dot Peen: Capable of 68 characters in a 2D barcode

How to Follow These Instructions

EtherMark Control may be used with your MeccoMark laser marker or with your Couth dot-peen markers. Some configuration steps described in this manual apply, no matter which type of marker you are using. Other steps may apply only to the laser marker or to the dot-peen marker. To make it easy to get started, the sections and steps in this manual are color-coded with a stripe in the right-hand margin for applicability. Red-striped text applies to laser units only, blue-striped text applies to dot-peen units only, and green-striped text applies to both types.

When parallel information is presented, that is, a similar task for either laser or dot-peen, but implemented in a manner specific to the device, then the info will be presented in a side-by-side format, with laser on the left side and dot-peen on the right side.

When a task applies *only* to a specific technology (laser or dot-peen), then the information will be presented across the full width of the page.

Color-coding Example

For this example, the text in this section applies equally to laser or dot-peen units. The green stripe in the right-hand margin means that you should read this section regardless of which type of marker you are configuring for control.

Laser Only Example

Some sections and steps in this manual apply only to the laser marker units. In that case, text will be striped red in the right-hand margin. If you do not have a laser marker in your setup you can safely ignore these sections and continue with green and blue striped sections.
This line shows a parallel task example – the info to the right of this text will be for a similar task, but specific to the dot-peen.

Dot-Peen Only Example

When specific information applies only to Couth dot-peen marker units, the text for that section or step will be striped blue in the right-hand margin. If your marking line uses only dot-peen marker technology then pay attention only to the blue and green striped information. Some installations make use of both types of marker technology. In those instances you will want to pay equal attention to all the information provided here.

Suggested Network Setup

The circuit that implements the EtherMark protocol is integrated within the control box for the laser or dot peening unit. Connecting the laser or dot peen control box to your Ethernet network also connects your EtherMark card to the network.

IP Addressing

When setting up EtherMark marking devices on your network, do not use *Dynamic Host Configuration Protocol* (DHCP). These devices are addressed with static IPs. As shipped, the EtherMark card that is installed in Mecco laser markers is preset with a static IP address of 192.168.4.3. The LEC in the laser control box is preset with a static IP address of 192.168.4.4. These addresses, however, may be reset to suit the needs of your installation using any standard networking utility. For example, if your setup includes multiple EtherMark enabled markers, you will need to ensure that each card has a unique IP address for control.

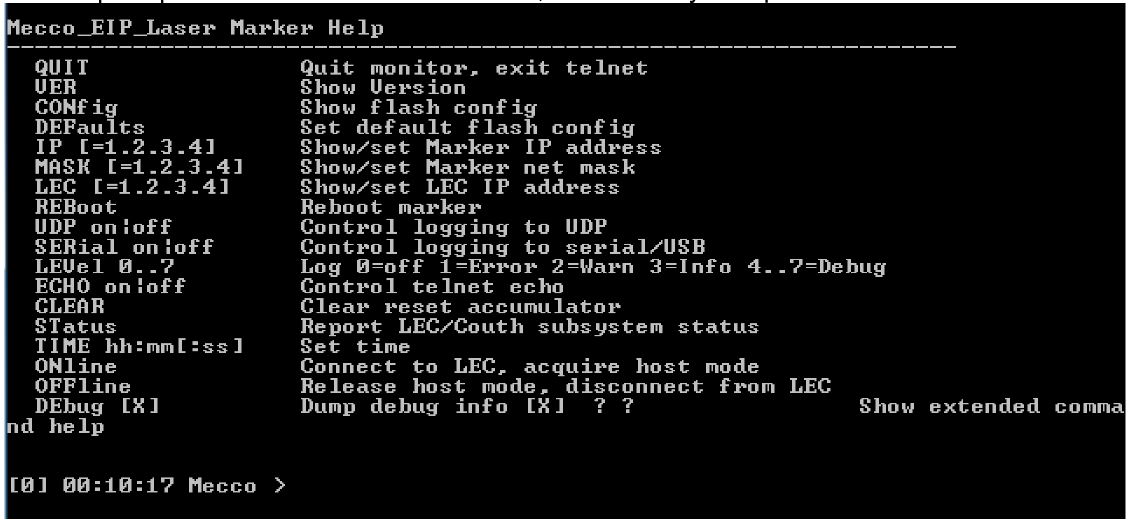
Telnet

Telnet or any other standard networking utility may be used to communicate with the EtherMark card across your local network connections. A networking utility is a reliable "back-end" tool that can be used directly from the command line or easily driven by other programs and scripts. Start your preferred utility (e.g., telnet) and connect with the EtherMark card using its IP address. The default IP address is 192.168.4.3

Example (using telnet):

```
C:> telnet [host]
C:>telnet 192.168.4.3
```

Once connected to the device, issue the help command to see the list of options available. This level of help is specific to the EtherMark devices, not the utility's help file.

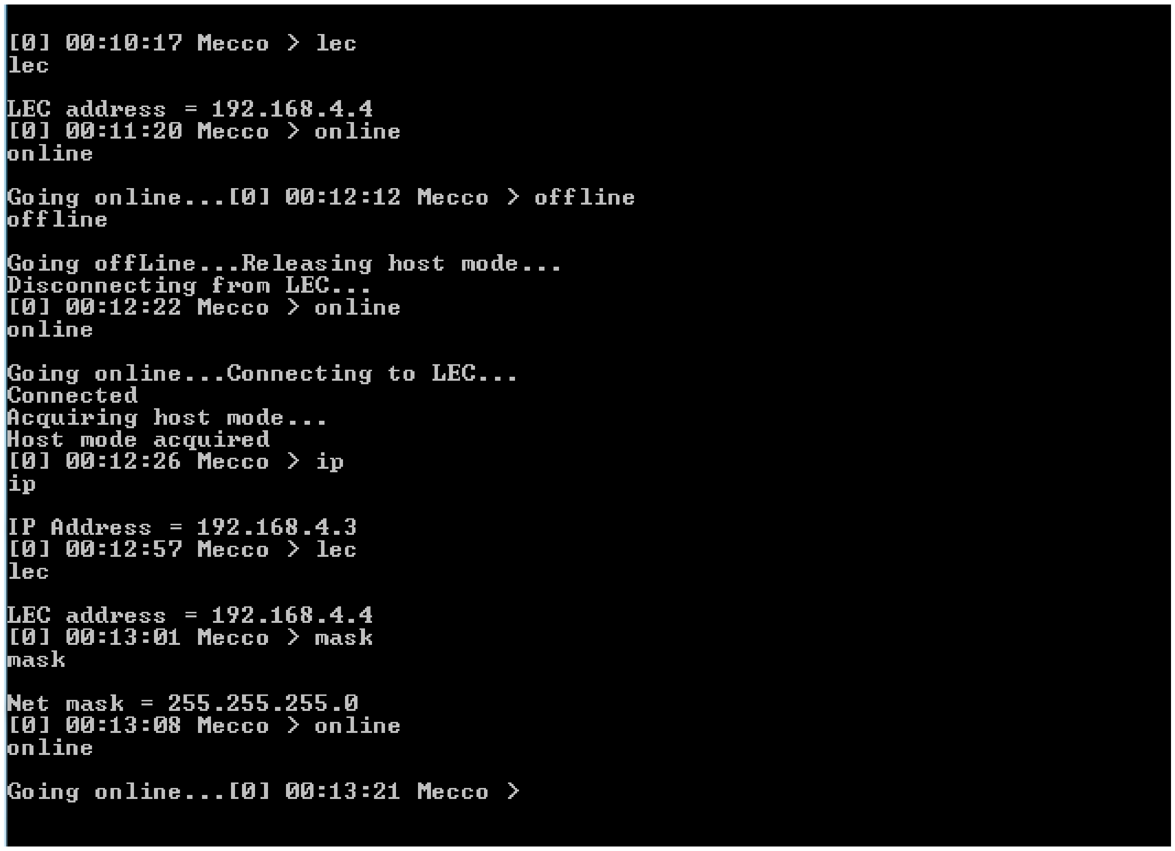


```
Mecco_EIP_Laser Marker Help
----------------------------------------------------------------------
  QUIT                   Quit monitor, exit telnet
  VER                    Show Version
  CONFig                 Show flash config
  DEFaults               Set default flash config
  IP [=1.2.3.4]          Show/set Marker IP address
  MASK [=1.2.3.4]        Show/set Marker net mask
  LEC [=1.2.3.4]         Show/set LEC IP address
  REBoot                 Reboot marker
  UDP on|off             Control logging to UDP
  SERial on|off          Control logging to serial/USB
  LEVel 0..7             Log 0=off 1=Error 2=Warn 3=Info 4..7=Debug
  ECHO on|off            Control telnet echo
  CLEAR                  Clear reset accumulator
  STatus                 Report LEC/Couth subsystem status
  TIME hh:mm[:ss]        Set time
  ONline                 Connect to LEC, acquire host mode
  OFFline                Release host mode, disconnect from LEC
  DEbug [X]              Dump debug info [X]  ? ?           Show extended comma
nd help


[0] 00:10:17 Mecco >
```

Figure 1 - Network Help File

As an example of communicating over the network with your devices, see below how to show the IP address of the LEC. The command lec is issued without arguments and it returns the current IP address of the card – 192.168.4.4 (factory default). Next, the online command is issued to put the LEC in HOST mode. HOST mode enables the LEC to receive instructions. The offline command releases the LEC from HOST mode. Back in HOST mode, the ip command, without arguments, returns the address of the EtherMark card (the "Marker").



```
[0] 00:10:17 Mecco > lec
lec

LEC address = 192.168.4.4
[0] 00:11:20 Mecco > online
online

Going online...[0] 00:12:12 Mecco > offline
offline

Going offLine...Releasing host mode...
Disconnecting from LEC...
[0] 00:12:22 Mecco > online
online

Going online...Connecting to LEC...
Connected
Acquiring host mode...
Host mode acquired
[0] 00:12:26 Mecco > ip
ip

IP Address = 192.168.4.3
[0] 00:12:57 Mecco > lec
lec

LEC address = 192.168.4.4
[0] 00:13:01 Mecco > mask
mask

Net mask = 255.255.255.0
[0] 00:13:08 Mecco > online
online

Going online...[0] 00:13:21 Mecco >
```

Figure 2 - Example Communicating with the LEC via Netcat

IMPORTANT NOTE	When using the `LEC=[1.2.3.4]` command, you are defining how the EtherMark card addresses the LEC card – **This command does NOT actually change the physical IP address of the LEC**. Use the WinLase LAN program to change the physical IP address of the LEC card to match the address known by the EtherMark card.

To change the address of the “Marker” or the LEC, issue the appropriate commands with a new static address as the argument, as shown in the examples:
`Mecco > lec =192.168.20.1` changes the IP address of the **LEC** to 192.168.20.1. Likewise,
`Mecco > ip =192.168.30.1` changes the **EtherMark** marker address to 192.168.30.1.

Communication

The EtherMark enables your markers to communicate directly with PLCs in your system, without the need of a host PC. It also enables your personnel to program the marker using the standard PLC ladder logic with which they are already familiar. There is no need for them to learn a new language when integrating an EtherMark enabled marker in your manufacturing process.

With a laser marker unit the EtherMark communicates with the LEC through an internal switch and then communicates to a network switch, issuing commands and receiving responses between the marker and the PLC.

In a Couth dot-peen marker, the EtherMark card communicates with the Couth T2 control unit via an internal RS232 connection. The dot-peen marker attaches to your network via an RS232 connection to a network switch. Commands and responses transit through the switch between the marker and the PLC.

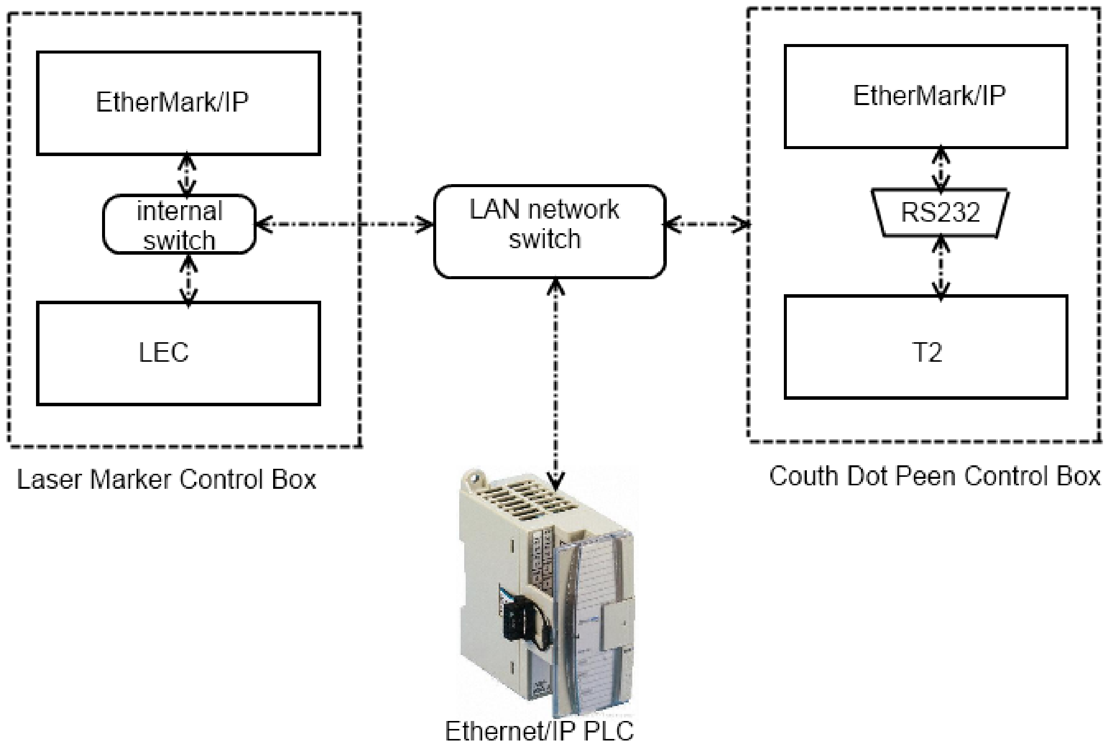


Figure 3 - Network Topology Example

Setup for Operations

To setup for operations a template is created and loaded onto the marker (laser or dot peen device). The template establishes the basic parameters for marking an item, such as defining job file name, text strings, X-Y coordinates to begin marking, etc. Once the template has been loaded, the variable parameters may be provided to the marker by any one of several means:

- Locally, through manual input issued at the HMI (Human Machine Interface)
- Locally, by stored inputs on a USB memory stick (limited to 2GB)
- Programmatically issued inputs via the Local Area Network (LAN)

RSLogix 5000

If you are integrating your EtherMark device into a system using Allen-Bradley PLCs you will need to setup the PLC to recognize the marker device using RSLogix 5000 (ver. 20 or better) from Rockwell Automation. The steps provided here are intended to outline the process in a general way. Refer to your RSLogix 5000 documentation for detailed information

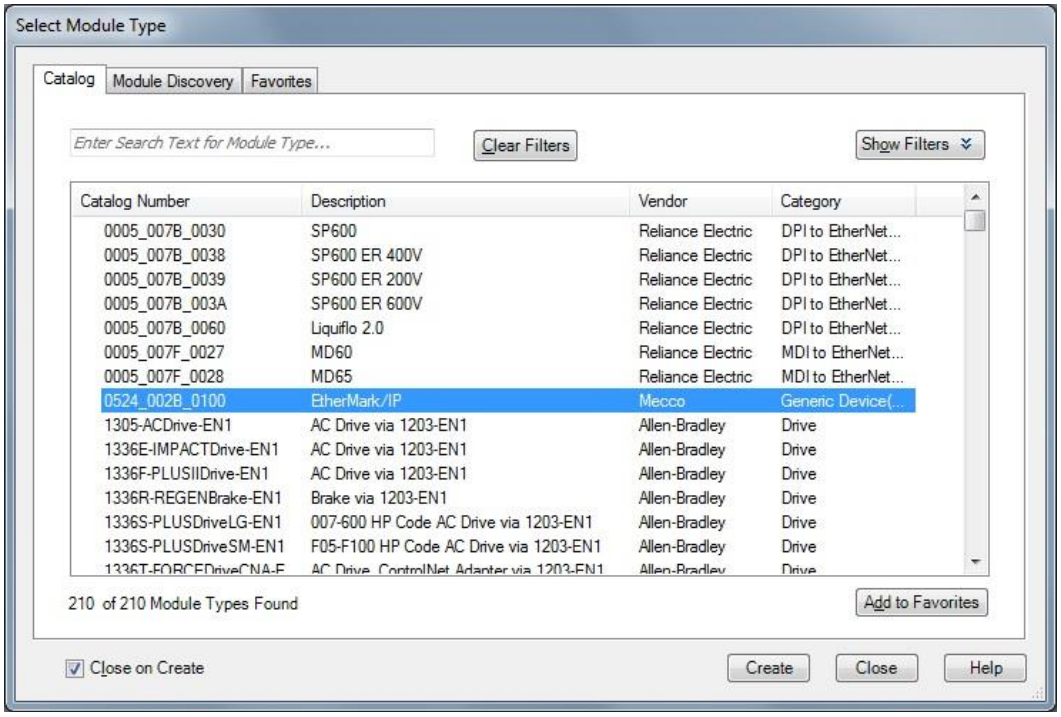
Add a New Module

1. Open a project, or create a new one.
2. Ensure that the PLC has a unique IP address on the same subnet as the marker you are defining.

IMPORTANT NOTE

When adding a new Ethernet module, **DO NOT** choose the "Ethernet/IP" catalog item

3. Add a new Ethernet module to your project by right-clicking to select NEW MODULE, then select the **EtherMark** catalog item ("Mecco - Generic Device").


Select Module Type

Catalog | Module Discovery | Favorites

Enter Search Text for Module Type...

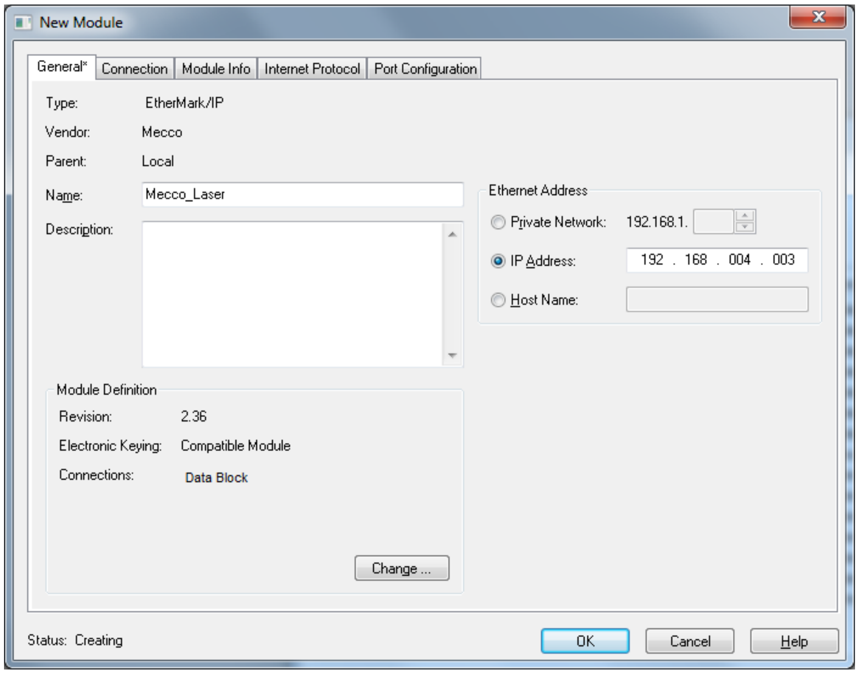
Catalog Number	Description	Vendor	Category
0005_007B_0030	SP600	Reliance Electric	DPI to EtherNet...
0005_007B_0038	SP600 ER 400V	Reliance Electric	DPI to EtherNet...
0005_007B_0039	SP600 ER 200V	Reliance Electric	DPI to EtherNet...
0005_007B_003A	SP600 ER 600V	Reliance Electric	DPI to EtherNet...
0005_007B_0060	Liquiflo 2.0	Reliance Electric	DPI to EtherNet...
0005_007F_0027	MD60	Reliance Electric	MDI to EtherNet...
0005_007F_0028	MD65	Reliance Electric	MDI to EtherNet...
0524_002B_0100	EtherMark/IP	Mecco	Generic Device(...
1305-ACDrive-EN1	AC Drive via 1203-EN1	Allen-Bradley	Drive
1336E-IMPACTDrive-EN1	AC Drive via 1203-EN1	Allen-Bradley	Drive
1336F-PLUSIIDrive-EN1	AC Drive via 1203-EN1	Allen-Bradley	Drive
1336R-REGENBrake-EN1	Brake via 1203-EN1	Allen-Bradley	Drive
1336S-PLUSDriveLG-EN1	007-600 HP Code AC Drive via 1203-EN1	Allen-Bradley	Drive
1336S-PLUSDriveSM-EN1	F05-F100 HP Code AC Drive via 1203-EN1	Allen-Bradley	Drive
1336T-FORCEDriveCNA-E	AC Drive, ControlNet Adapter via 1203-EN1	Allen-Bradley	Drive

210 of 210 Module Types Found

Figure 4 - Add a New Ethernet Module

1. Ensure that the **IP address** on the **GENERAL** tab is set to match the IP address of the EtherMark card inside your marker (**<u>not</u>** the IP address of the LEC).

2. Enter a unique descriptive name for your marker in the **Name** field.


New Module

General* | Connection | Module Info | Internet Protocol | Port Configuration

Type: EtherMark/IP
Vendor: Mecco
Parent: Local
Name: Mecco_Laser
Description:

Ethernet Address
Private Network: 192.168.1.
[✓ IP Address: 192 . 168 . 004 . 003]
Host Name:

Module Definition
Revision: 2.36
Electronic Keying: Compatible Module
Connections: Data Block

Status: Creating

Figure 5 - Configuring a New Module

IMPORTANT

NOTE

If timeout issues are encountered during operations, try increasing the **Connection** setting by increments of 10 ms. If timeout issues persist, troubleshoot your network.

1. On the **New Module** setup dialog, switch to the **CONNECTION** tab and set the timeout to at least 10 ms.

Test the New Module

2. Test communications between the EtherMark card and the PLC:

   a. From the menu, select **Communications > Go Online**

   b. Follow the prompt to load the project to the PLC

   c. Double-click the new module in the RSLogix tree and verify status “Running” is displayed in the lower left-hand corner of the dialog box.

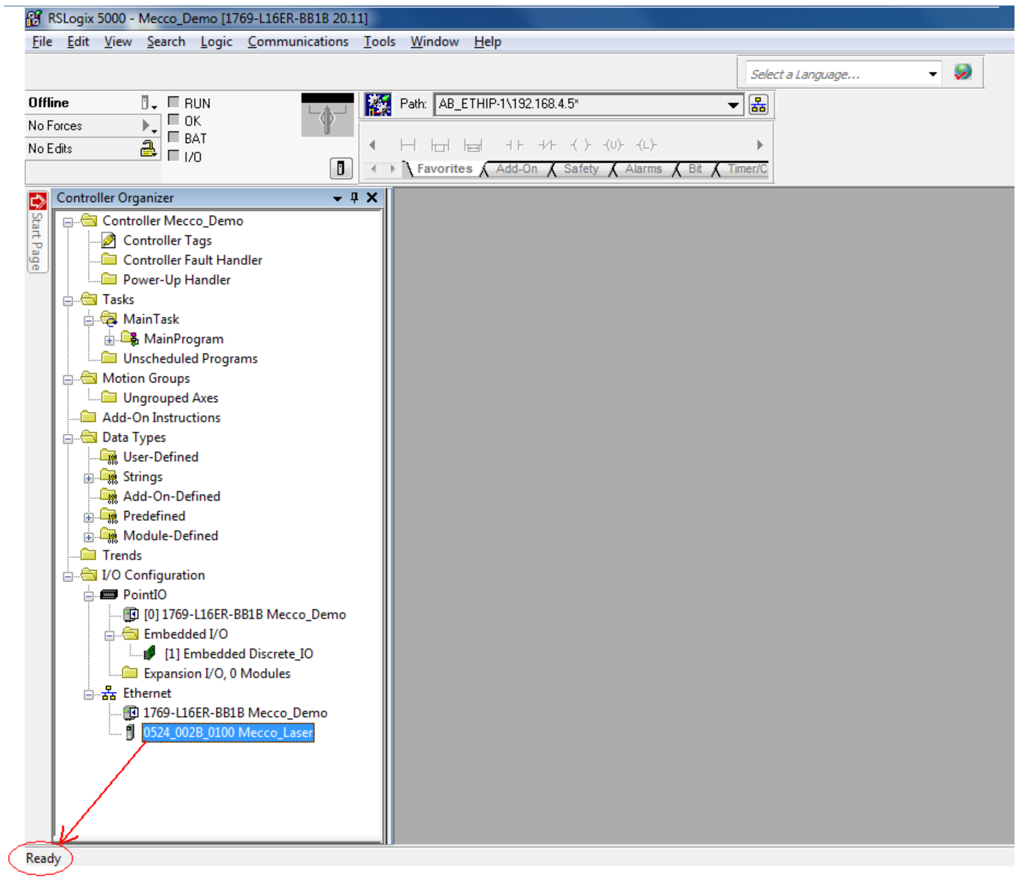


Figure 6 - EtherMark Card Communicating with PLC

Import EtherMark Logic

Your ladder logic project requires 2 rungs of logic that are specific to the type of marker your project is using. This logic is included in files on the USB stick included with your Mecco products.

1. Right-click in your MainRoutine panel and select **Import Rungs**.

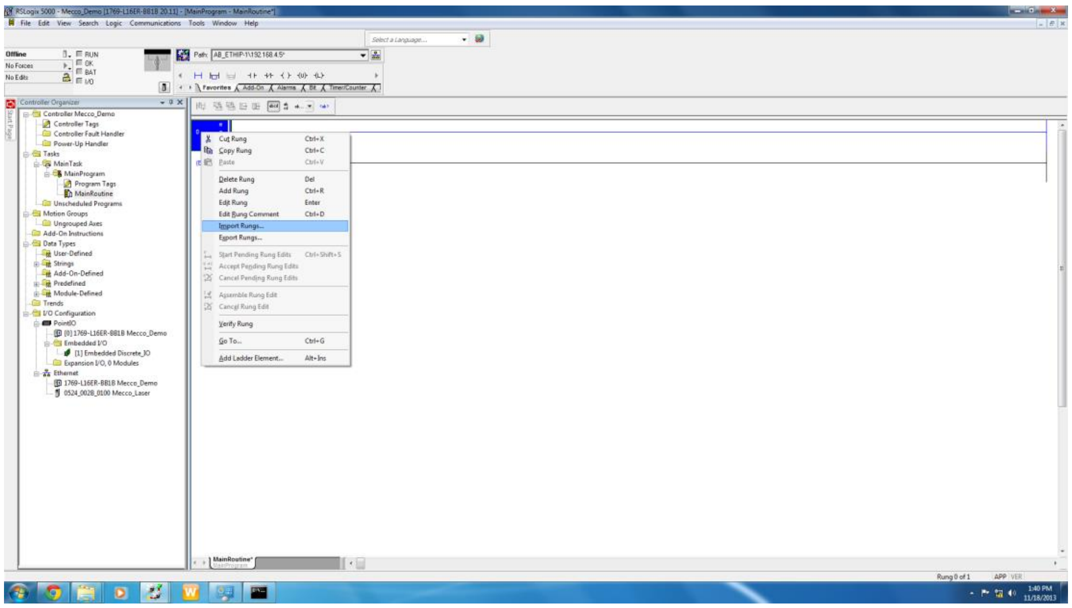
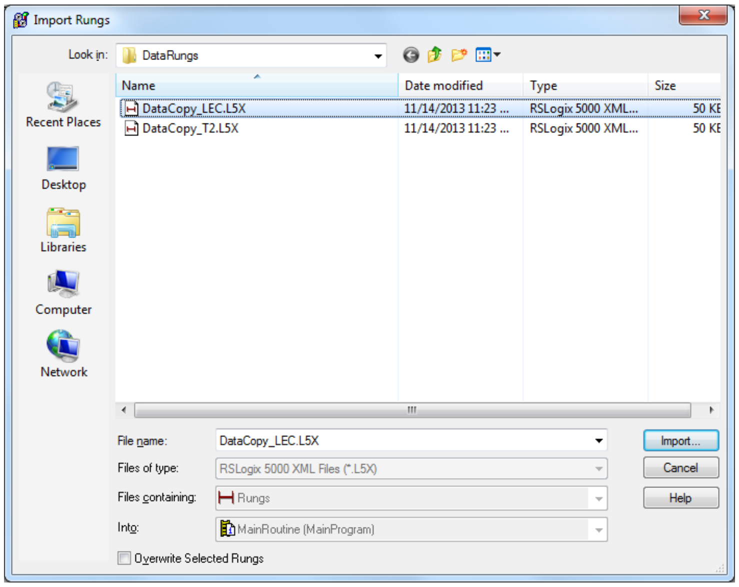


Figure 7 - Import Rungs

9. Navigate to the USB stick, select the file for the marker type you are configuring and click **Import**. (**DataCopy_LEC.L5X** is selected for laser markers; **DataCopy_T2.L5X** is selected for dot-peen markers.) In Figure 8, logic for a laser marker is selected.



Import Rungs

Look in: DataRungs

Name	Date modified	Type	Size
DataCopy_LEC.L5X	11/14/2013 11:23 ...	RSLogix 5000 XML...	50 KE
DataCopy_T2.L5X	11/14/2013 11:23 ...	RSLogix 5000 XML...	50 KE

File name: DataCopy_LEC.L5X

Files of type: RSLogix 5000 XML Files (*.L5X)

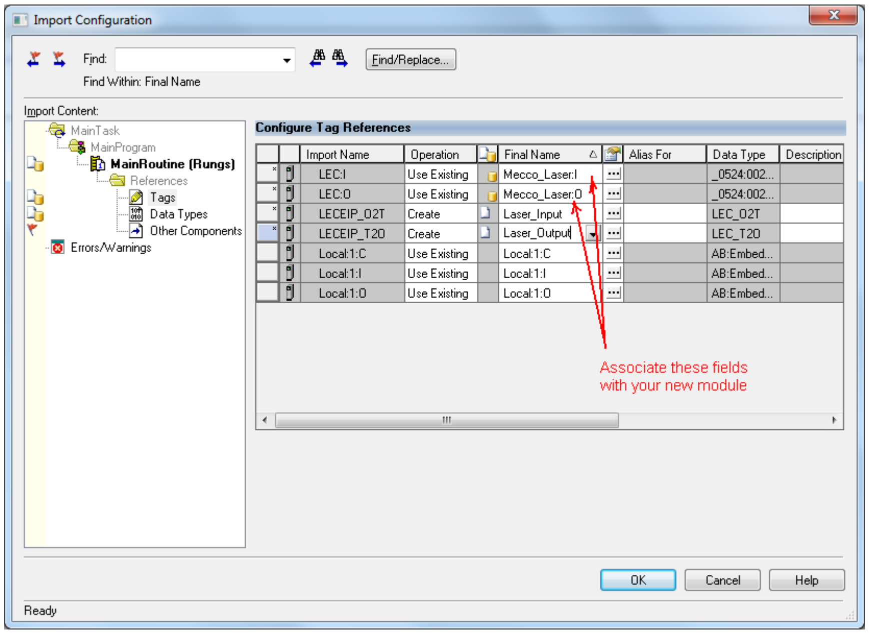
Files containing: Rungs

Into: MainRoutine (MainProgram)

Overwrite Selected Rungs

Figure 8 - Mecco Supplied Logic Rungs

10. On the Import Configuration dialog, select **Tags** in the option tree and associate LEC:I and LEC:O with the new module you created earlier. If configuring a dot-peen marker the "Import Name" labels will be T2:I and T2:O.

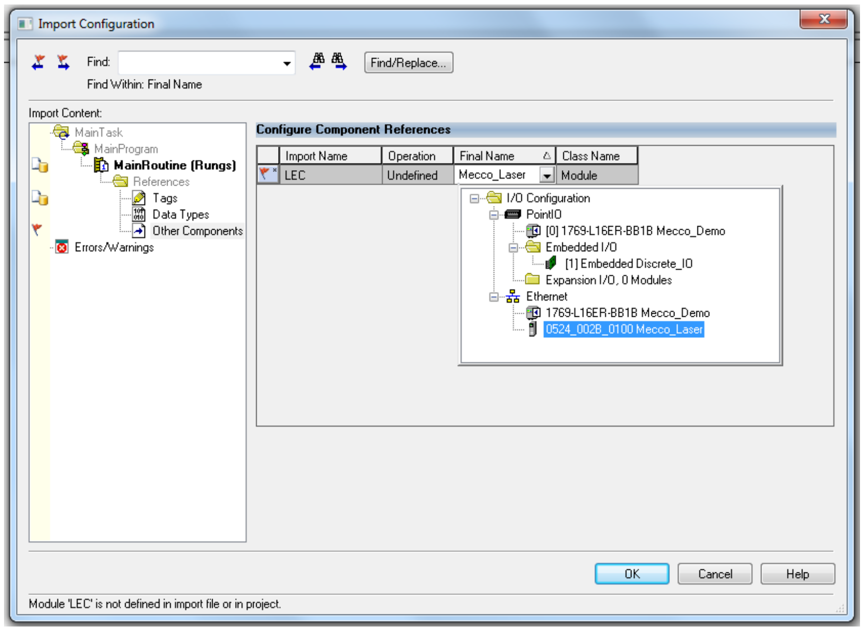


Configure Tag References

Import Name	Operation	Final Name	Alias For	Data Type	Description
LEC:I	Use Existing	Mecco_Laser:I		_0524:002...	
LEC:O	Use Existing	Mecco_Laser:O		_0524:002...	
LECEIP_O2T	Create	Laser_Input		LEC_O2T	
LECEIP_T2O	Create	Laser_Output		LEC_T2O	
Local:1:C	Use Existing	Local:1:C		AB:Embed...	
Local:1:I	Use Existing	Local:1:I		AB:Embed...	
Local:1:O	Use Existing	Local:1:O		AB:Embed...	

Figure 9 - Associate I/O with New Module

11. On the Import Configuration dialog, select **Other Components** in the option tree and associate your new module with the marker (LEC for laser; T2 for dot-peen).



Configure Component References

Import Name	Operation	Final Name	Class Name
LEC	Undefined	Mecco_Laser	Module

Module 'LEC' is not defined in import file or in project.

Figure 10 - Completing the Import Configuration

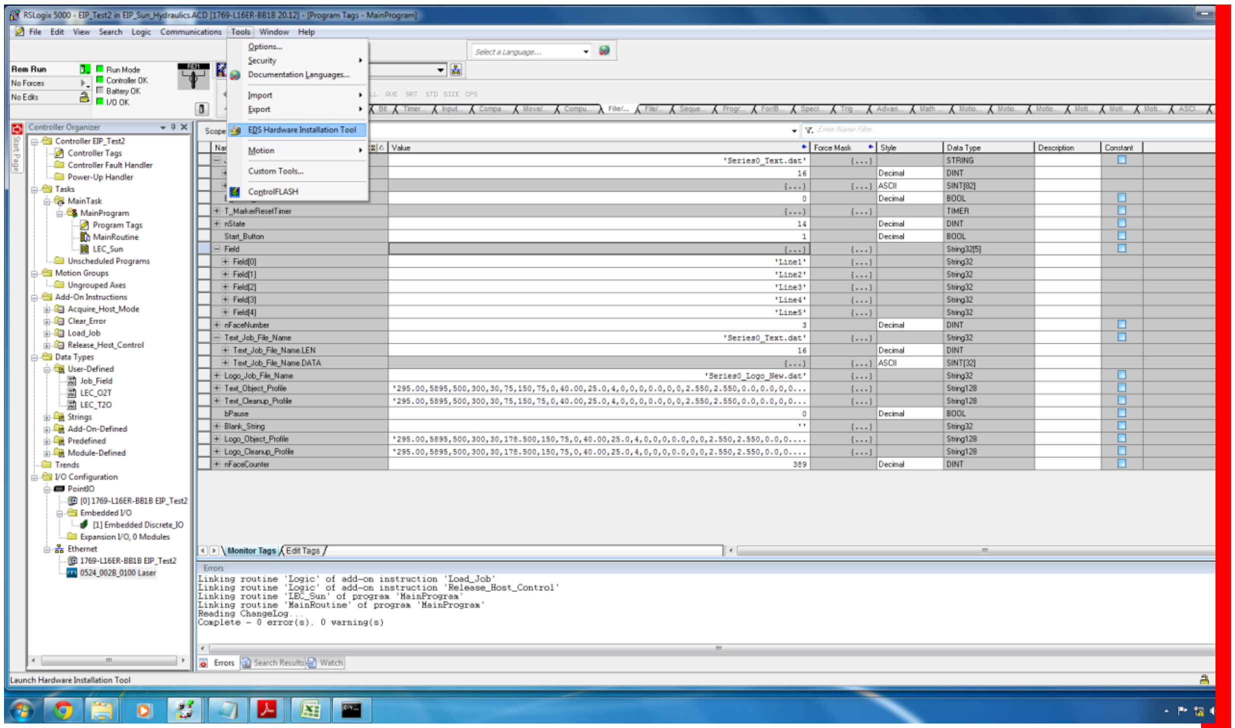
Creating the Template

The template for the laser marker is created using the WinLase. Refer to the WinLase LAN literature for specific instructions related to use of the application. WinLase provides support for the following options:

IMPORTANT NOTE

If specific fonts are required for your job, those fonts must be uploaded to the LEC card before sending the job to the laser.

- TrueType, OpenType and Laserfonts
- Graphic file import of DXF(versions 13, 14, 2000, 2004, 2007), DWG (versions13, 14, 2000), PLT, EMF, WMF, EX2, MCL, BMP, JPG, GIF, PCX
- Linear bar code types: Code39, CodaBar, Code93, Code128, Interleaved 2 of 5, POSTNET, UPCA, UPCE, EAN8, EAN13, BookLan
- 2D bar code types: DataMatrix, DensoQRCode, PDF417
- Extensive array of Automation objects: Wait For Input, Set Output, Time Delay, Message Box, XY Motion, Rotary Motion, Linear Motion, Serial Communication, Run Application, Alignment Tool, Laser Control
- Interfaces with intelligent motion controllers for multiple motion axis control
- Regional language support for: Chinese (Simplified), Chinese(Traditional), English (United States), French (France), German (Germany), Italian, Japanese, Korean, Spanish(Spain)
- Marker Library functionality is exposed in a COM Automation server interface (IAutomate and ILec)



Name	Value	Force Mask	Style	Data Type	Description	Constant
	'Series0_Text.dat'	{...}		STRING		☐
	16		Decimal	DINT		
	{...}	{...}	ASCII	SINT[82]		
	0		Decimal	BOOL		☐
T_MarkerResetTimer	{...}	{...}		TIMER		☐
nState	14		Decimal	DINT		☐
Start_Button	1		Decimal	BOOL		☐
Field	{...}	{...}		String32[5]		☐
Field[0]	'Line1'	{...}		String32		
Field[1]	'Line2'	{...}		String32		
Field[2]	'Line3'	{...}		String32		
Field[3]	'Line4'	{...}		String32		
Field[4]	'Line5'	{...}		String32		
nFaceNumber	3		Decimal	DINT		☐
Text_Job_File_Name	'Series0_Text.dat'	{...}		String32		☐
Text_Job_File_Name.LEN	16		Decimal	DINT		
Text_Job_File_Name.DATA	{...}	{...}	ASCII	SINT[32]		
Logo_Job_File_Name	'Series0_Logo_New.dat'	{...}		String32		☐
Text_Object_Profile	'295.00,5895,500,300,30,75,150,75,0,40.00,25.0,4,0,0,0,0.0,0,0,2.550,2.550,0.0,0.0,0,0...	{...}		String128		☐
Text_Cleanup_Profile	'295.00,5895,500,300,30,75,150,75,0,40.00,25.0,4,0,0,0,0.0,0,0,2.550,2.550,0.0,0.0,0,0...	{...}		String128		☐
bPause	0		Decimal	BOOL		☐
Blank_String	''	{...}		String32		☐
Logo_Object_Profile	'295.00,5895,500,300,30,178.500,150,75,0,40.00,25.0,4,0,0,0,0.0,0,0,2.550,2.550,0.0,0....	{...}		String128		☐
Logo_Cleanup_Profile	'295.00,5895,500,300,30,178.500,150,75,0,40.00,25.0,4,0,0,0,0.0,0,0,2.550,2.550,0.0,0....	{...}		String128		☐
nFaceCounter	389		Decimal	DINT		☐

```
Linking routine 'Logic' of add-on instruction 'Load_Job'
Linking routine 'Logic' of add-on instruction 'Release_Host_Control'
Linking routine 'LEC_Sun' of program 'MainProgram'
Linking routine 'MainRoutine' of program 'MainProgram'
Reading ChangeLog...
Complete - 0 error(s), 0 warning(s)
```

Figure 11 - Example Job Template

Register List

The register list defines the holding locations for storing commands that will be sent to the PLC or receiving responses back from the PLC. Registers may hold integers, strings, or arrays and may apply to the laser or the dot peen device only or to both. Refer to the tables that follow for complete descriptions of the available registers.

User Defined Data Types

Table 1 - LEC_O2T Data Type Definition

Name	Data Type	Description
X_Offset_mm_x100	INT	X axis offset in units of [millimeters times 100]
Y_Offset_mm_x100	INT	Y axis offset in units of [millimeters times 100]
Rotation_deg_x10	INT	Rotation in units of [degrees times 10]
X_Rotation_Center_mm_x100	INT	X coordinate of the center of rotation in units of [millimeters times 100]
Y_Rotation_Center_mm_x100	INT	Y coordinate of the center of rotation in units of [millimeters times 100]
X_Scale_x1000	INT	X scale factor in units of [scale times 1000]. For example, use 1000 for a scale of 1
Y_Scale_x1000	INT	Y scale factor in units of [scale times 1000]. For example, use 1000 for a scale of 1
Reserved	INT	Reserved
Marker_Command	INT	Numeric command from PLC to marker. See Table 4 - Marker Command Summary
Computed_Job_Field_Index	INT	The field index number for the computed field
Mark_Timeout_Sec	INT	The maximum time to allow for marking [seconds]. An error is generated if this time is exceeded.
Reserved_1	INT	Reserved
Barcode_Pass_Grade	INT	Pass grade for barcode verification
Max_Remarks	INT	Maximum number of remark attempts if camera verification fails
Short_Job_Field	Job_Field[5]	The field index numbers of up to five 32-character string fields
Long_Job_Field	Job_Field	The field index numbers of a single 64-character string field

Table 1 - LEC_O2T Data Type Definition

Name	Data Type	Description
Marker_Job_File_Name	String32	The name of the marker job file to load
Camera_Job_File_Name	String32	The name of the camera job file to load
Short_Field_String	String32[5]	Short string field values [32-characters]
Long_Field_String	String128	Long string field value [128-characters]

Table 2 - LEC_T2O Tag Definition

Name	Data Type	Description
Reserved	INT	
Last_Mark_Cycle_Time_Ms	DINT	The elapsed cycle time, or cycle time of the last mark in milliseconds
Status_Word	DINT	Laser: LEC Standard IO Word, Dot Peen: MC-2000 T2 Status Byte
Marker_Error_Code	INT	The marker error code (0 = No Error)
Controller_Error_Code	INT	The controller error code (LEC or MC-2000 T2) (0 = No Error)
Heart_Beat	DINT	Heartbeat indicator. Increasing number.
Input_Field_Checksum	DINT	Simple checksum of all input field bytes, Marker_Job_File_Name, and Camera_Job_File_Name
Echo_Job_File_Name	String32	An echo of the job file name that has been loaded
Results_Field	String128	A string results field generated by the marker
Marker_State_Desc	String64	A description of the current marker state
Marker_Error_Desc	String128	A description of the current marker error
Marker_State	INT	The current numerical state of the marker

Table 3 - Job Field Data Type Definition

Name	Data Type	Description
Field_Index	SINT	Object Number to which Field_String should write. On a Laser device, this will be the object number as listed in WinLase. On Dot Peen, this is the line number. 0 = The field is not used.
Uniqueness	SINT	1= Require a unique value for every mark, 0 = No uniqueness checking. If value is non-unique and field = 1, an error will be produced in the Marker_Error_Code register, and marking will not occur.
Verify	SINT	Bit 0 = Verify barcode against input field string Bit 1 = Verify OCR against input field string (applicable to long field string and short field string[0] only)
Justify	SINT	0 = Left Text Justification 1 = Center Text Justification 2 = Right Text Justification

Table 4 - Marker Command Summary

Command Number	Description	Status Number	Description
0	No Command	1	Marker Ready for Command
10	Acquire Host Control	19	Host Control Acquired
20	Release Host Control	29	Host Control Released
100	Load Job File	109	Job File Loaded
110	Load Multiple Jobs	119	Multiple Job File Loaded
150	Apply Laser Profile	159	Laser Profile Applied
160	Calculate Checksum	169	Calculate Checksum Complete
200	Prepare and Mark Job	209	Marking Complete
210	Prepare and Remark Job	219	Remark Job Complete
220	Prepare Job Only	229	Prepare Only Complete
230	Mark Job Only	239	Mark Only Complete
300	Abort Command	999	Aborted
500	Clear Concatenation String	509	Clear Concat String Complete
510	Concatenate String	519	Concatenate String Complete
600	Mark and Verify	609	Mark and Verify Complete
610	Verify Only	619	Verify Complete
999	Reset Error	999	Error

EtherMark Detailed Command Descriptions

Acquire Host Control, command number: 10

This command takes host control of the laser marker.   It is necessary to take host control of the laser marker prior to loading or marking job files.

Release Host Control, command number: 20

This command releases host control of the laser marker.   Host control of the laser marker should be released to allow other applications or hosts to control the marker.   For example, if off-line configuration of the marker is being performed using the PC based Winlase application. It is not necessary to release host control prior to powering down the marker.

Load Job, command number: 100

This command clears any/all previously loaded job file(s), and loads the job file specified in ***Marker_Job_File_Name*** into RAM. If the file does not exist, an error will be generated and the marker will enter the 999 state. Offsets and other transforms are applied (e.g., X_Offset_mmx100 = 100).

Load Multiple Jobs, command number: 110

This command is used to load multiple job files into RAM, on laser systems only.   Unlike command number 100, this command does not clear any prior jobs that were loaded into RAM.   The job file name is specified in ***Marker_Job_File_Name***. Offsets and other transforms are applied (e.g., X_Offset_mmx100 = 100).

Apply Laser Profile, command number: 150

IMPORTANT

NOTE

Refer to Winlase documentation for a qualitative description of profile definitions, mark mode, number of passes, and outline and fill.

The profile must be stored in the long string [128 bytes max]. The profile is applied either to a single object, as specified by the object index of the long string, **OR**, if the index of the long string is 0, to every mark-able object in the currently loaded job file.

Construct the profile as follows:

1. Create the profile in Winlase and **SAVE** it.
2. Open the .PRO file in a text editor. The first line should like similar to the following (although the profile file has many lines, only the first is used):

   P0=590.00,1475,500,800,30,253.7,125,50,13,40.00,12.0,68,22,11,44,33.0,0,0,0.000,0.000,0.0,0.0,0,0
3. Copy all of the information from the first line, after the prefix P0=. For example:

   590.00,1475,500,800,30,253.7,125,50,13,40.00,12.0,68,22,11,44,33.0,0,0,0.000,0.000,0.0,0.0,0,0
4. Append the values for the mark mode, the number of passes, and a flag to indicate if the profile should be applied to the object outline [0] or the object fill vector list [1].

   Mark Mode [0, 1, 2, 3, 4]
   0 = Mark Once
   1 = Mark Multiple
   2 = Two Pass Cut Clean
   3 = Three Pass Cut Clean
   4 = Four Pass Cut Clean

   Number of Passes (This property is only valid if Mark Mode is set to 'Mark Multiple')

   Outline or Fill Flag [0 or 1]

   Apply profile to outline = 0

   Apply profile to fill = 1

The final profile string should look similar to the following (the three appended attributes are underlined):

590.00,1475,500,800,30,253.7,125,50,13,40.00,12.0,68,22,11,44,33.0,0,0,0.000,0.000,0.0,0.0,0,0,<u>0,1,0</u>

Calculate Input Field Checksum, command number: 160

This command computes the simple sum of all of the string input fields, including the marker and camera job file names (***Marker_Job_File_Name*** and ***Camera_Job_File_Name***), the five short string fields (***Short_Field_String[ ]***), and the long string field (***Long_Field_String[ ]***). If a concatenated string field is defined, (long field = CONCAT) the concatenated string is used to compute the checksum. The result is returned in the ***input_field_checksum*** value. This function is also invoked automatically after the **Load Job File [100]**, **Prepare and Mark Job [200]**, **Prepare Job Only [220]**, And **Mark Job Only [230]** commands.

Prepare and Mark Job, command number: 200

This command prepares a job file for marking, and marks it. A job file must be loaded before this command is issued. For the laser only, which can accommodate multiple job files loaded in RAM, the job file name specified in ***Marker_Job_File_Name*** will be set as the active job file. For the dot peen, the most recently loaded job is active.

The real-time content of the short and long string fields will be sent to the marker. If specified, fields will be checked to insure that they have unique values.

The job will commence marking, and mark time will be monitored using the ***Mark_Timeout_Sec*** setting. If the mark exceed this time, an error will be generated and the marker will enter the 999 state.

The ***Last_Mark_Cycle_Time_Ms*** field will be updated in real time as the mark is in process. The marker will enter state 209 upon mark completion.

Prepare and Remark Job, command number: 210

The laser will mark only objects in the job file that are named with a suffix of **_REM** (CAPS required). Only objects without **_REM** suffix will be marked when in regular mark mode.

Prepare Job Only, command number: 220

This command prepares a job for marking, but does not initiate the mark. It is useful when cycle time is a priority, and allows the job to be prepared before the system may be ready to mark. A job file must be loaded before this command is issued. The real-time content of the short and long string fields will be supplied to the marker. If specified, fields will be checked to insure that they have unique values.

For the laser only, which can accommodate multiple job files loaded in RAM, the job file name specified in ***Marker_Job_File_Name*** will be set as the active job file. For the dot peen, the most recently loaded job is active.

Mark Job Only, command number: 230

The active job file will be marked immediately, without updating any field information. This command is intended to be used with **Prepare Job Only** (command 220) when more precise control of the preparation and marking process is required to minimize cycle time.

A job file must be loaded before this command is issued. The job will commence marking, and mark time will be monitored using the ***Mark_Timeout_Sec*** setting. If the mark exceed this time, an error will be generated and the marker will enter the 999 state. The ***Last_Mark_Cycle_Time_Ms*** field will be updated in real time as the mark is in process. The marker will enter state 239 upon mark completion.

Clear internal concatenation string, command number: 500

This command clears the internal concatenation string. The internal concatenation string is used when it is necessary to construct field information that is longer than 128 characters.

Concatenate string, command number: 510

This command concatenates, or appends, the ***Long_Field_String[0]*** to the internal concatenation string. This can be performed as many times as necessary to build a long string. It is recommended to clear the internal concatenation string using command number 500 first. The length limits are:

- 3000 characters maximum for the laser marker
- 75 characters maximum for the dot-peen marker

To specify use of the concatenated string when marking, first build the concatenated string. Next, set the long string field to the reserved command **CONCAT** (CAPS required). The Winlase object indexed by the long string will then be loaded with the internal concatenation string when the job is prepared. The field uniqueness requirement may be applied to the full length concatenated string. An internal copy of the previously used full length concatenated string is kept for this purpose.

Mark and Verify, command number: 600

This command performs a mark (exactly like command number 200), and then verifies the mark using a Cognex In-Sight® camera. The Cognex job file is specified in Camera_Job_File_Name. The IP address of the Cognex camera is a system flash setting. The number of attempted remarks is set in Max_Remarks and the passing barcode grade is set in Barcode_Pass_Grade. A special Cognex job file provided by Mecco is required, with preconfigured barcode and OCR fields. This job file must be loaded into the Cognex camera, the camera must be in the ONLINE state, and the job file's AcquireImage function must have its trigger parameter set to External, Manual, or Network.

The camera verification is performed after the mark is complete. Two (2) barcodes, and two (2) OCR fields may be verified, as configured in the Cognex job file. The long field (Long_Field_String[0]) and the first short field (Short_Field_String[0]) are used for vision verification. The .Verify attribute of these fields can be set as follows:

Bit 0 = Verify vision barcode result against input field string
Bit 1 = Verify vision OCR result against input field string

The long field corresponds to the first barcode and OCR result from the Cognex, and the first short field corresponds to the second set of barcode and OCR results. If the barcode string does not match, the barcode is below grade, or the OCR string does not match, a detailed marker verification error is generated. The results of verification are posted in the Results_Field.

The following is an example results field:

Verify error, BC1 match|BC1 pass grade: 3.00|OCR1 mismatch

(The long string is configured for a barcode and OCR check. The barcode content and grade passes, but the OCR field fails)

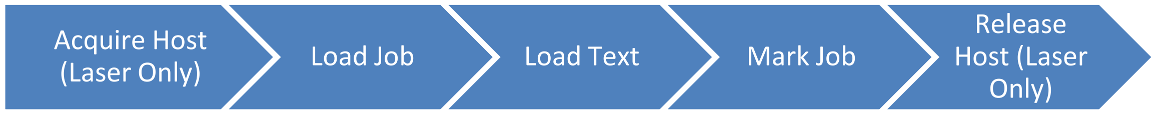
If the number of remarks is set to a non-zero value [Laser Only], and the first mark verification fails, a remark of the job file is attempted. See a description of the remark function above. A verification error is only generated once the permitted number of remarks has been exhausted.

Verify Only, command number: 610

To perform camera verification only, use command 610. The verification is configured exactly as described above for **Mark and Verify** (command 600), except no mark is performed.

Command Routine

Marking command sequences are nearly identical for the both Laser and Dot-Peen marking devices. Command Values are listed in brackets [ ]. Commands must be dropped ("0" command) before another command can be sent.



IMPORTANT

NOTE

Between each successive command, a value of 0 must be written to ***Marker_Command*** to drop the previous command.

The **Laser** routine is as follows:

1. Check for Ready [***Marker_State*** = 0] state
2. Send "Acquire Host Mode" Command [***Marker_Command*** = 10] (required for Laser only)
3. Check that host mode has been acquired [***Marker_State*** = 19] (required for Laser only)
   a) Send "0" Command, and Wait for Ready [***Marker_State*** = 1 (Host Mode Active)]
4. Send Job File to be loaded [***Marker_Job_File_Name*** = example.dat]
5. Send Load Job Command [***Marker_Command*** = 100]
6. Wait for Job to be loaded [***Marker_State*** = 109]
   a) Send "0" Command, and Wait for Ready [***Marker_State*** = 1]
7. Send Characters to be marked and set field object indices (if no data pushed, it will mark the last loaded data) [For example***: Short_Field_String[0]*** = "TEST MARK"; ***Short_Job_Field[0].Field_Index=1***]
8. Send Prepare and Mark Job Command [***Marker_Command*** = 200]
9. Wait for Mark to Complete [***Marker_State*** = 209]
   a) Send "0" Command, and Wait for Ready [***Marker_State*** = 1]

At this point you can either load a new job, send new strings, and continue the cycle. When a connection is to be terminated (required for laser only):

10. Send Release Host Mode Command [***Marker_Command*** = 20]
11. Wait for Host Mode Released [***Marker_State*** = 29]

IMPORTANT

NOTE

The ready state

The **Dot Peen** routine is as follows:

1. Check for Ready [***Marker_State*** = 0] state
2. Send Job File to be loaded [***Marker_Job_File_Name*** = example.txt]
3. Send Load Job Command [***Marker_Command*** = 100]
4. Wait for Job to be loaded [***Marker_State*** = 109]
    a) Send "0" Command, and Wait for Ready [***Marker_State*** = 0]
5. Send Characters to be marked and set field object indices (if no data pushed, it will mark the last loaded data) [For example***: Short_Field_String[0]*** = "TEST MARK"; ***Short_Job_Field[0].Field_Index=1 (to mark on line 1 of the Job File)***]
6. Send Prepare and Mark Job Command [***Marker_Command*** = 200]
7. Wait for Mark to Complete [***Marker_State*** = 209]
    a) Send "0" Command, and Wait for Ready [***Marker_State*** = 1]