

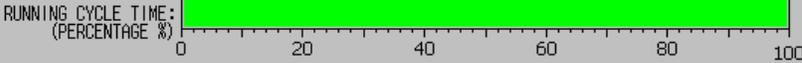
eHMI Scripting & Functionality Library

eHMI with MEL-FACS

Version D1.0

Document: eHMI Scripting & Functionality Library

NO CYCLE	NO MODE	HOME	WORK COMPLETED	FIELDBUS LAYOUT	ENTER STATION NAME HERE		56/56/56 09:10															
ALARM 1						03/01/13 09:10:33																
MODEL TYPE: <input type="text" value="ABCDEFGHIJKLMN"/>		PALLET NUMBER: <input type="text" value="****"/>																				
<div style="display: flex; flex-direction: column; align-items: center;"> <div style="margin-bottom: 10px;">▲▲</div> <div style="margin-bottom: 10px;">▲</div> <div style="margin-bottom: 10px;">▼</div> <div style="margin-bottom: 10px;">▼▼</div> </div>	BUILD/NO	BUILD	STATUS	TASK MODE	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">CYCLE PROGRESS</td> <td rowspan="16" style="text-align: center; vertical-align: middle;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">ENABLE TASK</td></tr> <tr><td style="text-align: center;">BYPASS (PART OK)</td></tr> <tr><td style="text-align: center;">CONT BYPASS (PART NOK)</td></tr> <tr><td style="text-align: center;">RESET TASK</td></tr> <tr><td style="text-align: center;">REJECT PART</td></tr> <tr><td style="text-align: center;">RESET FACS SEQUENCE</td></tr> </table> </td> </tr> <tr><td style="text-align: center;">RF READ CMP</td></tr> <tr><td style="text-align: center;">ALL TASKS DONE</td></tr> <tr><td style="text-align: center;">CYCLE COMPLETE</td></tr> <tr><td style="text-align: center;">RF WRITE CMP</td></tr> <tr><td style="text-align: center;">FACS UPDATED</td></tr> <tr><td style="text-align: center;">RDY TO RELEASE</td></tr> </table>			CYCLE PROGRESS		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">ENABLE TASK</td></tr> <tr><td style="text-align: center;">BYPASS (PART OK)</td></tr> <tr><td style="text-align: center;">CONT BYPASS (PART NOK)</td></tr> <tr><td style="text-align: center;">RESET TASK</td></tr> <tr><td style="text-align: center;">REJECT PART</td></tr> <tr><td style="text-align: center;">RESET FACS SEQUENCE</td></tr> </table>	ENABLE TASK	BYPASS (PART OK)	CONT BYPASS (PART NOK)	RESET TASK	REJECT PART	RESET FACS SEQUENCE	RF READ CMP	ALL TASKS DONE	CYCLE COMPLETE	RF WRITE CMP	FACS UPDATED	RDY TO RELEASE
	CYCLE PROGRESS		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">ENABLE TASK</td></tr> <tr><td style="text-align: center;">BYPASS (PART OK)</td></tr> <tr><td style="text-align: center;">CONT BYPASS (PART NOK)</td></tr> <tr><td style="text-align: center;">RESET TASK</td></tr> <tr><td style="text-align: center;">REJECT PART</td></tr> <tr><td style="text-align: center;">RESET FACS SEQUENCE</td></tr> </table>	ENABLE TASK				BYPASS (PART OK)	CONT BYPASS (PART NOK)		RESET TASK	REJECT PART	RESET FACS SEQUENCE									
	ENABLE TASK																					
	BYPASS (PART OK)																					
	CONT BYPASS (PART NOK)																					
	RESET TASK																					
	REJECT PART																					
	RESET FACS SEQUENCE																					
	RF READ CMP																					
	ALL TASKS DONE																					
	CYCLE COMPLETE																					
	RF WRITE CMP																					
	FACS UPDATED																					
	RDY TO RELEASE																					
	01	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																				
02	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					
03	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					
04	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					
05	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					
06	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					
07	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					
08	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					
09	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					
10	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					
11	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					
12	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					
13	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					
14	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					
15	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst																					



PROMPT 1

TASK STATUS	MANUAL	DIAGNOSTIC	FAULT	PRODUCTION	MAINTENANCE	REPAIR	INFORMATION
-------------	--------	------------	-------	------------	-------------	--------	-------------

Version D1.0

Original Date: 02-12-2013

Revised Date: 06-06-2016

Document Version History

Date	Version No.	Document Name	Revision	Revised By
March 05, 2013	D0.1	Scripting & Functionality - eHMI	First Draft	eFlex Systems - Integration Team
June 06, 2016	D1.0	Program Book & Guidelines - eHMI	Modifications from May 2016 program changes.	eFlex Systems – Integration Team

DRAFT

Content

Section 1)

An Introduction to the eHMI.....	5
---	----------

Section 2)

Required Project Settings.....	6-9
2.1 Option Operating System	6
2.2 Controller Setting.....	8

Section 3)

Project Scripts.....	10-16
3.1 eHMI Cyclical.....	10
3.2 eHMI Clear Cache.....	10
3.3 eHMI Main Screen	11
3.4 eHMI Task Unlock	11
3.5 eHMI Tasks A.....	12
3.6 eHMI Tasks B.....	13
3.7 eHMI Tasks C.....	13
3.8 eHMI Tasks D	14
3.9 eHMI Sensor Descriptions.....	15
3.10 eHMI Model Descriptions	15
3.11 eHMI Cycle Time	16
Screen Scripts	17-21
3.12 eHMI Stitch Task Switching.....	17
3.13 eHMI Socket Task Switching	17
3.14 eHMI Multi Task Switching	18
3.15 eHMI Camera Task Switching	18
3.16 eHMI Generic Task Switching	19
3.17 eHMI Barcode Task Switching	19
3.18 eHMI Lube Task Switching	19
3.19 eHMI Test Task Switching	20

3.20 eHMI Press Task Switching 20

3.21 eHMI Robot Task Switching 21

DRAFT

1 An Introduction to the eHMI

Scope of this document:

This document was created to assist the system integrator in obtaining a deeper understanding of how the eHMI works. In this document you will read through detailed descriptions of what each function does, and where its application is. Beyond this, we will look into how the script performs many of the functions asked of the eHMI standard. As well as seeing GOT configuration settings and devices required to make everything work with your application of MelfACS.

What the eHMI will do for you:

The eHMI is an application that runs on a standard dedicated hardened HMI not a PC based HMI, i.e. Mitsubishi GOT (The current supported range is the GOT2000 series). Our eHMI was developed to leverage the dynamic flexible environment of eFlex Assembly to support Lean flexible assembly. Previously things like symbolic information and communications addressing were hardcoded into the HMI which the eHMI platform now eliminates. The eHMI has a master user screen that presents the operator with all of the important information required for their work in station i.e. Task indicators and sequencing, Part status, task status, cycle time, model code, and manual operation buttons for all of the different task types that are configured in eFlex for that station. With this, it is now possible to consistently mimic what is seen in eFlex Assembly on the HMI. As changes are made in the eFlex configuration, they are instantly reflected in all subsystems including eHMI. This makes life easier for all users: Production, engineering and maintenance. Just like we have common logic for the stations, we can now have a universal baseline for all HMI projects that is tied into eFlex Framework. OEMs now have an HMI that is plug and play ready to support any MelfACS line. They will take our eHMI package and add on as they see fit. Everyone on the plant floor now has the ability to easily see exactly what the configuration, and status is in each station via the eHMI.

2 Required Project Settings

2.1 Option Operating System

In order to successfully run the eHMI GOT package, there are a few common settings that must be in place prior to downloading your project. If you receive the project file from eFlex Systems, most of the settings covered in this chapter will already be set.

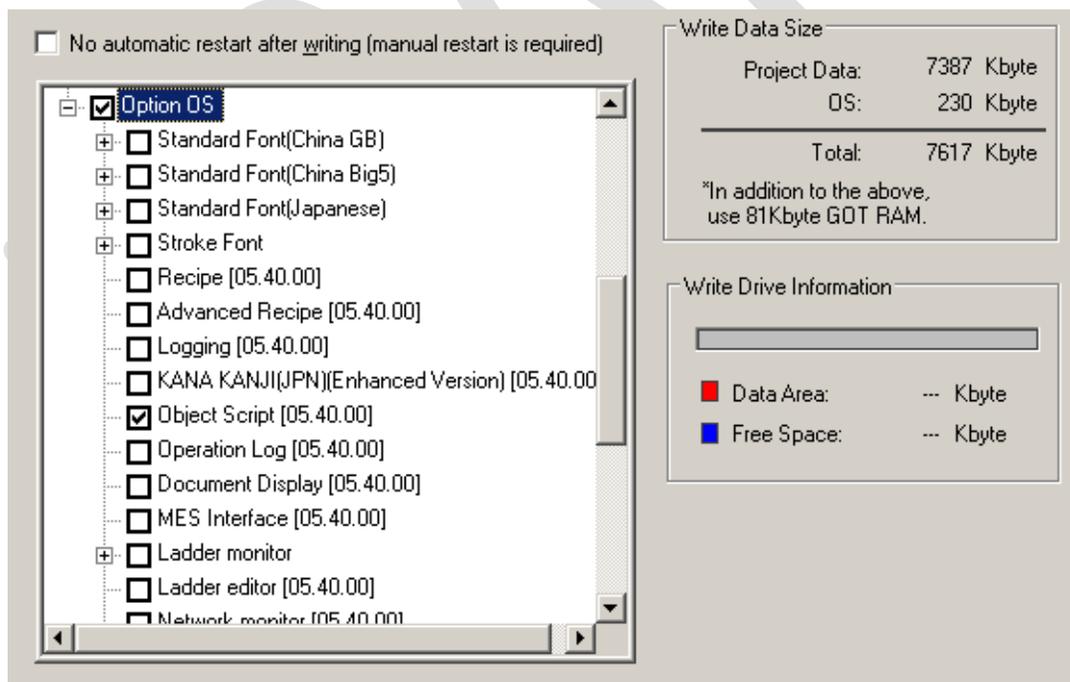
The first step's settings do not stay consistent in the project. When downloading a project, regardless of where it came from, you will need to download specific optional operating systems.

- **Note that the entire project directory should be written to the C drive.**



1) The first of which is the Object Script Option OS.

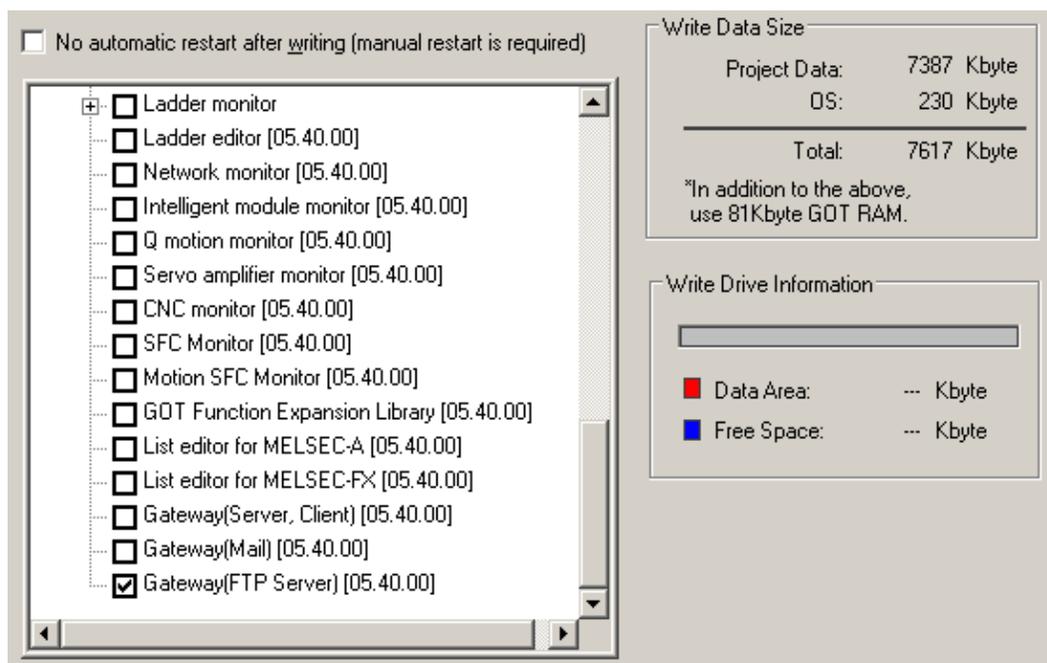
- The Object Script Option OS is what allows screen objects to perform functions much more powerful than what can be performed with just the standard object settings. This is a must have, with or without the eHMI package.



(Continued on the next page)

2) The second Option OS that must be installed is the FTP Server Option OS.

- The FTP Server Option OS is what allows the GOT to accept incoming packets of data from the eFACS server.
- This is how text files generated in eFACS, are sent over to the GOT.
- (We will look at the FTP Server setting in section 2.2)



Other Important Information:

The procedure outlined above only works for models of GOT with the GOT2000 designation. If you are using a 1000 series of GT-15** or lower, you will need an additional expansion option function board. Along with the function board you will need greater memory for the main project directory. This function board is built into the GOT2000 series, which also have greater standard memory.

2.2 Controller Setting

In order for eFACS to be able to communicate with the GOT, and send it the required text files, you must activate the FTP server setting.

Procedure:

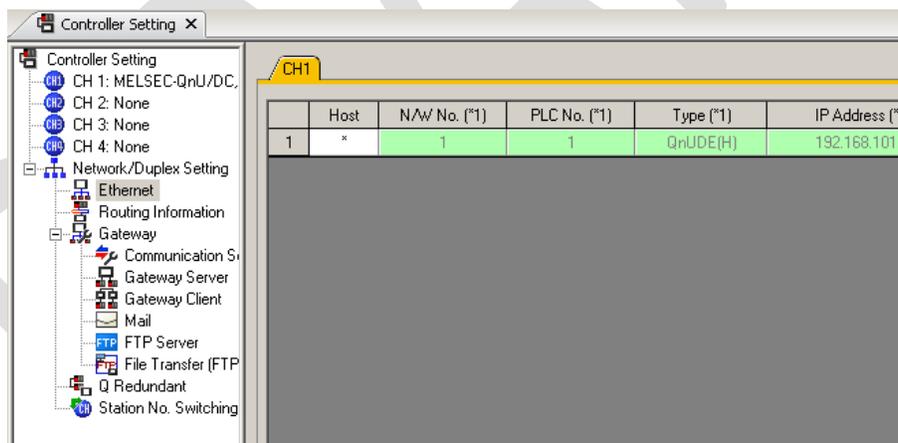
Open the eHMI GT Designer 3 project file and find the “Controller Setting” under the “System” pane of your browser.

Expand “Controller Setting”

Expand “Network/Duplex Setting”

Under Ethernet:

- Make sure your configuration settings match what is configured in MELSOFT Navigator
- If they do not match, you must make sure to properly configure the Navigator GOT settings and then reflect the parameters to the controller.
- If they still do not match, navigate back into GT Designer 3. Under the Tools menu bar, click the “Option” setting. From here under the iQ Works Interaction tab, make sure that “Enable an editing of parameters set in MELSOFT Navigator” is not selected.
-



- **(Note that the green hue is indicative of a locked connection between MELSOFT Navigator and GT Designer 3.)**

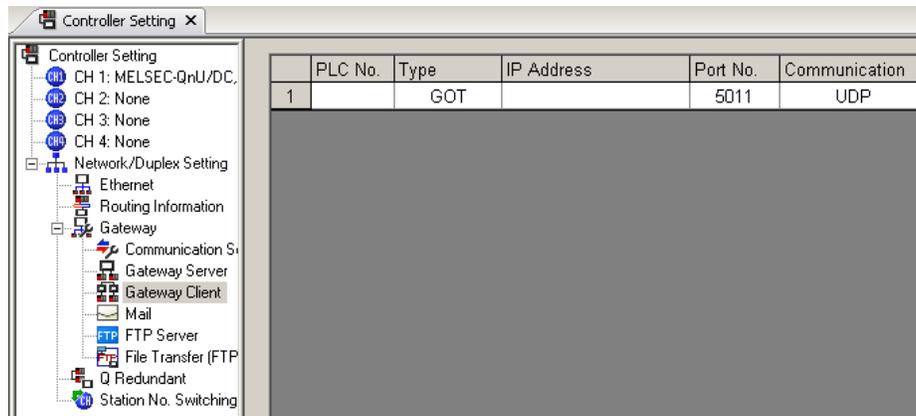
Once your Ethernet settings are correct, you can proceed to the next step.

(Continued on the next page)

Expand “Gateway” and enter “Gateway Client”

Under Gateway Client:

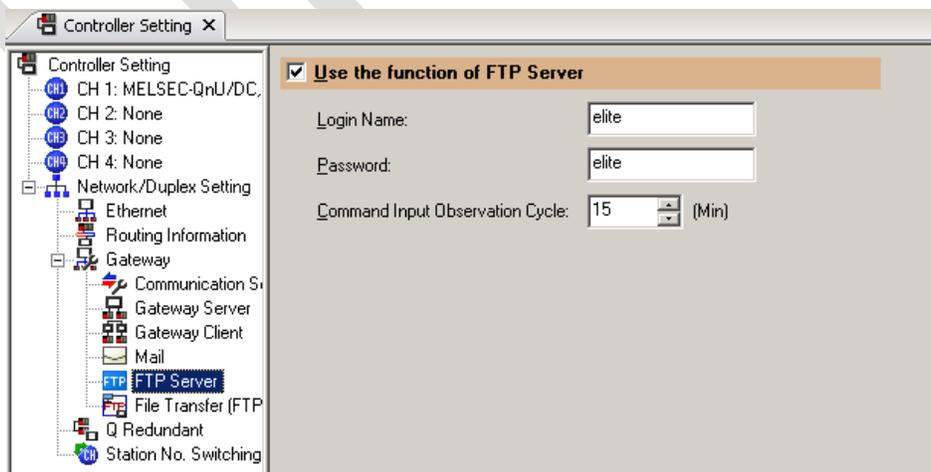
- Make sure that the “Type” field is set to GOT.
- Make sure the “Port No.” is 5011, and the “Communication” field reads UDP.
- Press “Apply” followed by “OK”.



The final step is to configure the FTP Server Settings.

Still under Controller Setting, enter the FTP Server window:

- Confirm that the checkbox at the top of the screen entitled “Use the function of FTP Server” is checked.
- Confirm that the Login Name and Password fields both read “elite”.
- Set the Command Input Observation Cycle to 15 minutes.
- Press “Apply” followed by “OK”.



3 Project Scripts

3.1 “eHMI Cyclical”

This script checks to see if new eHMI data exists, if there are any valid triggering conditions for updating the eHMI system, and manages the update process. This script will detect when `eHMI_New.csv` arrives, and rename it to `eHMI.csv` if the new file is found. The triggering conditions for updating the eHMI system are listed below.

- System first scan
- Operator select new footprint
- Version number mismatch (new `eHMI.csv` file present)
- Manual trigger

Once an update request is detected, this script will call the next script by setting GB63010 (See section: “3.2 eHMI Clear Cache”). If the update process is in sequence, this script manages the interrupt (from eHMI Maintenance screen), and checks to see when the update sequence is complete.

The figure below shows project script setting information.

The screenshot shows a 'Script Edit' window with the following fields and buttons:

- Order: 1
- Script No.: 920 (with a 'View...' button)
- Script Name: eHMI Cyclical (with 'Import...' and 'Edit Script...' buttons)
- Comment: eHMI Cyclical (with 'Syntax Check' button)
- Data Type: Signed BIN16 (dropdown menu)
- Trigger Type: Ordinary (with 'Trigger Setting...' button)
- Buttons: OK, Cancel

3.2 “eHMI Clear Cache”

This script clears out internal GB and GD registers that are used by the eHMI system. Once complete, the script will trigger the next required script by setting GB63011 (See section: “3.3 eHMI Main Screen”).

The figure below shows project script setting information.

Script Edit

Order: 2

Script No.: 923

Script Name: eHMI Clear Cache

Comment: eHMI Clear Cache

Data Type: Signed BIN16

Trigger Type: Rise

Trigger Device: GB63010

3.3 “eHMI Main Screen”

This script reads data from `eHMI.csv`. The data is intended for Main Screen use only. Once complete, the script will trigger the next required script by setting GB63012 (See section: “3.4 eHMI Task Unlock”).

The figure below shows project script setting information.

Script Edit

Order: 3

Script No.: 930

Script Name: eHMI Main Screen

Comment: eHMI Main Screen

Data Type: Signed BIN16

Trigger Type: Rise

Trigger Device: GB63011

3.4 “eHMI Task Unlock”

This script reads data from `eHMI.csv`. The data is used to control access to task maintenance screens. Once complete, the script will trigger the next required script by setting GB63013 (See section: “3.5 eHMI Tasks A”).

The figure below shows project script setting information.

3.5 “eHMI Tasks A”

This script reads data from `eHMI.csv`. The data is used to populate the follow task maintenance information (accessible from the task maintenance screens).

- Station Tasks
- Stitch Tasks
- Multispindle Main Tasks
- Pick Tasks
- Error Proofing Tasks
- Press Tasks
- Lube Tasks
- Test Main Tasks
- Camera/Vision Main Tasks
- Barcode Main Tasks
- Robot Tasks
- Generic Tasks
- Socket Main Tasks

Once complete, the script will trigger the next required script by setting GB63014 (See section: “3.6 eHMI Tasks B”).

The figure below shows project script setting information.

The 'Script Edit' dialog box shows the following configuration for 'eHMI Tasks - A':

- Order: 5
- Script No.: 932 (with a 'View...' button)
- Script Name: eHMI Tasks - A (with 'Import...' and 'Edit Script...' buttons)
- Comment: eHMI Tasks - A (with a 'Syntax Check' button)
- Data Type: Signed BIN16 (dropdown menu)
- Trigger Type: Rise (with a 'Trigger Setting...' button)
- Trigger Device: GB63013
- Buttons: OK, Cancel

3.6 “eHMI Tasks B”

This script reads data from `eHMI.csv`. The data is used to populate the follow task maintenance information (accessible from the task maintenance screens).

- Multispindle Sub Tasks

Once complete, the script will trigger the next required script by setting GB63015 (See section: “3.7 eHMI Tasks C”).

The figure below shows project script setting information.

The 'Script Edit' dialog box shows the following configuration for 'eHMI Tasks - B':

- Order: 6
- Script No.: 933 (with a 'View...' button)
- Script Name: eHMI Tasks - B (with 'Import...' and 'Edit Script...' buttons)
- Comment: eHMI Tasks - B (with a 'Syntax Check' button)
- Data Type: Signed BIN16 (dropdown menu)
- Trigger Type: Rise (with a 'Trigger Setting...' button)
- Trigger Device: GB63014
- Buttons: OK, Cancel

3.7 “eHMI Tasks C”

This script reads data from `eHMI.csv`. The data is used to populate the follow task maintenance information (accessible from the task maintenance screens).

- Test Sub Tasks
- Barcode Sub Tasks
- Socket Sub Tasks

Once complete, the script will trigger the next required script by setting GB63016 (See section: “3.8 eHMI Tasks D”).

The figure below shows project script setting information.

The screenshot shows a 'Script Edit' dialog box with the following fields and buttons:

- Order: 7
- Script No.: 334 (with a 'View...' button)
- Script Name: eHMI Tasks - C (with 'Import...' and 'Edit Script...' buttons)
- Comment: eHMI Tasks - C (with a 'Syntax Check' button)
- Data Type: Signed BIN16 (dropdown menu)
- Trigger Type: Rise (with a 'Trigger Setting...' button)
- Trigger Device: GB63015
- Buttons: OK, Cancel

3.8 “eHMI Tasks D”

This script reads data from `eHMI.csv`. The data is used to populate the follow task maintenance information (accessible from the task maintenance screens).

- Camera/Vision Sub Tasks

Once complete, the script will trigger the next required script by setting GB63017 (See section: “3.9 eHMI Sensor Descriptions”).

The figure below shows project script setting information.

Script Edit

Order: 8

Script No.: 935

Script Name: eHMI Tasks - D

Comment: eHMI Tasks - D

Data Type: Signed BIN16

Trigger Type: Rise

Trigger Device: GB63016

3.9 “eHMI Sensor Descriptions”

This script reads data from `Sensor.csv`. The data is used to populate sensor descriptions on the sensor task maintenance screens. Once complete, the script will trigger the next required script by setting GB63018 (See section: “3.10 eHMI Model Descriptions”).

The figure below shows project script setting information.

Script Edit

Order: 9

Script No.: 960

Script Name: eHMI Sensor Descriptions

Comment: eHMI Sensor Descriptions

Data Type: Signed BIN16

Trigger Type: Rise

Trigger Device: GB63017

3.10 “eHMI Model Descriptions”

This script reads data from `Model.csv`. The data is used to populate model descriptions on the model lookup screen. This is the last script in the eHMI update sequence.

The figure below shows project script setting information.

Script Edit

Order: 10

Script No.: 961

Script Name: eHMI Model Descriptions

Comment: eHMI Model Descriptions

Data Type: Signed BIN16

Trigger Type: Rise

Trigger Device: GB63018

3.11 “eHMI Cycle Time”

This script finds 75% of the configured cycle time.

Script Edit

Order: 11

Script No.: 962

Script Name: eHMI Cycle Time Script

Comment: eHMI Cycle Time Script

Data Type: Signed BIN16

Trigger Type: Rise/Fall

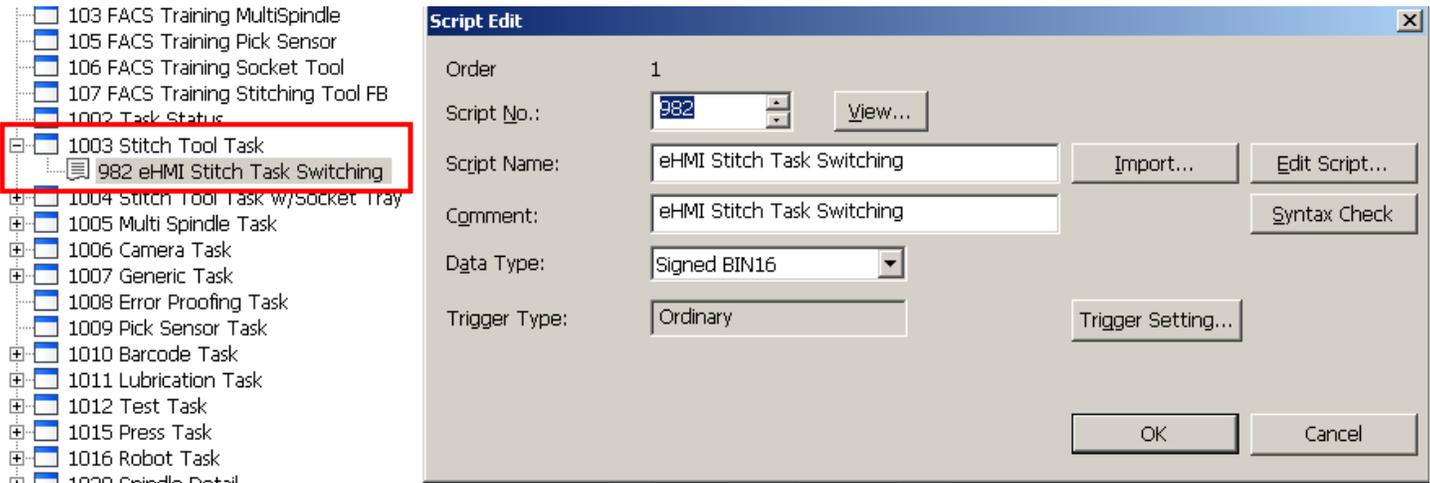
Trigger Device: eHMI_Display.Gen_FACSRcvComp

Screen Scripts

3.12 “eHMI Stitch Task Switching”

This script is called when an operator switches the main stitch task to display on the Stitch Task Maintenance Screen. It populates relevant data since only one task is displayed at a time.

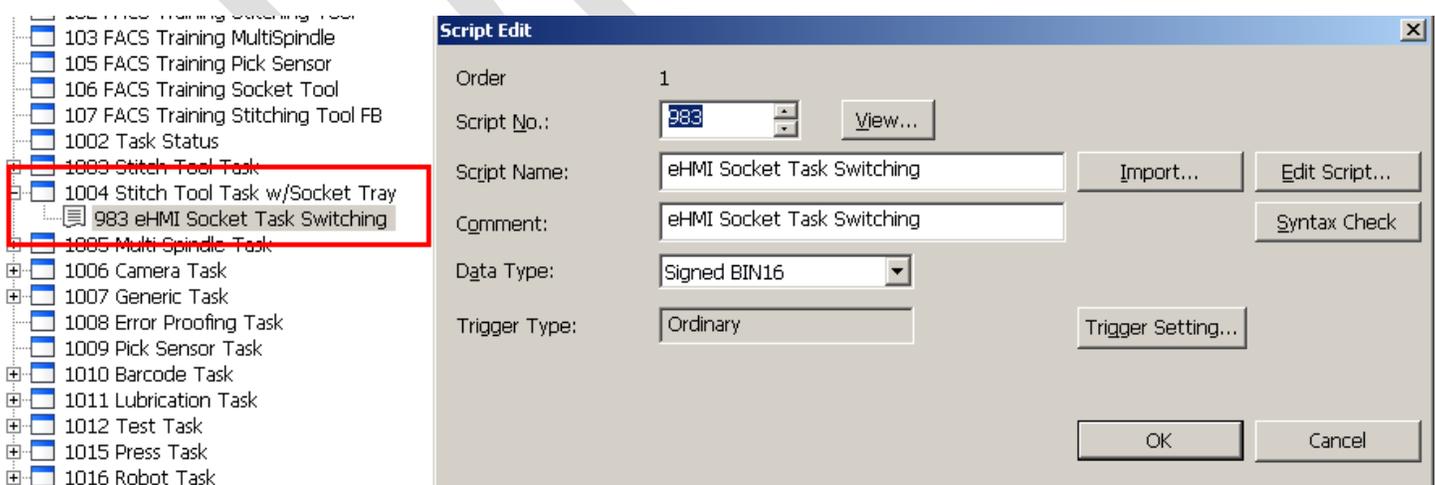
The figure below shows project script setting information.



3.13 “eHMI Socket Task Switching”

This script is called when an operator switches the main socket task to display on the Socket Task Maintenance Screen. It populates relevant data since only one task is displayed at a time.

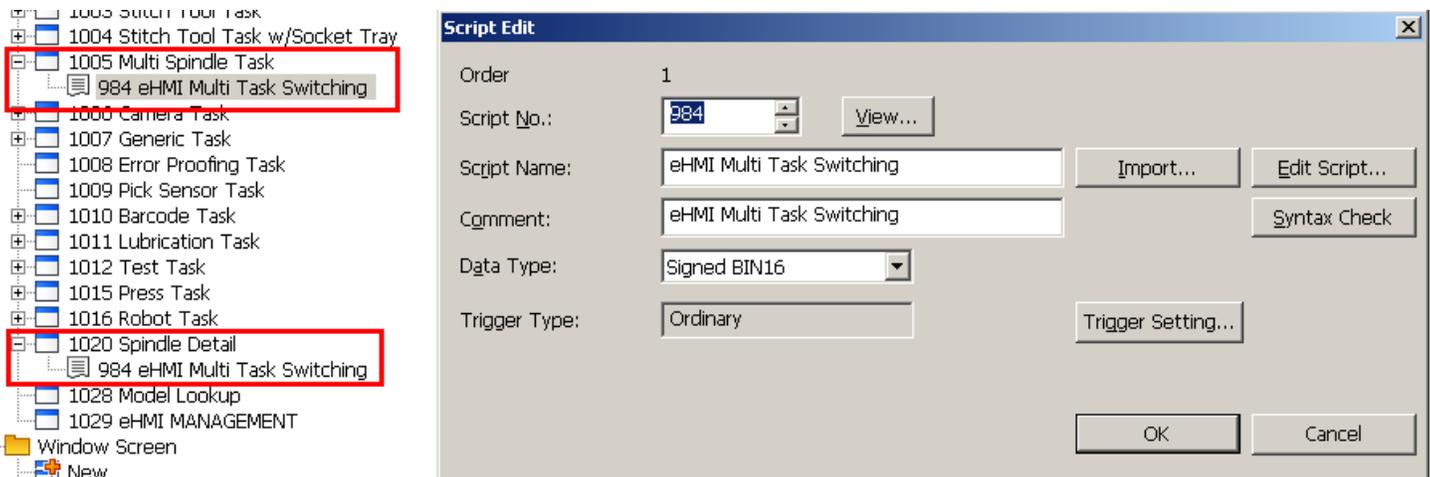
The figure below shows project script setting information.



3.14 “eHMI Multi Task Switching”

This script is called when an operator switches the main multispindle task to display on the Multispindle Task, and Spindle Detail Maintenance Screens. It populates relevant data since only one task is displayed at a time.

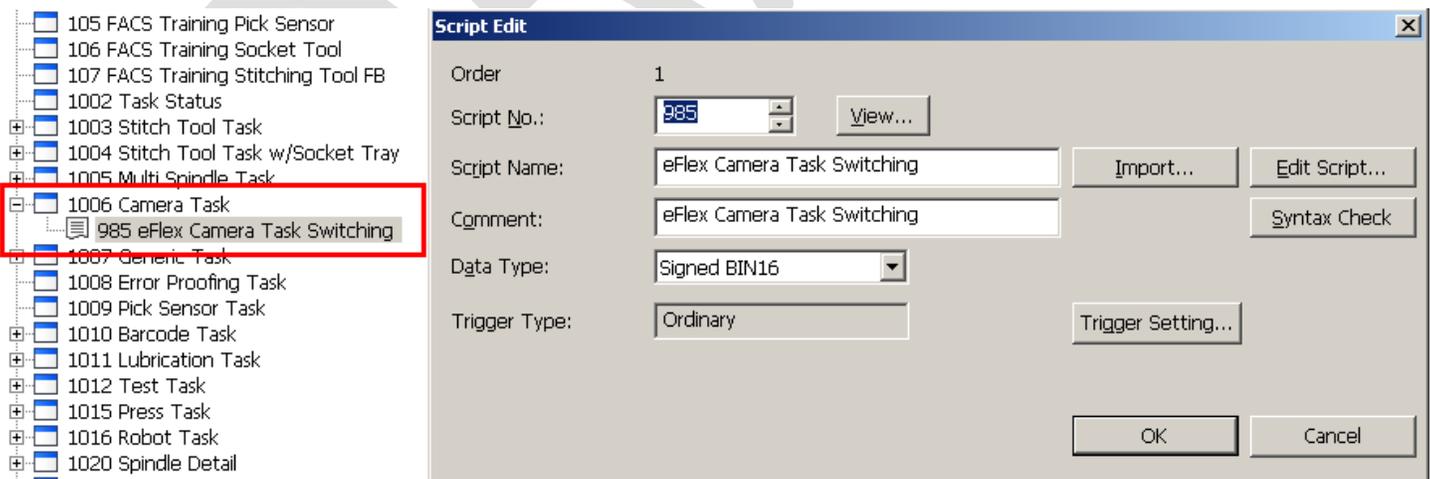
The figure below shows project script setting information.



3.15 “eHMI Camera Task Switching”

This script is called when an operator switches the main camera task to display on the Camera Task Maintenance Screen. It populates relevant data since only one task is displayed at a time.

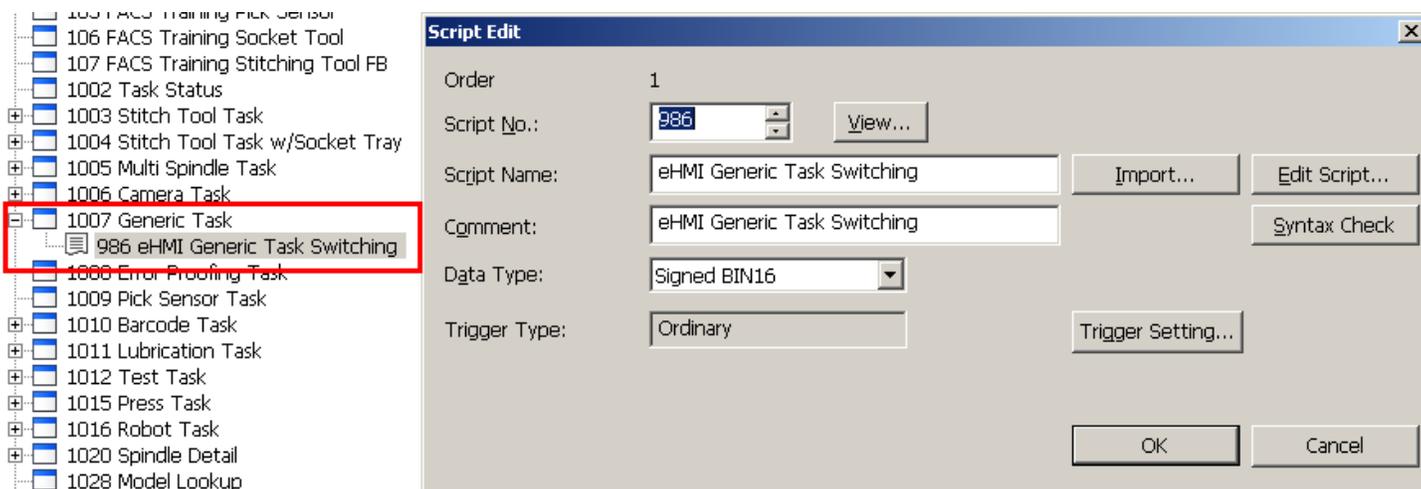
The figure below shows project script setting information.



3.16 “eHMI Generic Task Switching”

This script is called when an operator switches the main generic task to display on the Generic Task Maintenance Screen. It populates relevant data since only one task is displayed at a time.

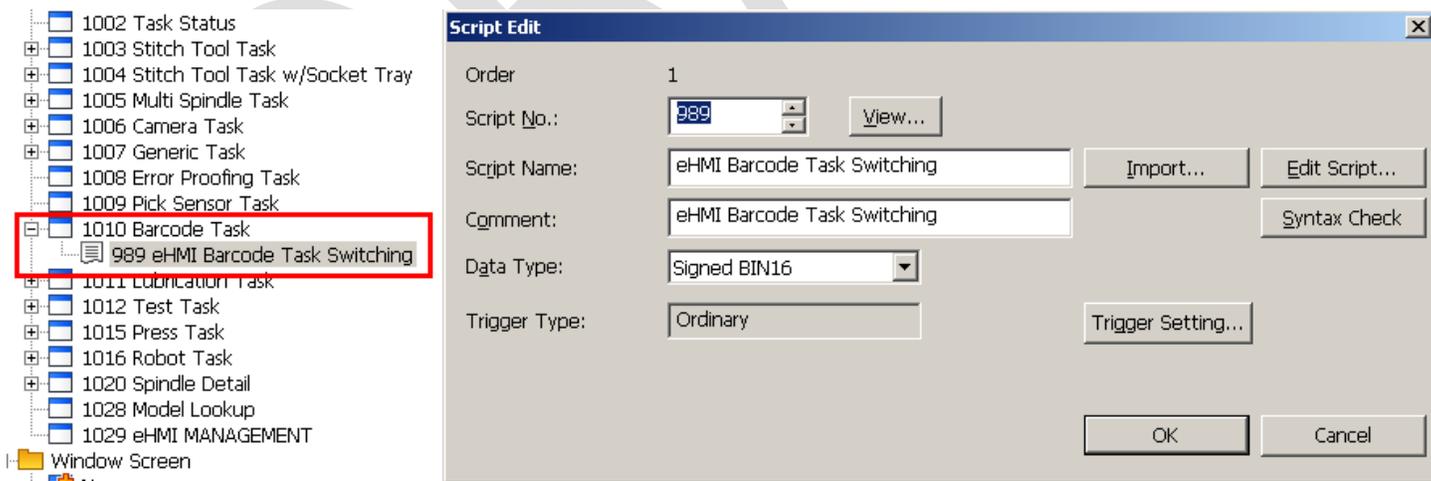
The figure below shows project script setting information.



3.17 “eHMI Barcode Task Switching”

This script is called when an operator switches the main barcode task to display on the Barcode Task Maintenance Screen. It populates relevant data since only one task is displayed at a time.

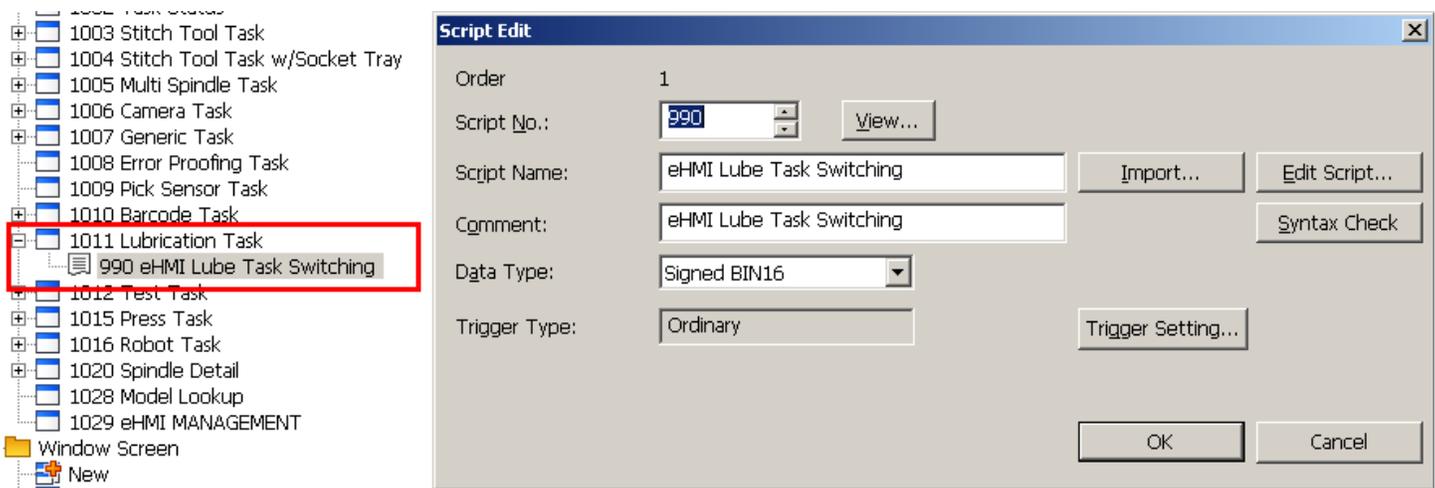
The figure below shows project script setting information.



3.18 “eHMI Lube Task Switching”

This script is called when an operator switches the main lube task to display on the Lube Task Maintenance Screen. It populates relevant data since only one task is displayed at a time.

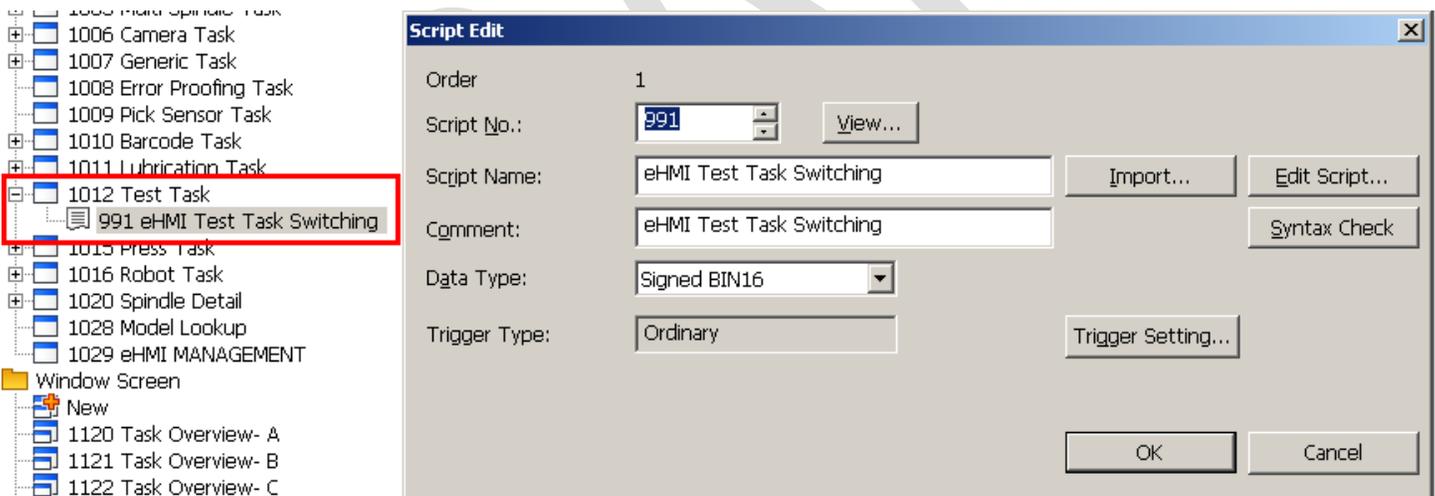
The figure below shows project script setting information.



3.19 “eHMI Test Task Switching”

This script is called when an operator switches the main test task to display on the Test Task Maintenance Screen. It populates relevant data since only one task is displayed at a time.

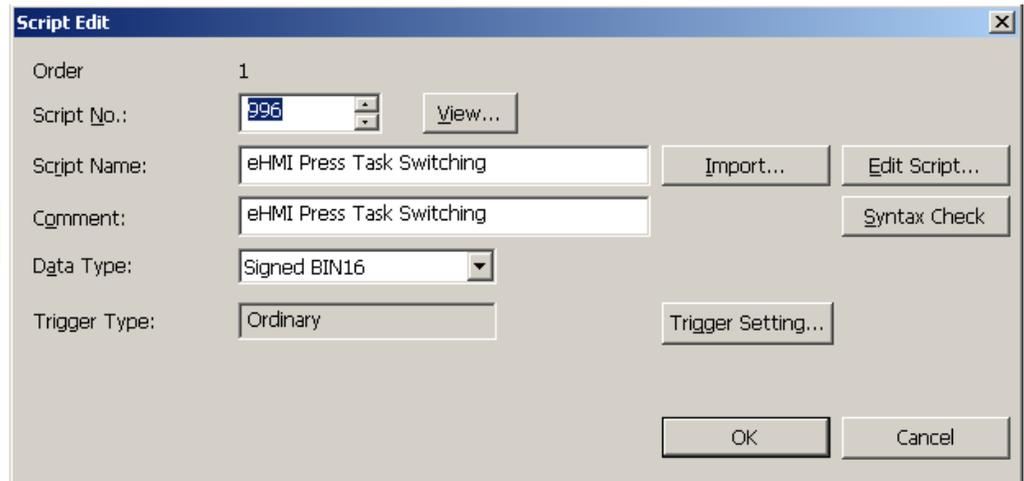
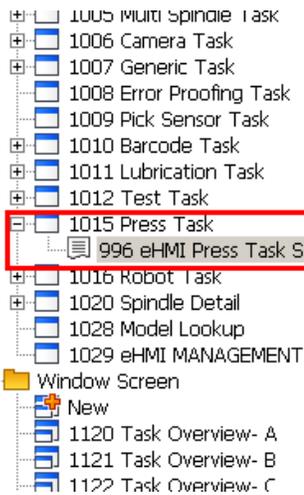
The figure below shows project script setting information.



3.20 “eHMI Press Task Switching”

This script is called when an operator switches the main press task to display on the Press Task Maintenance Screen. It populates relevant data since only one task is displayed at a time.

The figure below shows project script setting information.



3.21 “eHMI Robot Task Switching”

This script is called when an operator switches the main robot task to display on the Robot Task Maintenance Screen. It populates relevant data since only one task is displayed at a time.

The figure below shows project script setting information.

