



A Metrology Software Products Ltd document

ezLinkNC

Installation, Configuration and User Guide

- a guide for system administrators -

*Metrology Software Products Ltd
6J, Greensfield Court,
Alnwick, Northumberland,
ENGLAND, NE66 2DE*

Tel: (+44) 01665 608193

Fax: (+44) 01665 608194

[Email: postmaster@metsoftpro.com](mailto:postmaster@metsoftpro.com)

History

<i>When</i>	<i>Who</i>	<i>What</i>
<i>26/09/2007</i>	<i>Peter Hammond</i>	<i>Issue 1</i>
<i>27/09/2007</i>	<i>Peter Hammond</i>	<i>Added machine passoff results</i>
<i>25/09/2008</i>	<i>Chris Hobson</i>	<i>Siemens840d automatic mode</i>
<i>29/09/2008</i>	<i>Chris Hobson</i>	<i>FANUC automatic mode</i>
<i>24/04/2009</i>	<i>Chris Hobson</i>	<i>IPlusPlus mode</i>
<i>08/6/2009</i>	<i>Peter Hammond</i>	<i>Remove passoff results. Only Installation text</i>



1 Contents

1 Contents.....	2
2 Installation Guide.....	3
3 System Overview.....	4
3.1 Siemens 840D Configuration.....	4
3.2 Probing data and fitting results.....	5
3.3 Configuring ezLinkNC for the Siemens connection.....	6
3.4 Initial Powerup of remote PC and ezLinkNC startup.....	6
3.5 Exporting probing data from the controller to the remote pc.....	7
3.6 Importing an alignment to the controller.....	7
3.7 Configuring Siemens840d for Automatic mode.....	7
3.8 Configuring FANUC for Automatic mode.....	8
4 Configuring EZLinkNC to read from a serial port.....	9
4.1 Use of Perl.....	9
4.1.1 SerialInit.....	9
4.1.2 SerialOpened.....	9
4.1.3 SerialData.....	9
4.1.4 SerialClosed.....	9
4.1.5 SerialNoActivity.....	9
4.2 [Serial] configuration settings.....	10
4.2.1 ExportPerl.....	10
4.2.2 PerlPlugin.....	10
4.2.3 PerlParam.....	10
4.2.4 PerlInclude.....	10
4.2.5 PollTime.....	10
4.2.6 PortSetting.....	10
4.3 Session logging plugin.....	10
4.4 Supporting subroutines.....	11
4.4.1 SerialMessClear.....	11
4.4.2 SerialMessLine.....	11
4.4.3 SerialDie.....	11
4.5 Advice for writing/debugging Perl.....	11
5 Configuring ezLinkNC to act as an I++ proxy.....	13
5.1 Configuring the proxy server.....	13
5.2 Configuring the I++ proxy plugin.....	13

Illustration Index

Illustration 1: Siemens 840D configuration.....	4
---	---

Index of Tables

Table 1: Siemens controller directories.....	5
Table 2: Remote PC directories.....	5



2 Installation Guide

- 1) Please ensure that the target PC meets the minimum hardware specification and operating system requirements before attempting to proceed with an installation.
- 2) ezLinkNC can be installed by inserting the product CD and following the install program instructions.
- 3) By default the installation process creates a full installation of the software into the path:

C:\Program Files\msp\ezLinkNC\ezlinkNC_XXXXX

It will also appear in the startup menu. Start -> Programs -> Msp -> ezLinkNC -> ezLinkNC_XXXXXX

- 4) It is very important that the user accounts for the RemotePC have complete read/write access to the program directories and the directories described in section 3.2 above.

Typically the safest mechanism is to allow full access to the following installation directories :-

C:\Program Files\msp

and

C:\Program Files\delcam

3 System Overview

EzLinkNC provides an interface between a PC and a machine tool controller. The exact nature of the interface depends on the specific controller. However, the aim is to present at the PC data in a common format irrespective of the machine controller type.

This section summarizes the various controller configurations.

3.1 Siemens 840D Configuration

The Siemens 840D controller has a PC integrated into the controller. It is possible to connect to this PC via a standard ethernet connection. EzLinkNC takes advantage of this to provide a file-based transfer mechanism.

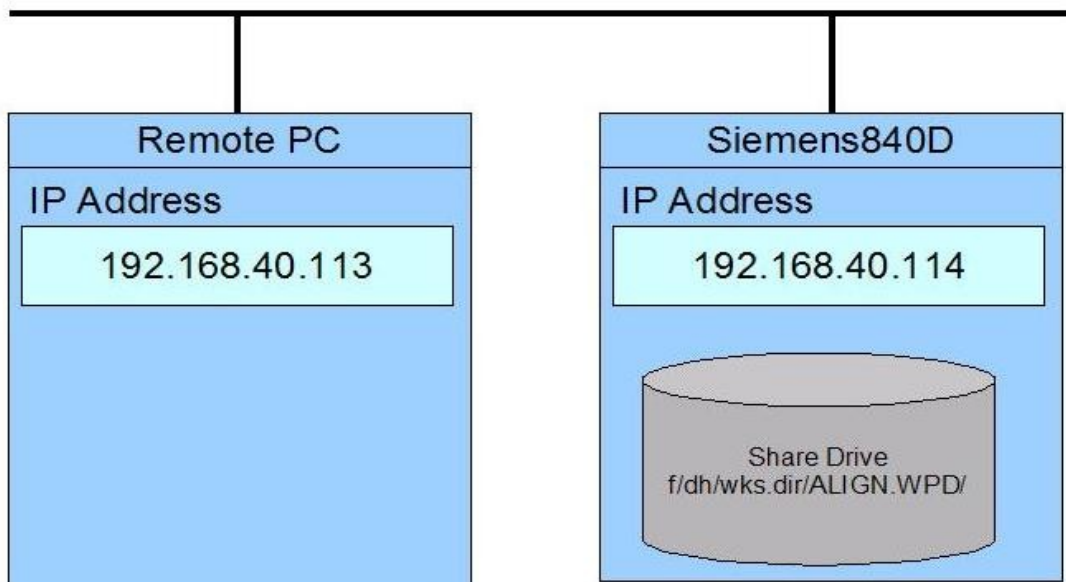


Illustration 1: Siemens 840D configuration

The controller and remote PC are connected by Ethernet. Both PC's are given IP addresses and are both members of the "Workgroup". The standard Siemens controller is configured such that the F: drive contains the data that is usable by the remote PC. This directory must be shared such the remote PC has access to it.

Configuring the Siemens controller requires someone familiar with the workings of the controller to configure the network. This document describes what is required and assumes that personnel familiar with the controller would be able to

For convenience it might be helpful on the remote PC to map the shared network drive on the controller PC however this is not essential.

3.2 Probing data and fitting results.

The probing cycle on the Siemens controller writes the probed data to a file. On completion of the probing the file is saved and made visible to the Remote PC. During the configuration it must be decided where this file is to be located. Similarly the Remote PC will also create a Siemens sub program file that comprises the result of PSFixture part alignment. Again the location of this must be decided.

The following tables gives an example of the file. In this case note the path includes the name of the controller pc, "Waricut". Also notice that both the probing data and the alignment paths are the same.

Siemens controller directories	
Probing data path	//Waricut/f/dh/wks.dir/ALIGN.WPD/Probing.spf
PsFixture alignment path	//Waricut/f/dh/wks.dir/ALIGN.WPD

Table 1: Siemens controller directories

EzLinkNC reads probing data from the controller and formats it into the standard MSP format that can be used by other MSP products. During this conversion the data is saved onto the local hard disk. The probing data files are given a unique name in the following format :-

PC_yyyymmddhhmmss.MSR for example PC_20070926174325.MSR

The process of transferring the data from the controller to the remote PC is known as Exporting the data from the controller and is actioned through the File->Export menu.

EzLinkNC configuration provides the facility to define the paths on the Remote PC into which the probing data from the controller is to be exported to.

Remote PC directories	
Probing data path	C:\Waricut\MSR

Table 2: Remote PC directories

3.3 Configuring ezLinkNC for the Siemens connection.

EzLinkNC is configured via file called ezlinknc.cfg which is located in the same directory as the ezlinkNC executable. (C:\Program Files\msp\ezLinkNC\ezLinkNC_30105 for example)

In the section above the paths where the data is to be read and written were defined. Below is an example configuration file with this information.

```
[Connection]
MachineType=Siemens840d
;Fanuc16i
;Siemens840d
AutoMode=false

[Siemens840d]
SPFProbingFile=//Waricut/f/dh/wks.dir/ALIGN.WPD/probing.spf
MSRToDir=C:\Waricut\MSR
ExportPerl=Siemens840d_Export.pl
;
; Used by "Export" to move file from Siemens PC, process and save in psFixture application
; runs Perl program specified by ExportPerl
;
FitFromDir=C:\Waricut\MSR
; Directory in which to store the fix result, Perl decides on the file
; name which at present is PsFix.spf
FitToDir=\\Waricut\fdh\wks.dir\ALIGN.WPD
ImportPerl=Siemens840d_Import.pl
;
; Use by "Import" to move file from psFixture to Siemens PC
; runs Perl program specified by ImportPerl
;
PollTime=200
;
; Time in milliseconds between polls for ProbingFile
;

[Fanuc16i]
PollTime=100 ; Defaults to 100 milliseconds
```

The file has been annotated to provide help on the various configuration parameters. The **AutoMode** parameter in the **[Connection]** section can be set to true to enable automatic operation for those **MachineType** that support the mode – See individual machines for details.

3.4 Initial Powerup of remote PC and ezLinkNC startup

The remote PC should not be powered up until the controller is powered. This allows the network connections to be established correctly between the remote PC and the controller.

Once powered start ezLinkNC. If the Remote PC does not have access to the Controller PC then the capture log will give warnings, for example :

```
* W *      SPFProbingFile directory //Waricut/f/dh/wks.dir/ALIGN.WPD/probing.spf not found
* W *      FitToDir Directory \\Waricut\fdh\wks.dir\ALIGN.WPD not found
* M *      Waiting for //Waricut/f/dh/wks.dir/ALIGN.WPD/probing.spf
```

The prefix W indicates a warning the M indicates an information message.

If the RemotePC does not connect to the controller PC then it may be for the following reasons :-

- 1) Ensure that both PC's are connected to the network. (Not some installations provide a network switch that allows the controller pc to be accessed by the machine manufacturer remotely.
- 2) The connection requires the remote PC to log in to the controller PC. To do this open up a windows explorer. Open up the My Network Places -> Microsoft Windows Network -> WorkGroup

Now click on the Controller PC. (for example "Waricut") you will be asked to provide a user and password. (For Siemens the default is "auduser" and "SUNRISE" note capitals!)

When you have connected to the controller PC close down ezLinkNC and restart. The two warnings should have disappeared.

3.5 Exporting probing data from the controller to the remote pc

During normal operation ezLinkNC is polling the Probing data path on the controller PC to see if a file is available and if it is it posts a message into the log to inform the user that a file is available for export.

A message is provided to indicate that ezLinkNC is waiting for a file and where it is looking.

The export button is greyed out until a probing file is available.

On pressing the export button the probing file is read from the controller and, if necessary reformatted to MSP's required format and saved to the probing data path of the remote PC with the correct unique name.

3.6 Importing an alignment to the controller

EzLinkNC operates in conjunction with Delcam Fixture and allows the result of the Fixturing operation to be imported into the controller. The import menu option, File, Import provides this facility.

After selecting this option a file select dialog is opened. This defaults to the directory where Delcam Fixture has written the alignment file. For the Siemens machine the file will have the suffix .spf.

Select from the list the correct file. It has the same unique number as the probing data file and additional part serial number information.

When the file has been selected it is transferred to the controller pc and a message is displayed in the capture log to indicate this.

3.7 Configuring Siemens840d for Automatic mode

The Siemens840d variant of ezLinkNC supports automatic mode, in this mode the Import and Export tasks occur automatically using trigger files. The existence of a trigger file is sufficient to initiate the required task. Text shown in **bold** refers to the *key* in the config file, and it is the *value* of the key that is used.

Export

The export action is triggered by the **SPFProbingFile** becoming available, this is the same file that allows a manual export. The Ridders format probing data is converted to a *timestamped* MSR file in the same way as a manual export. After the export a second trigger file is created to inform the NCConnector software that it can transfer the MSR data to the appropriate NCProduct for further processing. The path to the secondary trigger file is specified by the key **AutoExportTrigger** in the config file.

The Perl program which performs the export is passed the name of the **AutoExportTrigger** file as an optional parameter, this is always passed when in Automatic Mode and never passed when in Manual Mode. The Perl creates the specified file and writes two lines to it

[/path/to/created/MSR/PC_timestamp.msr](#)

%

this allows NCConnector to watch for the single trigger file as specified by **AutoExportTrigger**, but pass the generated file back to the NCProduct. The % on the second line allows the user of the trigger file to determine that the trigger file is fully formed. Note however that to prevent any timing issues the export Perl actually



creates the trigger file with a temporary name and then renames it to **AutoExportTrigger** when writing is complete.

It is the responsibility of NCConnector to delete the trigger file when it has passed the data back to the archive of the NCProduct.

Import

The import action is trigger by the **AutoImportTrigger** file becoming available. As with export, the trigger file must contain two lines

[/path/to/created/PC_timestamp.spf](#)

␣

When a fully formed **AutoImportTrigger** becomes available ezLinkNC automatically runs the import Perl, in exactly the same way as if the import had been run manually. Note that when running in automatic mode the **FitFromDir** key is not used (In manual mode this is used to determine the default directory that the file selector opens in)

3.8 Configuring FANUC for Automatic mode

The FANUC variant of ezLinkNC supports automatic mode, in this mode the Import and Export tasks occur automatically.

Export

The export action is trigger by the probing program simulating a probe trigger with a point number of 0. When this happens the name of MSR file that has been created, in the Capture directory, written to the **AutoExportTrigger** file. As with the Siemens840d two lines are written to the **AutoExportTrigger**

[/path/to/created/MSR/PC_timestamp.msr](#)

␣

It is the responsibility of NCConnector to delete the trigger file when it has passed the data back to the archive of the NCProduct.

Note that the capture directory is specified by the **LocalPath** key in the **[CAPTURE]** section of the config file.

Import

The import action is trigger by the **AutoImportTrigger** file becoming available. As with export, the trigger file must contain two lines

[/path/to/created/PC_timestamp.fit](#)

␣

When a fully formed **AutoImportTrigger** becomes available ezLinkNC automatically parses the file and updates the registers on the controller. EzLinkNC then deletes the trigger file.



4 Configuring EZLinkNC to read from a serial port

The configuration file **ezlinknc.cfg** is located in the same directory as the executable, **EZLinkNC.exe**, the `<installdir>`. The configuration file must be edited and the **[Connection]** section located, the key **MachineType** must be set to have the value **Serial**.

```
[Connection]
MachineType=Serial
```

Once this change has been made EZLinkNC will read a second configuration file `<installdir>/common/cfg/CommPortRecieve.cfg` which contains the settings for the serial port that is to be read. The parameters should be self explanatory

```
[COMMSETUP]
CommPort=COM1
BaudRate=2400
DataBits=8
Parity=0
StopBits=2
XOnXOff=true
```

The configuration file is read each time a File->Connect operation is performed and so there is no need to exit EZLinkNC to trial new settings, simply disconnect and Connect again. EZLinkNC will automatically attempt to Connect when it is started.

4.1 Use of Perl

EZLinkNC has the ActiveState version of the Perl interpreter embedded within it, this allows unlimited flexibility in data formats that can be processed because EZLinkNC routes all serial port activity through a Perl program.

A default Perl program is provided which allows a custom Perl module to be loaded, it is the module which performs the main processing. In normal use there should be no need to alter the standard Perl program, instead write custom modules, or plugin. A plugin must provide the following subroutines, which are called by the main program

4.1.1 SerialInit

This is called after the Perl interpreter has been started, and before the connection to the Serial port is established. The subroutine is passed a parameter which is read from the configuration file, see later. This subroutine can perform initialization of the module.

4.1.2 SerialOpened

Called when the port has been opened, the subroutine is passed a parameter which names the serial port. After this subroutine has been called the plugin should be ready to receive and process the stream of data coming from the serial port.

4.1.3 SerialData

Called each time a packet of data is read from the serial port. The data is passed as a parameter and the module is responsible for processing the data, it may be necessary to cache the data until there is sufficient to process.

4.1.4 SerialClosed

The connection has been closed, the Perl program will be unloaded when this subroutine returns and so the module must finalise any output and close down. As with SerialOpened the subroutine is passed the name of the serial port that is closing.

4.1.5 SerialNoActivity

EZLinkNC reads the serial port in a separate thread to the main application thread, the main application polls this message queue at specified intervals. (See later). If there has been no activity since the last poll then this



subroutine is called. The subroutine may decide to do nothing, or it may alert the user after a given time period, the action to take is to be decided by the plugin author.

4.2 [Serial] configuration settings

The **ezlink.cfg** file has a section named **[Serial]** which contains all of the setting used to customise the Perl. All of the standard Perl shipped with EZLinkNC is located in the `<installdir>/common/scripts` directory. It should not normally be necessary to either edit these files, or place custom plugins into that directory. Custom plugins can be placed in a vendor specific directory. Each of the parameters in the **[Serial]** section are described below

4.2.1 ExportPerl

This parameter defaults to **ExportPerl=Serial_Export.pl** and shouldn't need changing, it defines the Perl program that is run each time serial activity occurs. The program is clearly documented to explain how it works.

4.2.2 PerlPlugin

This parameter names the Perl module which supplies the implementation of the subroutines described earlier. It defaults to **PerlPlugin=Serial_Template.pm** and that file can be used as a guide to writing a custom plugin. The default plugin causes messages to be displayed in the EZLinkNC command window, showing a guide to the serial port activity.

4.2.3 PerlParam

The value of this parameter is passed to the SerialInit subroutine when the Perl program is first loaded. By default it has no value and so an empty string is passed to SerialInit. This parameter can be used, for example, to pass a directory name where a plugin is to store data files.

4.2.4 PerlInclude

Perl allows modules to be located in a series of include directories, when running Perl at the command line include paths are specified via a `-I` argument. EZLinkNC always runs Perl with an include of `<installdir>/common/scripts` so that the MSP supplied modules can be located. The **PerlInclude** parameter allows a semi-colon separated list of vendor specific include directories to be specified. It is in these directories that custom plugins and modules should be located.

4.2.5 PollTime

This specifies the time, in milliseconds, that EZLinkNC polls its internal event queue for serial activity. The serial port is being read constantly and any data is queued for processing when the next **PollTime** occurs. If data has been read then the **SerialData** subroutine will be called, otherwise **SerialNoActivity** is called. The parameter default to **PollTime=200** leading to 5 polls per second.

4.2.6 PortSetting

This parameter names the configuration file that holds the settings for the serial port itself. It defaults to **PortSettings=common/cfg/CommPortReceive.cfg** but can be changed to the full path of another file, if required.

4.3 Session logging plugin

A standard plugin called **Serial_SessionLogger.pm** is supplied, it stores all of the serial port events into a binary session file. The session file can then be replayed via the **Serial_SessionPlayer.pl** Perl program, allowing offline development of a plugin. When using the session logger the **PerlParam** parameter must be set to the name of a directory to store the session files in. Session files are created with a time stamped name, and an **mss** extension of the format **yyyymmddhhmmss.mss**

The program **Serial_SessionPlayer.pl** has comments explaining how it is used.

4.4 Supporting subroutines

The standard main program **Serial_Export.pl** supplies standard subroutines which allow a plugin to pass data back to the EZLinkNC command window.

4.4.1 SerialMessClear

Causes the EZLinkNC command window to be cleared, when the Perl program returns. New text can be added via SerialMessLine immediately after a SerialMessClear if required.

4.4.2 SerialMessLine

Causes a line of text to be displayed in the EZLinkNC command window, the subroutine takes an array of arguments, argument maps to a field in the EZLinkNC command window. If an argument is of zero length, "", then "----" is displayed in the field. If this behaviour is not required then specify a single space for the argument, this will cause nothing to appear in the field.

SerialMessLine can be called multiple times, to display more than one line. Due to the way data is passed back to EZLinkNC it is important to note that the pipe (|) and semi-colon (;) characters cannot be displayed. (These characters are used as line/field separators when passing the data back to EZLinkNC).

Due to the way Perl processes subroutine arguments you can either pass an array, or multiple arguments to SerialMessLine

```
SerialMessLine(@l_Args);
SerialMessLine("N$l_N", "", $l_X, $l_Y, l_Z);
```

The second method is probably the more obvious and natural to use in plugins.

4.4.3 SerialDie

This should be used when a fatal error occurs in a plugin and it wishes to abort, it is basically a wrapper to the standard Perl **die** but allows the error message to be routed back to EZLinkNC.

4.5 Advice for writing/debugging Perl

Running and debugging Perl is an advanced topic, trying to write Perl modules and have them run error free first time in EZLinkNC is unlikely to be successful. It is easier to use the SessionLogger to capture the serial port data and then work outside of EZLinkNC until the plugin is working correctly.

EZLinkNC ships with the command line version of ActiveState Perl, this is *<installdir>/Perl.exe* which can be used outside of EZLinkNC. All of the standard Perl modules are located in *<installdir>/modules/Perl*, if your plugin uses any standard Perl modules you need to make these available to the Perl program via the *-I* argument.

A typically command would be

```
<installdir>/perl -I<installdir>/modules/perl -I<installdir>/common/scripts
-I/your/modules <installdir>/common/scripts/Serial_SessionPlayer.pl "params"
Plugin.pm /path/to/log.mss
```

clearly this is not trivial and a batch or script file is recommend to automate this task.

A graphical debugger for Perl is available from CPAN (Comprehensive Perl Archive Network).

A brief summary of instructions would be

1. Search <http://search.cpan.org> for ptkdb
2. Download the file, and open it with Winzip
3. Extract ptkdb.pm to *<installdir>/modules/perl/devel/ptkdb.pm*
4. Start Perl with the new command line shown below



```
<installdir>/perl -d:ptkdb -I<installdir>/modules/perl  
-I<installdir>/common/scripts -I/your/modules  
<installdir>/common/scripts/Serial_SessionPlayer.pl "params" Plugin.pm  
/path/to/log.mss
```

5 Configuring ezLinkNC to act as an I++ proxy

The section is under development and is subject to review

ezLinkNC can be configured to listen for I++ connections on a TCP/IP port and can modify the commands/replies being sent to/from a real I++ server. The following changes are required to the ezlinnc.cfg file located in the same directory as the executable

5.1 Configuring the proxy server

Set up to act in I++ mode by specifying the machine type as IPP

```
[Connection]
MachineType=IPP
```

Then configure the TCP/IP port number on which to listen to I++ connections, note that the I++ standard uses port 1294

```
[SERVER]
PORTS=1294
1294=IPLUSPLUS
```

The PORTS entry defines that ezLinkNC will monitor that TCP/IP port for incoming connections, the section line defines that the port is used for the I++ protocol – both lines must be present and both must contain the appropriate port number (In the above example the standard 1294 is used).

The scheme is identical to that used in uccServer and is arranged so that in the future other ports can be added for different protocols.

It is normal to configure ezLinkNC to monitor the standard port so that client software need not be changed, however the real I++ server must then be changed to monitor a different port (Assuming ezLinkNC and uccServer are running on the same computer).

A second set of config entries are then used to inform ezLinkNC of the connection details of the I++ server and also to fine tune the settings of ezLinkNC

```
[IPP]
PORT=1293
;HOST=127.0.0.1
;RETRY=3
;RETRYINTERVAL=5.0
;PollTime=200
```

PORT defines the port number on which ezLinkNC attempts to contact the I++ server. HOST allows a different IP address from the local host to be given (Default = Localhost). RETRY and RETRYINTERVAL specify how many times connection is attempted and the interval between retries – before failure is reported.

PollTime is specific to ezLinkNC and it defines how long between starting ezLinkNC and attempting to start listening for connections. The default is fine for most usage and the parameter may be used for other things at a later date.

5.2 Configuring the I++ proxy plugin

The I++ proxy has the facility to route the I++ traffic through one or more plugins. The plugin is configured in the ezLink.cfg file.

The Standard installation includes the I++ comparator plug this would be configured as follows.

```
[IPP]
PlugIn1=..\install\plugins\ezIPPCompare.dll
```