

ORMEC MMI-8000 FAMILY

Quick Start Guide

Copyright © 2009

ORMEC Systems Corp.

All Rights Reserved

19 Linden Park

Rochester, NY 14625

<http://www.ormec.com>

February 19, 2009

Table of Contents

List of Figures	iii
Introduction	1
Overview	3
Communication Connections.....	3
HMI to SMLC or ORION using Ethernet	3
HMI to SMLC RS-232	4
HMI to ORION RS-485	4
Using an MMI-8000 with an SMLC.....	5
SMLC HMI interface	5
Adding Registers to the SMLC’s Virtual Modbus PLC.....	5
SMLC Project using Ethernet	6
SMLC Project using RS-232	8
SMLC Data Types Supported.....	9
INT	9
DINT	10
REAL	11
STRING	12
Using Bit objects.....	12
Using an MMI-8000 with an ORION.....	15
ORION HMI Interface	15
MAPPING Registers in the ORION’s MotionBASIC program	15
ORION Project using Ethernet.....	15
ORION Project using RS-232	17
ORION Data Types Supported.....	18
DIO@	18
Integer (%).....	19
Long Integer (&)	20
Single Precision (!).....	20
String (\$).....	21
Converting EB500 projects to EB8000	23

Other Differences between EasyBuilder 500 and EasyBuilder 8000 25

Considerations when replacing an existing MMI-320/640 with an MMI-8xxx..... 26

Appendix A 27

 SMLC to MMI-8xxx Serial Cable 27

 ORION to MMI-8xxx Serial Cable 28

List of Figures

Figure 1 - EasyBuilder 8000 Main Help Screen	1
Figure 2 - Table of MMI-8000 Model Numbers	3
Figure 3 - Table of accessories	3
Figure 4 - HMI to SMLC/ORION Ethernet connection	4
Figure 5 - HMI to SMLC RS-232 connection	4
Figure 6 - HMI to ORION RS-485 connection	4
Figure 7 - SMLC Plc Configuration, HMI Settings showing both protocols enabled.....	5
Figure 8 – SMLC, adding blocks of Modbus registers.....	5
Figure 9 – SMLC, Modbus register numbers	6
Figure 10 - SMLC register blocks in the HMI interface.....	6
Figure 11 - SMLC Ethernet, selecting HMI model	6
Figure 12 - SMLC Ethernet, create new device	7
Figure 13 - SMLC Ethernet, device properties	8
Figure 14 - SMLC Ethernet, setting IP address	8
Figure 15 - SMLC RS-232, device properties	9
Figure 16 – SMLC, data types supported in the HMI interface	9
Figure 17 - SMLC, accessing an INT	10
Figure 18 - SMLC, INT Numeric Format settings	10
Figure 19 – SMLC, accessing a DINT.....	11
Figure 20 - SMLC, DINT Numeric Format settings.....	11
Figure 21 - SMLC, accessing a REAL	11
Figure 22 - SMLC, REAL Numeric Format settings.....	12
Figure 23 - SMLC, accessing STRINGS.....	12
Figure 24 - SMLC, assigning HMI bits to SMLC bits	13
Figure 25 - ORION, register offset by MBX.....	15
Figure 26 - ORION Ethernet, selecting the HMI Model.....	16
Figure 27 - ORION Ethernet, create new device	16
Figure 28 - ORION Ethernet, device properties	17
Figure 29 - ORION RS-485, 4-wire, device properties	18
Figure 30 – ORION, MotionBASIC data types supported in the HMI interface	18
Figure 31 - ORION, accessing DIO@ points.....	19
Figure 32 - ORION, accessing integers	19
Figure 33 - ORION, integer Numeric Format settings	20
Figure 34 - ORION, accessing long integers	20
Figure 35 - ORION, long integer numeric format	20
Figure 36 - ORION, accessing single precision variables	21
Figure 37 - ORION, single precision numeric format.....	21
Figure 38 - ORION, accessing String variables.....	22
Figure 39 - Translate MT500 Project.....	23
Figure 40 - Translate settings.....	23
Figure 41 - Translate, Device Type and Address changes table	24
Figure 42 - Translate, restoring Window names.....	24
Figure 43 - Translate, changing the LW address for Keypads	25

Introduction

This manual offers a quick start guide to:

- Interfacing MMI-8xxx HMIs to ORMEC SMLC controllers
- Interfacing MMI-8xxx HMIs to ORMEC ORION controllers
- Converting EasyBuilder 500 projects to EasyBuilder 8000

For additional installation, hardware, system setup and programming details, refer to the EasyBuilder 8000 online documentation which is available from the Help menu within EasyBuilder 8000.

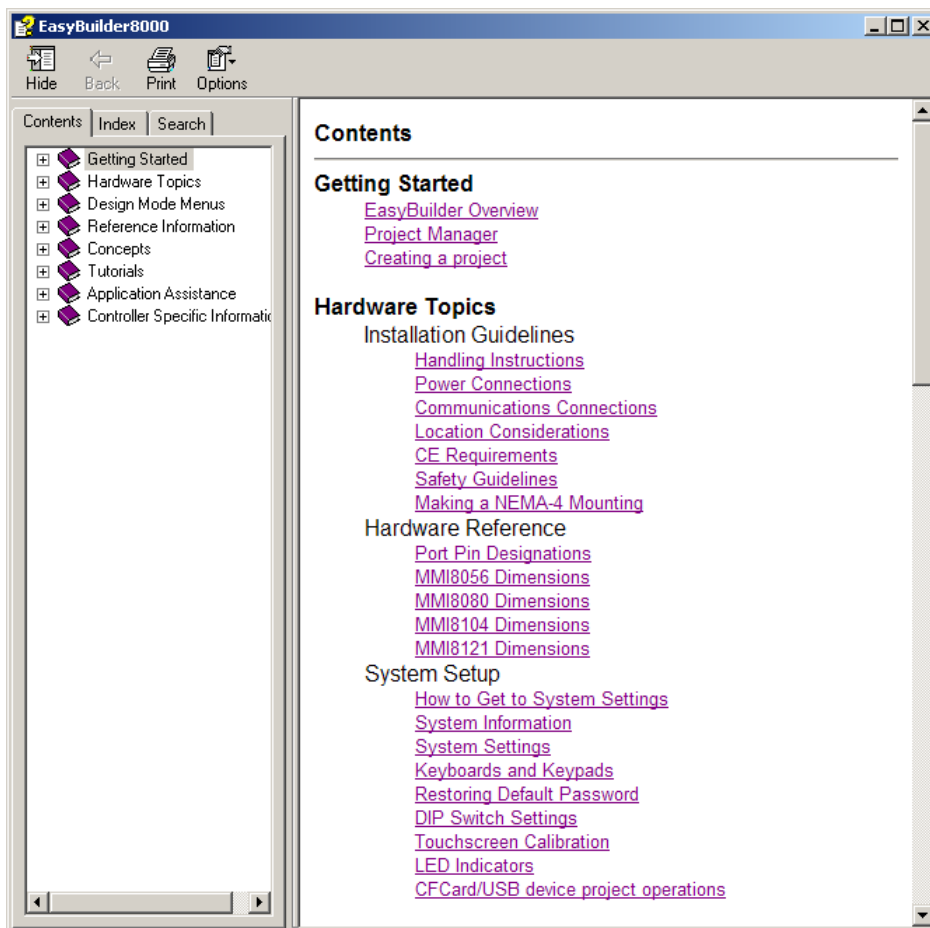


Figure 1 - EasyBuilder 8000 Main Help Screen

Overview

ORMEC EasyBuilder 8000 supports the MMI-8000 family. The available HMI models are listed below. Each model comes with an Ethernet crossover cable which is used for programming the HMI, and may also be used to interface the HMI to the SMLC or ORION. Each HMI also comes with the EBD-SDK8/C EasyBuilder 8000 Software Development Kit on a CDROM.

ORMEC EasyBuilder 500 is no longer available. EasyBuilder 500 versions 2.5 - 2.7.2 were used to program the obsolete MMI-320 and MMI-640 HMI models. Note that the serial programming cable used with EasyBuilder 500 and MMI-320/640 models is no longer required for programming the MMI-8000 models.

Model	Description
MMI-8056	5.6" TFT with Ethernet, 24VDC
MMI-8080	8.0" TFT with Ethernet, 24VDC
MMI-8104	10.4" TFT with Ethernet, 24VDC
MMI-8121	12.1" TFT with Ethernet, 24VDC

Figure 2 - Table of MMI-8000 Model Numbers

Serial connections between MMI-8000 models and either an SMLC or ORION use different serial cables than the MMI-320/640 models. The table below lists the part numbers of the new serial communication cables required for MMI-8000 models:

Part Number	Description
CBL-SMLC-MMI8-nn	SMLC to MMI-8xxx Serial Cable, nn is length in ft.
CBL-ORN-MMI8-nn	ORION to MMI-8xxx Serial Cable, nn is length in ft.

Figure 3 - Table of accessories

Communication Connections

All programming of the HMI is done via the Ethernet connection. The connection between the HMI and SMLC/ORION may either be serial or Ethernet. In general, Ethernet is the best choice as it provides the most flexibility.

HMI to SMLC or ORION using Ethernet

Since the development connection to the HMI must be Ethernet, if you are using the Ethernet connection to the SMLC/ORION you should use a hub or switch to allow you to connect all devices simultaneously. If you use the crossover cable to develop your HMI program you will have to disconnect the cable from the PC and connect it to the SMLC/ORION to test your program.

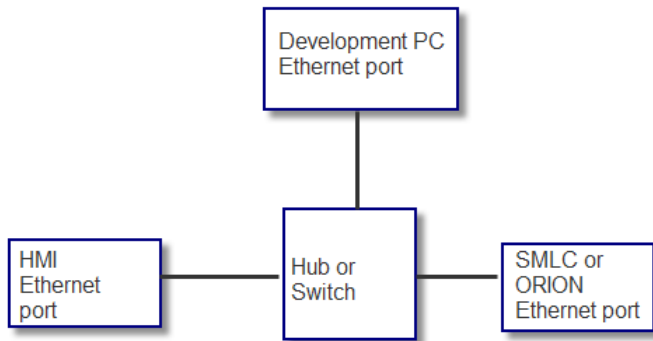


Figure 4 - HMI to SMLC/ORION Ethernet connection

HMI to SMLC RS-232

If the connection to the SMLC will be Modbus RTU via RS-232 then you can use the crossover cable between your PC and the HMI as the development connection. You may also use a hub/switch in this configuration.

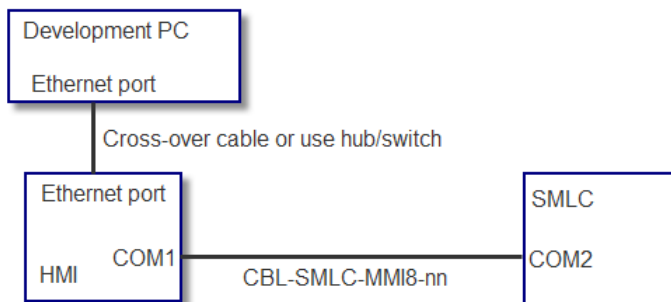


Figure 5 - HMI to SMLC RS-232 connection

HMI to ORION RS-485

If the connection to the ORION will be Modbus RTU via RS-485 4W then you can use the crossover cable between your PC and the HMI as the development connection. You may also use a hub/switch in this configuration.

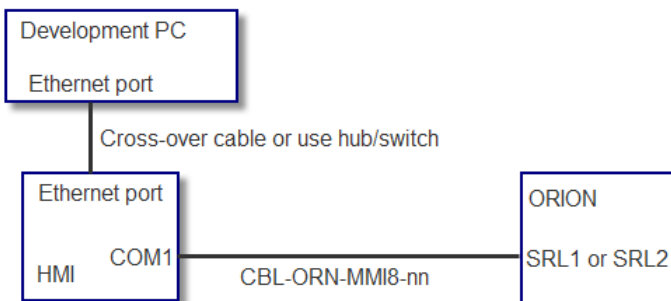


Figure 6 - HMI to ORION RS-485 connection

Using an MMI-8000 with an SMLC

SMLC HMI interface

The SMLC supports both Modbus/TCP (Ethernet) and Modbus/RTU (RS-232) as HMI interfaces. These protocols need to be enabled in the PLC Configuration in CoDeSys. You can enable either protocol individually or both simultaneously. Note that Modbus/RTU is always on the SMLC’s COM2 port. Modbus/TCP can be enabled on either of the SMLC’s Ethernet ports, but not both at the same time. Enter the IP address of the SMLC Ethernet port that you wish to use in the EthernetPort field.

Index	Name	Value	Default
1	StationAddress	1	1
2	Type	Both Modbus/RTU_Modbus/TCP	No HMI
3	EthernetPort	192.168.0.253	192.168.0.253
4	COM2_BaudRate	19200	19200
5	COM2_DataBits	8	8
6	COM2_StopBits	1	1
7	COM2_Parity	None	None

Figure 7 - SMLC Plc Configuration, HMI Settings showing both protocols enabled

Adding Registers to the SMLC’s Virtual Modbus PLC

Right click on the HMI/Modbus Interface on the PLC Configuration tab in CoDeSys and select a block of registers to add. The available data types for each block are shown below:

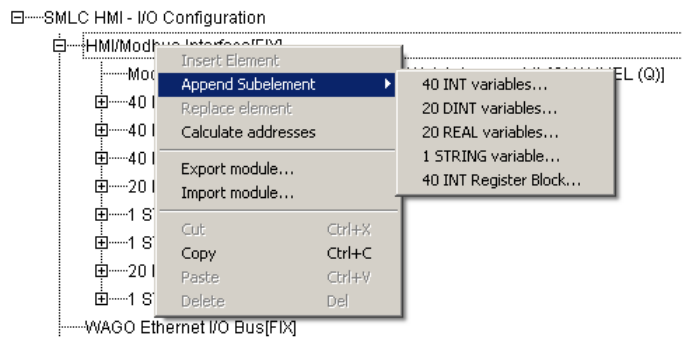


Figure 8 - SMLC, adding blocks of Modbus registers

Each time you add a block of registers, the register number is indicated by the physical address in the SMLC.

%QW1 = Register 1

%QW2 = Register 2

Etc.

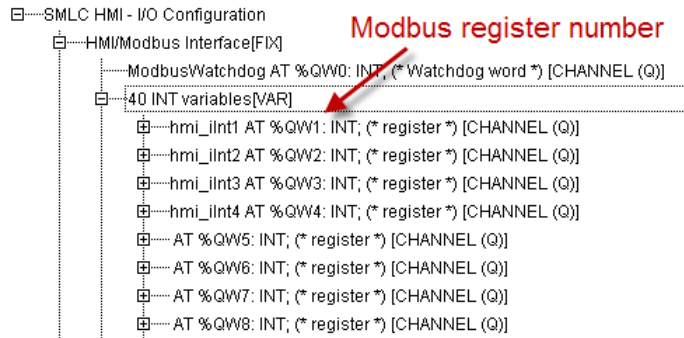


Figure 9 - SMLC, Modbus register numbers

The SMLC supports up to 2000 registers that are available to the Modbus interface.

For the purposes of this exercise, insert the following register blocks:

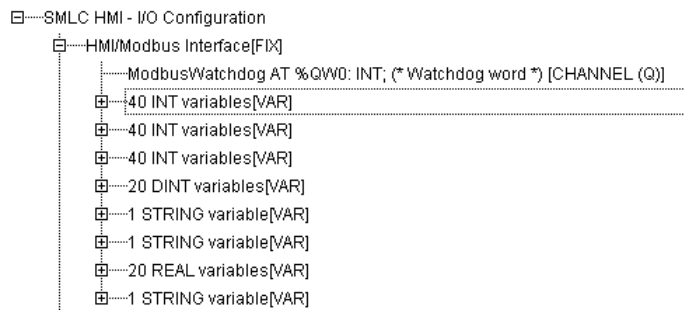


Figure 10 - SMLC register blocks in the HMI interface

SMLC Project using Ethernet

Create a new EasyBuilder 8000 project and select the HMI model

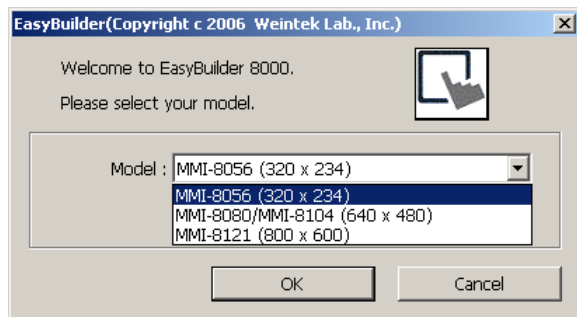


Figure 11 - SMLC Ethernet, selecting HMI model

The System Parameter dialog automatically appears. Click on the New button to create a new device that the HMI will communicate with.

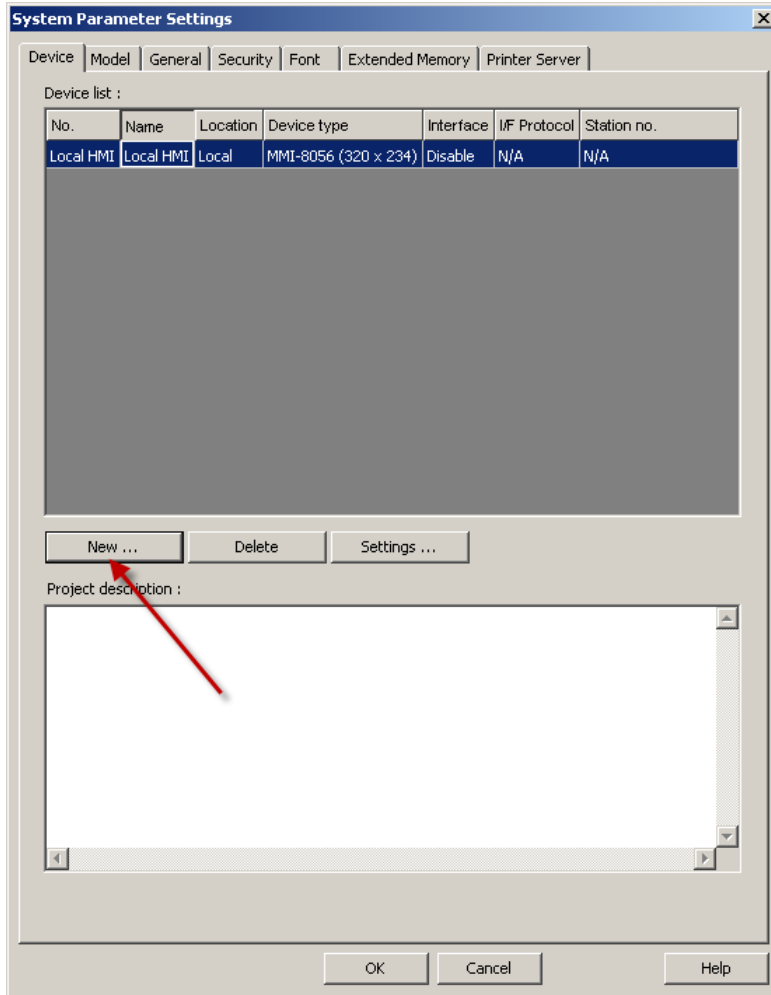


Figure 12 - SMLC Ethernet, create new device

First select MODBUS TCP/IP for the PLC type. Second, type in the name of this device. If you type in the name first, it will be reset when you pick the PLC type. Third, click on the Settings button to set the IP address of the SMLC.

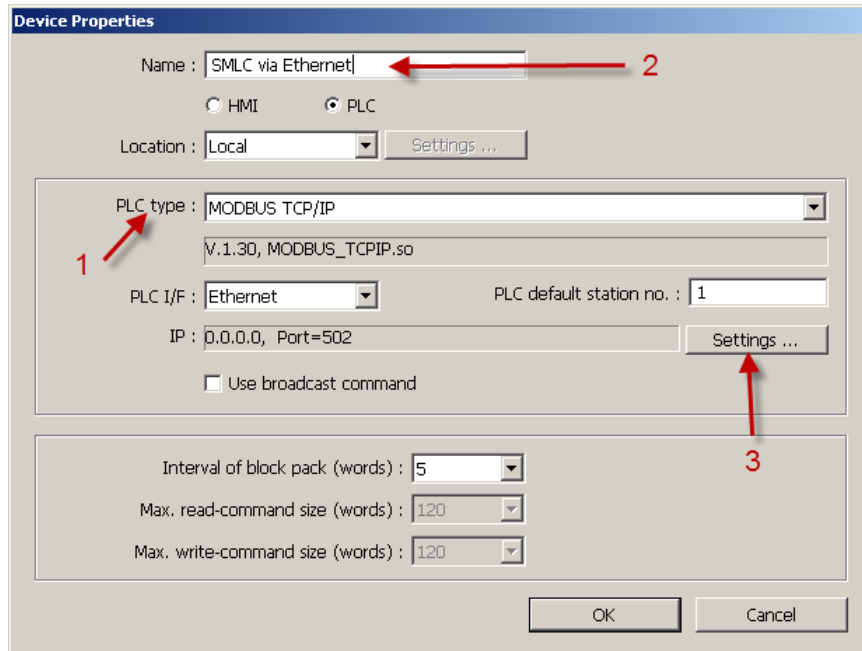


Figure 13 - SMLC Ethernet, device properties

Type in the IP address of the SMLC Ethernet port that has Modbus/TCP communications enabled. Leave the Port no. and other settings at the default values.

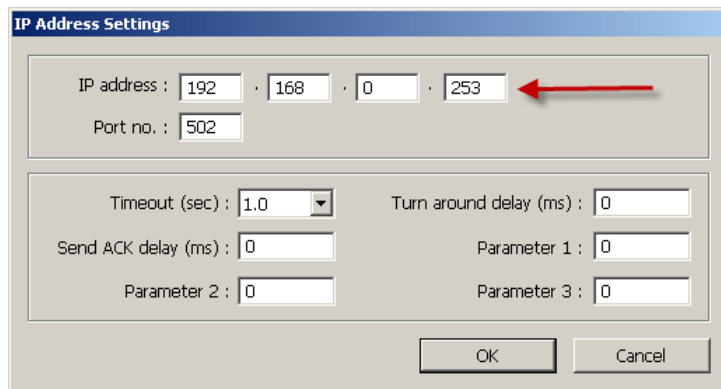


Figure 14 - SMLC Ethernet, setting IP address

SMLC Project using RS-232

Follow the same steps as configuring SMLC Project using Ethernet until you get to the Device Properties page. Here select MODBUS RTU, RS-232 and set the COM1 settings to 19200, N, 8, 1 as shown below.

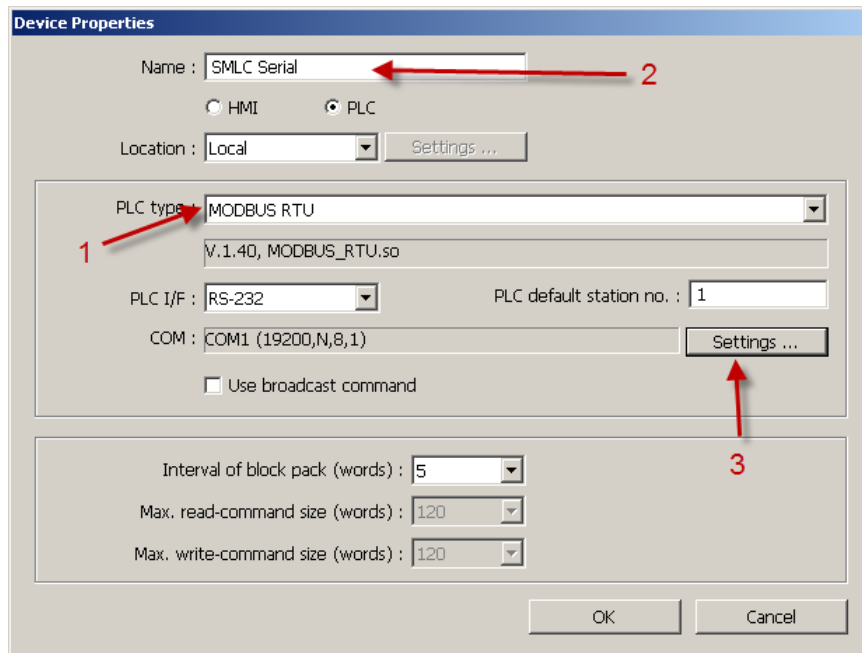


Figure 15 - SMLC RS-232, device properties

SMLC Data Types Supported

The Modbus interface in the SMLC supports the following IEC-61131-3 data types:

IEC Data Type	Number of Modbus registers	HMI Device Type
INT	1	4x
REAL	2	4x
DINT	2	4x
STRING	41 (HMI access is limited to 16)	4x

Figure 16 - SMLC, data types supported in the HMI interface

The following section demonstrates how to apply these data types in the HMI project in EasyBuilder.

INT

To demonstrate an integer we insert a Numeric Display Object in EasyBuilder. Select the PLC Name, which is the name we assigned to the SMLC Device, in this example: SMLC via Ethernet. For device type select 4x. For address type in 1, this corresponds to the variable at %QW1 in our SMLC’s HMI interface.

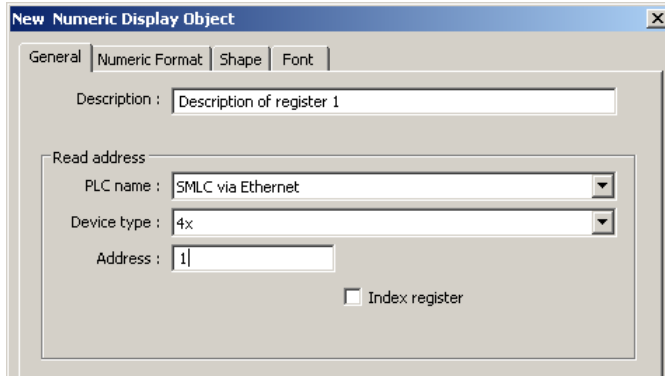


Figure 17 - SMLC, accessing an INT

Go to the Numeric Format tab and set the Data format to 16-bit Signed so that the data type in the HMI matches that in the SMLC's HMI interface. To access the full range of the SMLC's INT set the Limits to -32768 to +32767 (or any sub-range of this that your application requires).

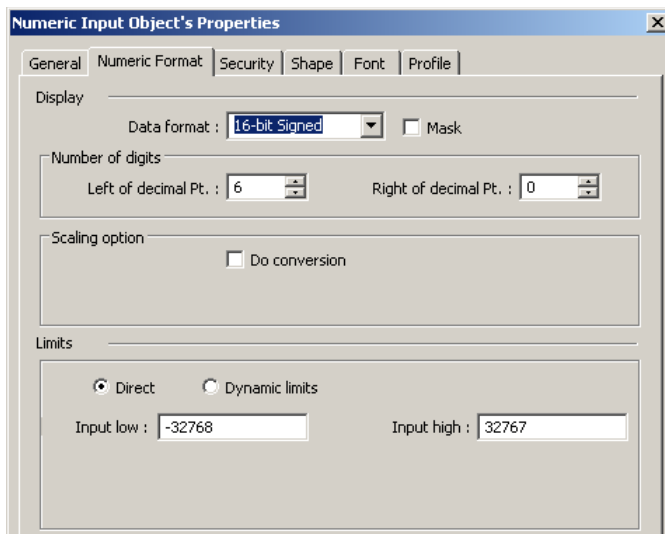


Figure 18 - SMLC, INT Numeric Format settings

DINT

In the SMLC's HMI interface DINT variables occupy two registers. To access a DINT value in the SMLC, select as Device Type of 4x and enter the register number of the first register in the DINT in the Address field.

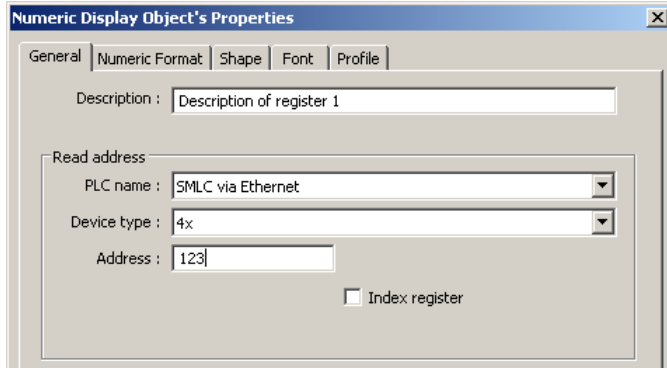


Figure 19 - SMLC, accessing a DINT

Go to the Numeric Format tab and select 32-bit Signed as the Data format.

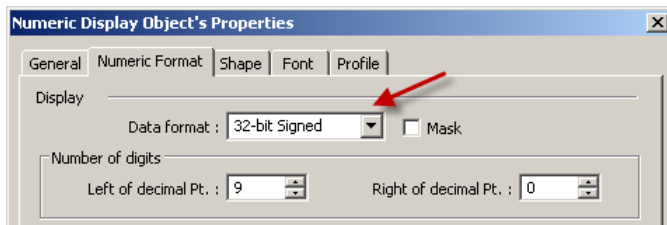


Figure 20 - SMLC, DINT Numeric Format settings

REAL

In the SMLC’s HMI interface REAL variables occupy two registers. To access a REAL value in the SMLC, select the Device Type of 4x and enter the register number of the first register in the REAL in the Address field

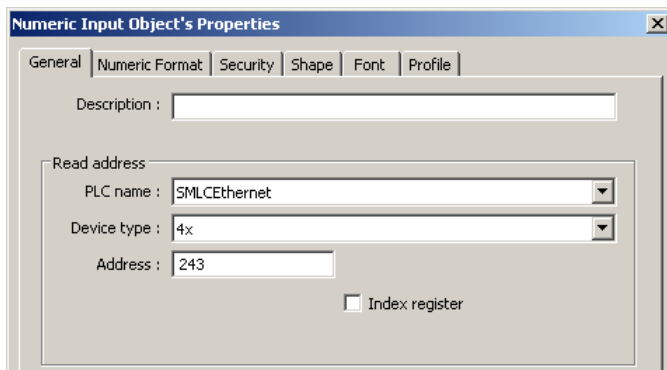


Figure 21 - SMLC, accessing a REAL

Go to the Numeric Format tab and select 32-bit Float as the Data format.

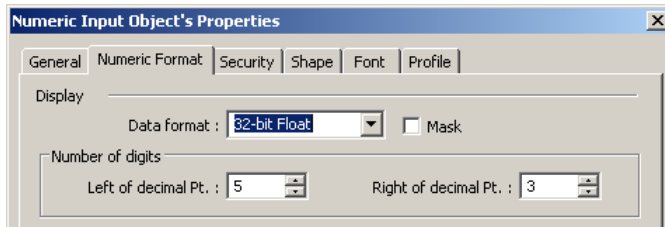


Figure 22 - SMLC, REAL Numeric Format settings

STRING

In the SMLC's HMI interface DINT variables occupy 41 registers. To access a STRING value in the SMLC, enter the register number of the first register. The HMI's display of strings is limited to 32 characters (16 words) so select the number of words (up to 16).

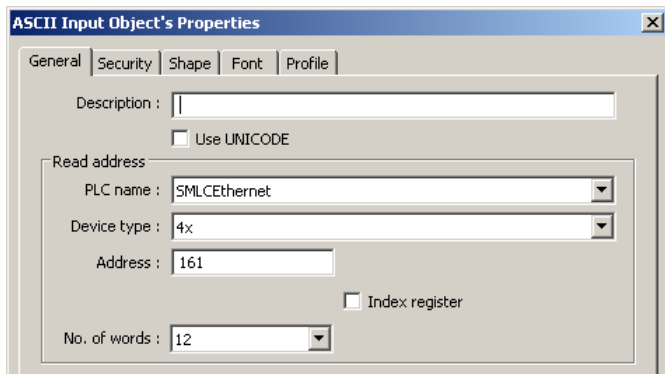


Figure 23 - SMLC, accessing STRINGS

Using Bit objects

Bit objects in the HMI can access individual bits in INT registers in the SMLC. When specifying bit addresses in EasyBuilder 8000, the Device Type 4x_bit is selected. The address of the bit is then entered in the form rbb where r is the register number and bb is the bit number 00 to 15. Some examples are:

115 – bit 15 of register 1

1000 – bit 00 of register 10

12307 – bit 07 of register 123

Do not use slashes, colons, underscores or any other characters to delimit the bit number from the register number.

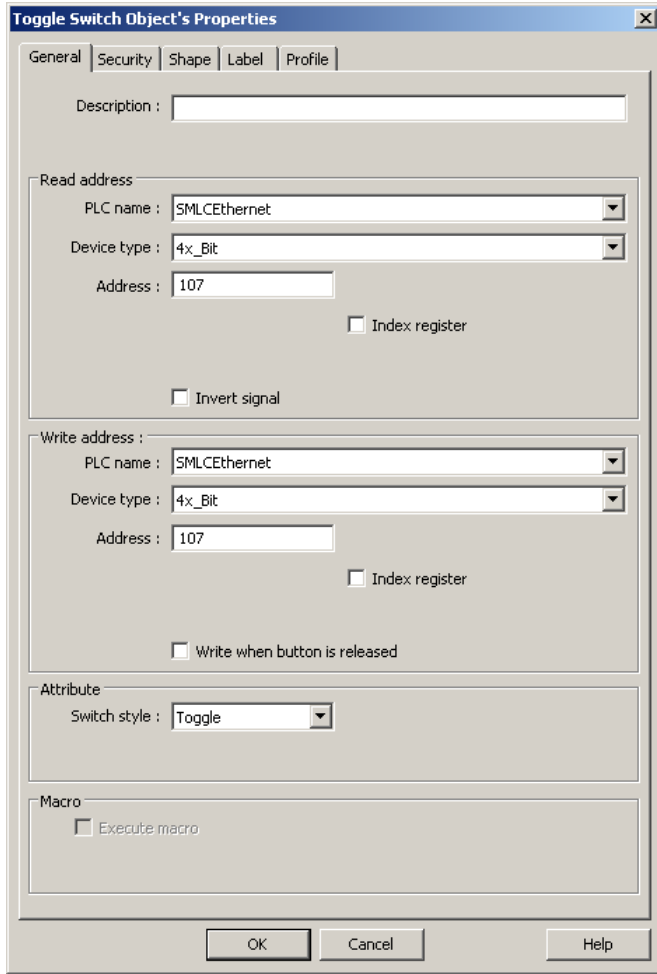


Figure 24 - SMLC, assigning HMI bits to SMLC bits

Using an MMI-8000 with an ORION

ORION HMI Interface

The ORION supports interfacing with an MMI-8xxx using MBX-MDB (Modbus), MBX-QE (Quantum Ethernet aka Modbus/TCP) or MBX-QP (QuickPanel). You must install one of these MBXs and have enough MotionCredits to run them. There is one difference between these three protocols. MBX-MDB uses a zero-based Modbus offset, MBX-QP and MBX-QE use a one-based Modbus offset. What this means is that if you specify register 1 in the MMI-8xxx it will access register 0 in MBX-MDB.

	MMI-8xxx	MBX-MDB	MBX-QP	MBX-QE
1 st Register	1	0	1	1

Figure 25 - ORION, register offset by MBX

MAPPING Registers in the ORION's MotionBASIC program

The ORION uses MBX-MAP to assign Modbus register numbers to MotionBASIC program variables. MBX-MAP must be installed. It does not require any MotionCredits. Once MBX-MAP has been installed you can create the mappings in your MotionBASIC program.

```
map erase
erase reg :dim reg(120)
for i=0 to 120
  map i to reg(i)
  reg(i)=i
next i
map 121..122 to long1&
map 123..124 to long2&
map 161..201 to string1$
map 202..242 to string2$
map 283..324 to string3$
map 243..244 to float1!
map 245..246 to float2!
map 247..248 to float3!
```

ORION Project using Ethernet

Create a new EasyBuilder 8000 project and select the HMI model

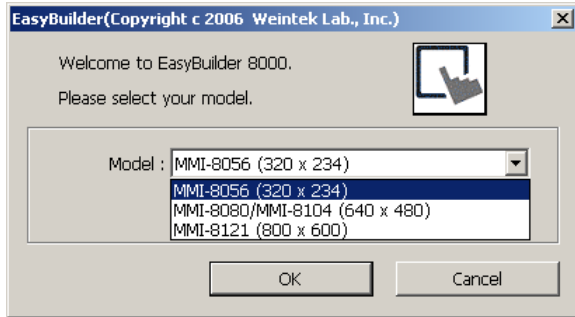


Figure 26 - ORION Ethernet, selecting the HMI Model

The System Parameter dialog automatically appears. Click on the New button to create a new device that the HMI will communicate with.

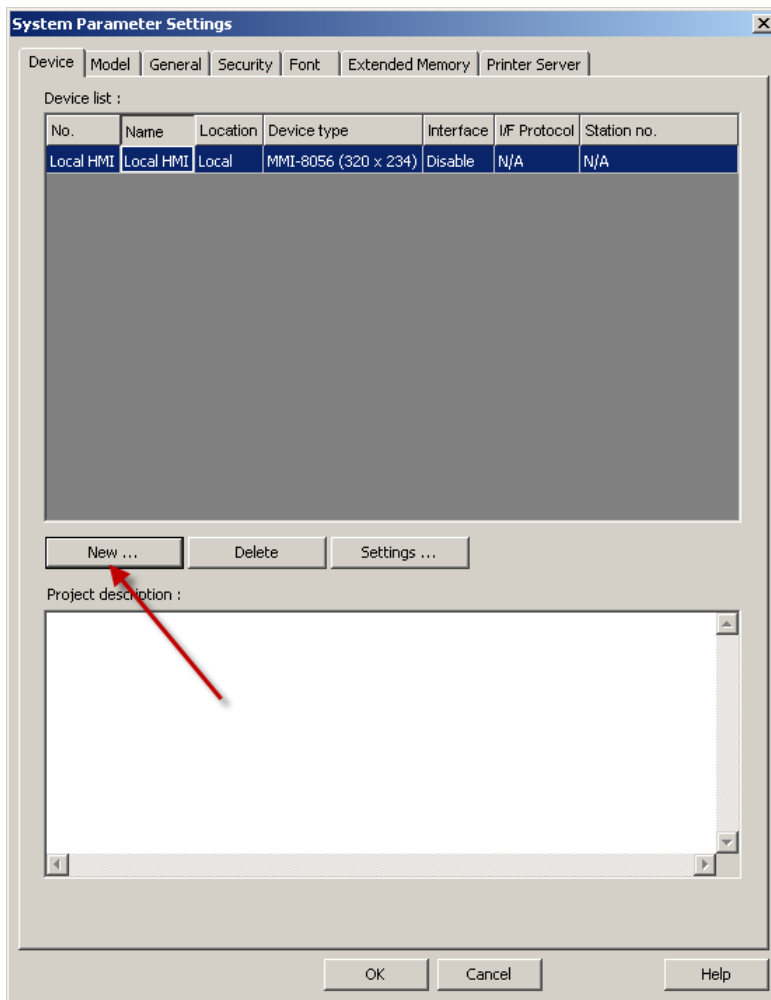


Figure 27 - ORION Ethernet, create new device

First select MODBUS TCP/IP for the PLC type. Second, type in the name of this device. If you type in the name first, it will be reset when you pick the PLC type. Third, click on the Settings button to set the IP

address of the ORION. Type in the IP address of the ORION Ethernet port. Leave the Port no. and other settings at the default values.

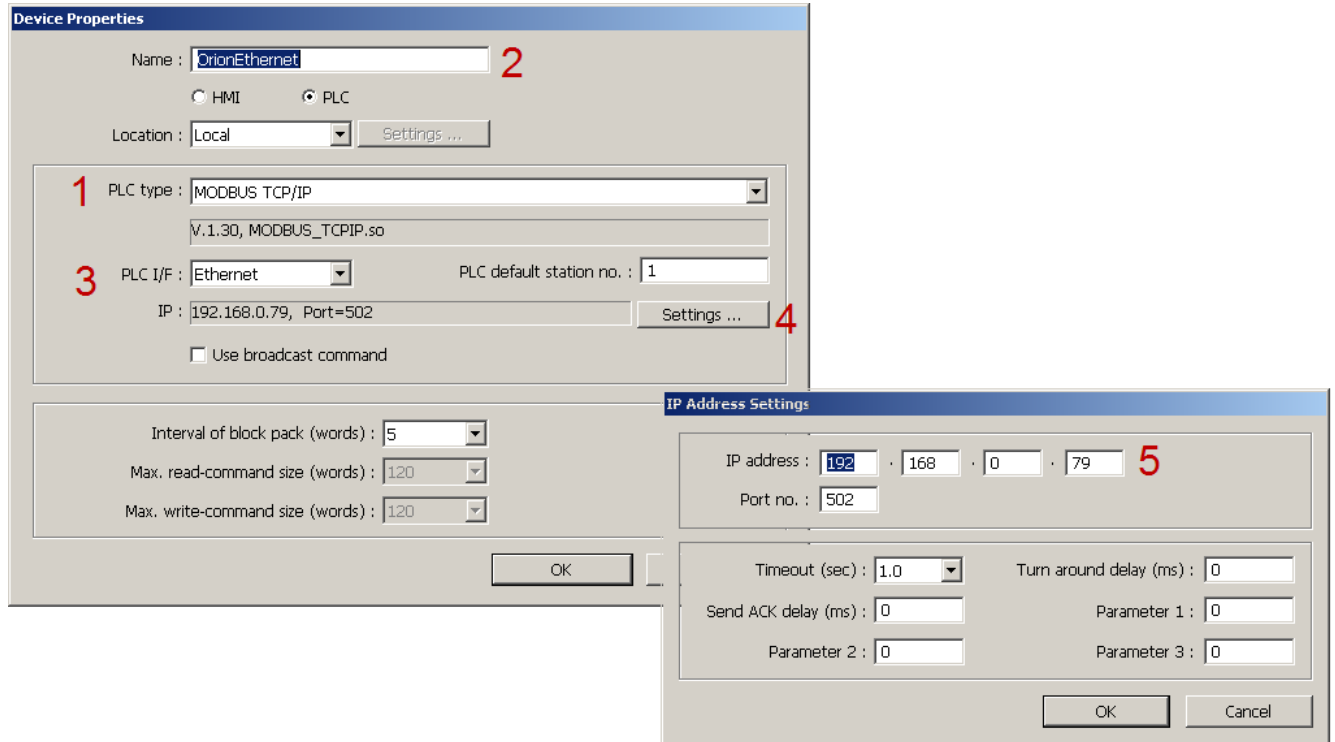


Figure 28 - ORION Ethernet, device properties

ORION Project using RS-232

Follow the same steps as configuring SMLC Project using Ethernet until you get to the Device Properties page. Here select MODBUS RTU, RS-485 4W and set the COM1 settings to 192000,N,8,1 as shown below.

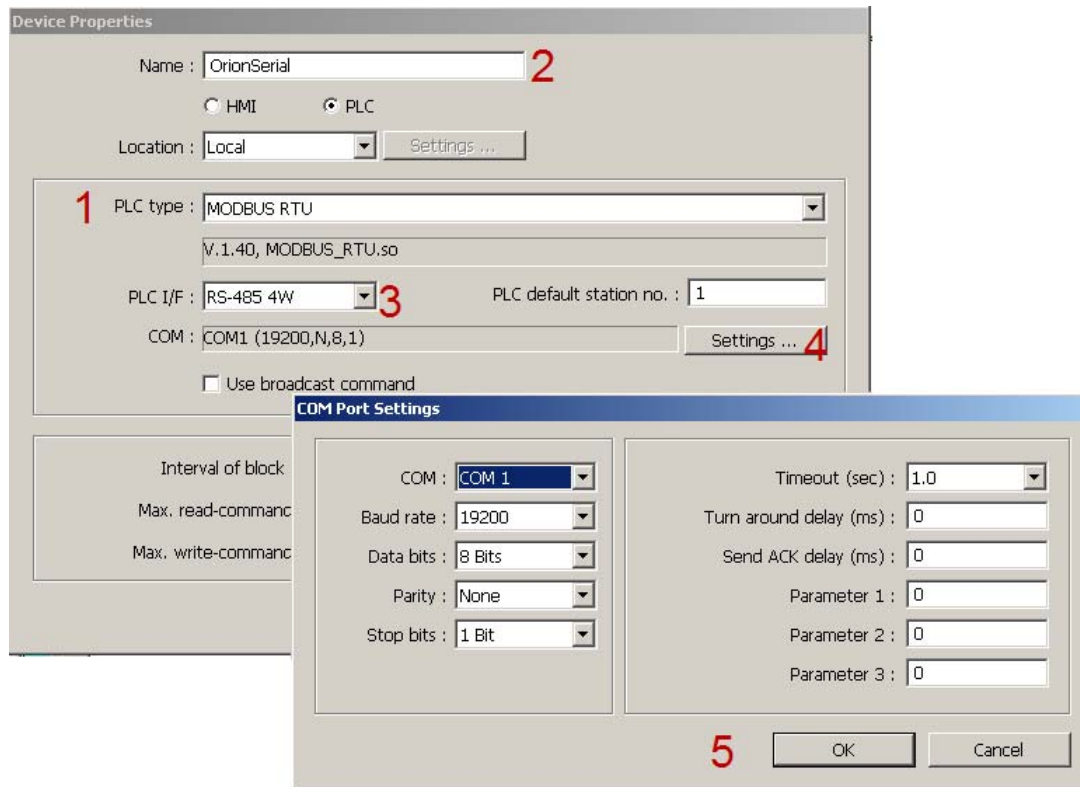


Figure 29 - ORION RS-485, 4-wire, device properties

ORION Data Types Supported

The Modbus interface (used by either MBX-MDB or MBX-QP) in the ORION supports the following data types:

MotionBASIC Data Type	Number of Modbus registers	HMI Device Type
DIO@	0	0x or 1x
Integer%	1	4x
LongInteger&	2	5x
Single!	2	5x
String\$	16 (limited by the HMI)	4x

Figure 30 - ORION, MotionBASIC data types supported in the HMI interface

The following section demonstrates how to apply these data types in the HMI project in EasyBuilder.

DIO@

To demonstrate accessing ORION DIO@ variables we insert a Toggle Switch in EasyBuilder. Select the PLC Name, which is the name we assigned to the ORION Device, in this example: ORION Serial. For device type select 0x, this is a Modbus Output address. For address type in 16, this corresponds to DIO@(16) in the ORION. This assumes that the IO.MODE@(16)='O'.

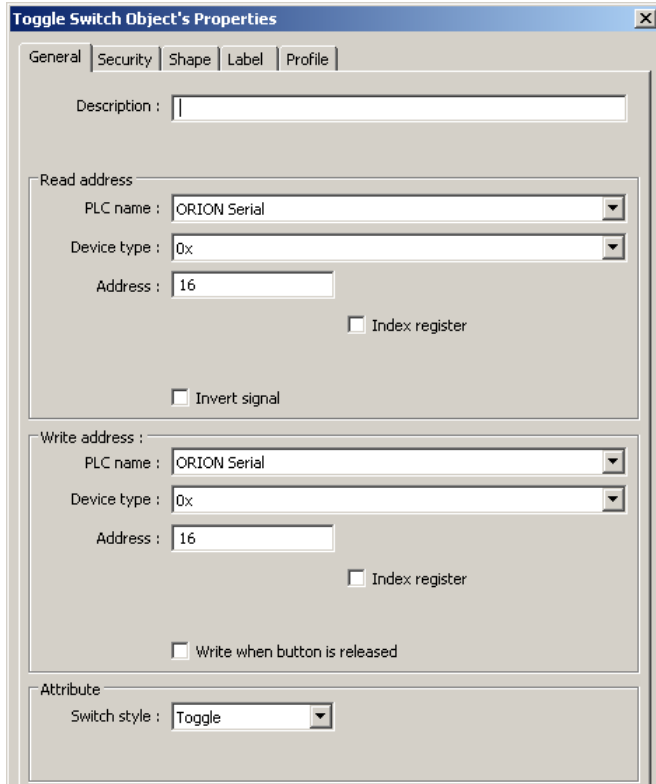


Figure 31 - ORION, accessing DIO@ points

To access an ORION DIO@ point that is an input use Device type: 1x. 1x corresponds to a Modbus Input address. If the ORION has Extended I/O enabled you can access the 1st 40 DIO@ points on the ORION in this manner.

Integer (%)

To demonstrate accessing an integer we insert a Numeric Display Object in EasyBuilder. Select the PLC Name, which is the name we assigned to the ORION device, in this example: ORION Serial. For device type select 4x. For address type in 1, this corresponds to the variable mapped to register 1 in the ORION.

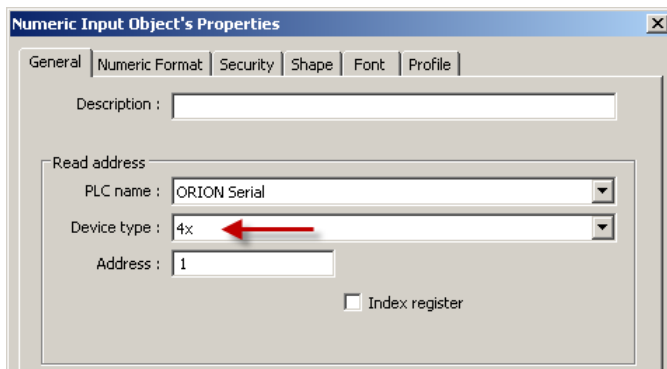


Figure 32 - ORION, accessing integers

Go to the Numeric Format tab and set the Data format to 16-bit Signed so that the data type in the HMI matches that of a MotionBASIC integer. To access the full range of the integer set the Limits to -32768 to +32767 (or any sub-range of this that your application requires).

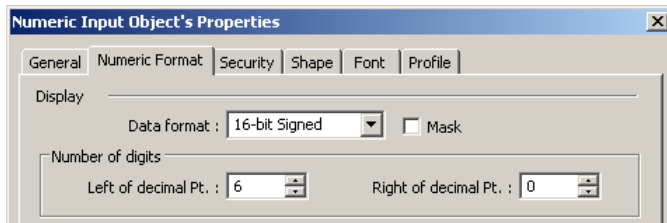


Figure 33 - ORION, integer Numeric Format settings

Long Integer (&)

In the ORION we MAP two consecutive registers to a long integer. To access a long integer from the HMI, select a Device Type of 5x and enter the register number of the first register in the Address field. The Device Type of 5x swaps the words within the long integer before displaying them on the HMI. Alternatively you could reverse the mapping of the individual words in the MotionBASIC program and then use a Device Type of 4x.

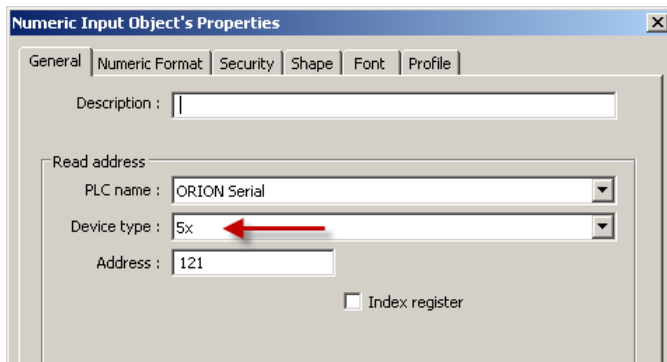


Figure 34 - ORION, accessing long integers

Go to the Numeric Format tab and set the Data format to 32-bit Signed so that the data type in the HMI matches that of a MotionBASIC long integer.

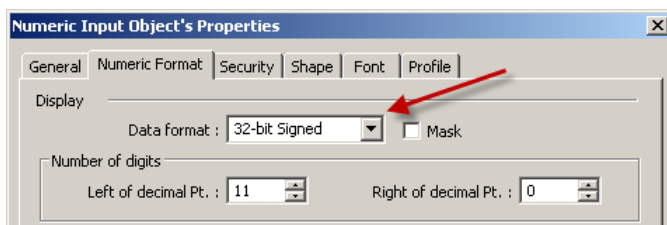


Figure 35 - ORION, long integer numeric format

Single Precision (!)

In the ORION we MAP two consecutive registers to a single precision floating point value. To access a single precision mapped variable from the HMI, select a Device Type of 5x and enter the register number

of the first register in the Address field. The Device Type of 5x swaps the words within the single precision value before displaying them on the HMI. Alternatively you could reverse the mapping of the individual words in the MotionBASIC program and then use a Device Type of 4x.

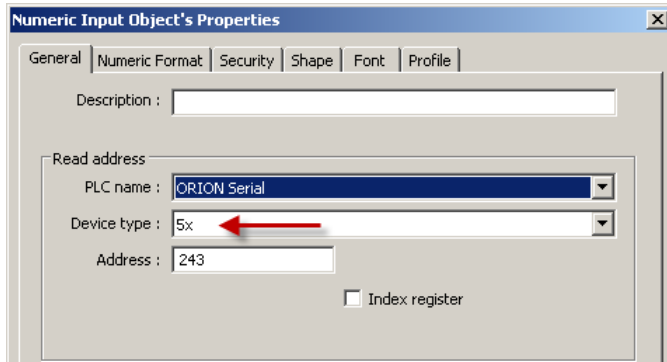


Figure 36 - ORION, accessing single precision variables

Go to the Numeric Format tab and set the Data format to 32-bit Float so that the data type in the HMI matches that of a MotionBASIC single precision variable.

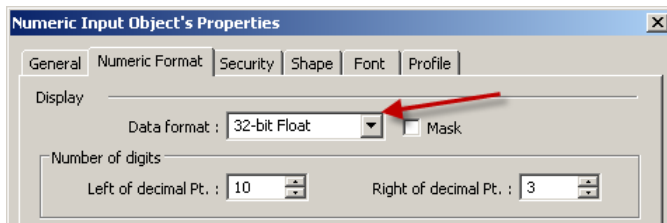


Figure 37 - ORION, single precision numeric format

String (\$)

In MotionBASIC you can map to the entire length of a string variable (up to 255 characters). However, the HMI is limited to displaying 32 character strings. Each word contains two characters so limit the number of words to 16 or less. Use Device type 4x and the Address should be the first mapped register of the string. Check the Reverse high/low byte checkbox to properly display the string.

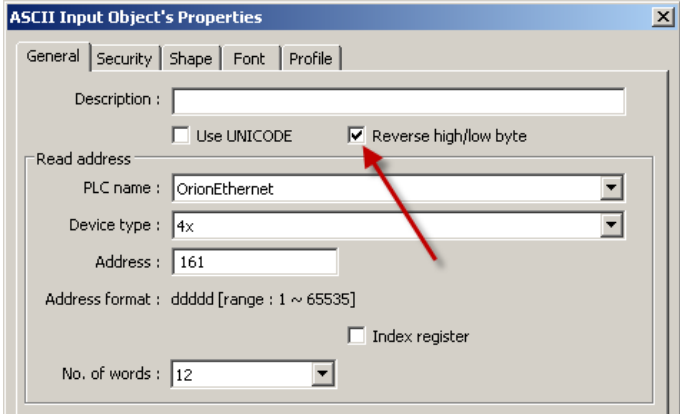


Figure 38 - ORION, accessing String variables

Converting EB500 projects to EB8000

Steps for converting an EB500 project into an EB8000 project

1. Compile the existing project with EasyBuilder 500, which will generate an .EOB file with the same name as the project.
2. Open EasyBulder 8000. Navigate to Tools and select Translate MT500 Project.

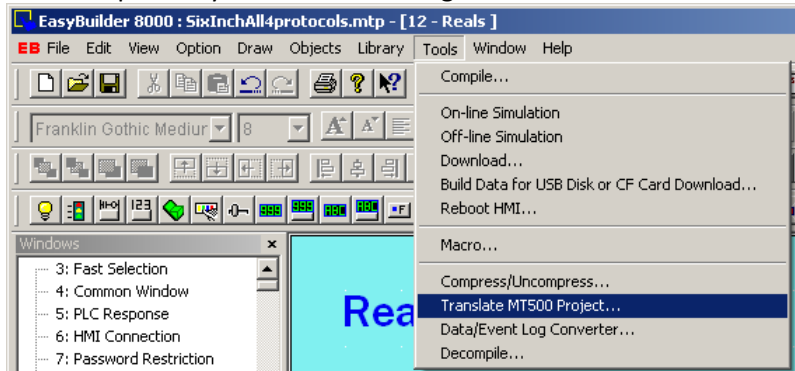


Figure 39 - Translate MT500 Project

3. Browse to and select the .EOB files generated in step 1 above.
4. Select the location to save the translated project
5. Change the default font to: Franklin Gothic Medium
6. Leave Delete "Direct Window" objects checkbox unchecked

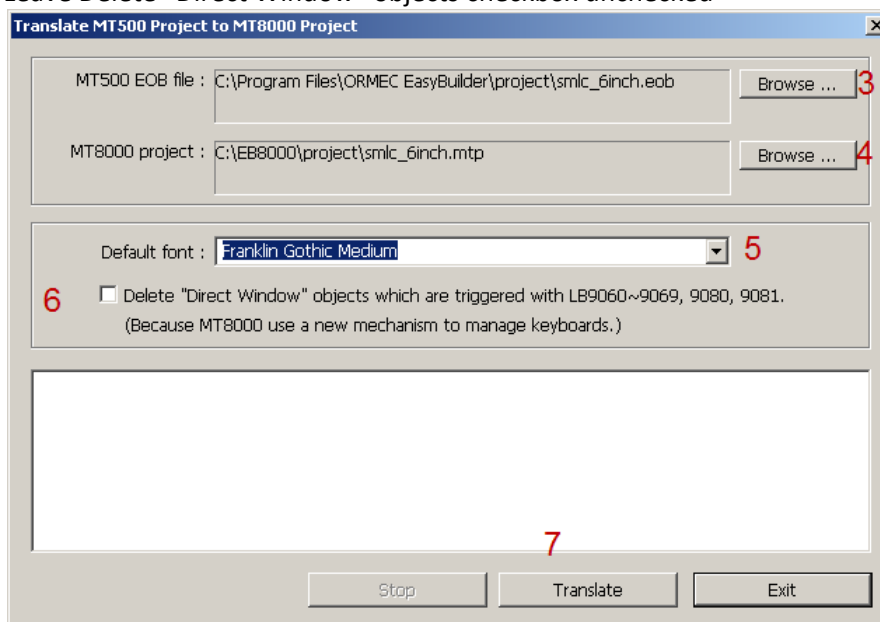


Figure 40 - Translate settings

7. Click the Translate button. Press Exit when the translation is done.
8. Open the newly created .MTP file in EasyBuilder 8000
9. Open the System Parameter Settings screen (from the Edit menu). If the old communication protocol was either ORION_Modbus or SMLC_Modbus, change it to Modbus RTU. If the old communication protocol was Modbus TCP/IP you can leave it unchanged. **IMPORTANT NOTE:**

when changing from ORION_Modbus or SMLC_Modbus to Modbus RTU some of the Device Types and Addresses may now need to be updated! The following table shows the mapping of Device Type and Address format between the protocols.

	EasyBuilder 500		EasyBuilder 8000
Device Type	ORION_Modbus	SMLC_Modbus	Modbus RTU
Register	Hold_4x	Hold_4x	4x
Bit object	Reg4x_bb	Reg4x_bb	4x_Bit
Address Format ¹			
Register	R	R	R
Bit object	R_bb or R/bb	R_bb or R/bb	Rbb

¹Where R is the Register number and bb is the bit number

Figure 41 - Translate, Device Type and Address changes table

10. Recreate the window names used in your original project. The translation process renames all Windows to Window_nn where nn is the window number.

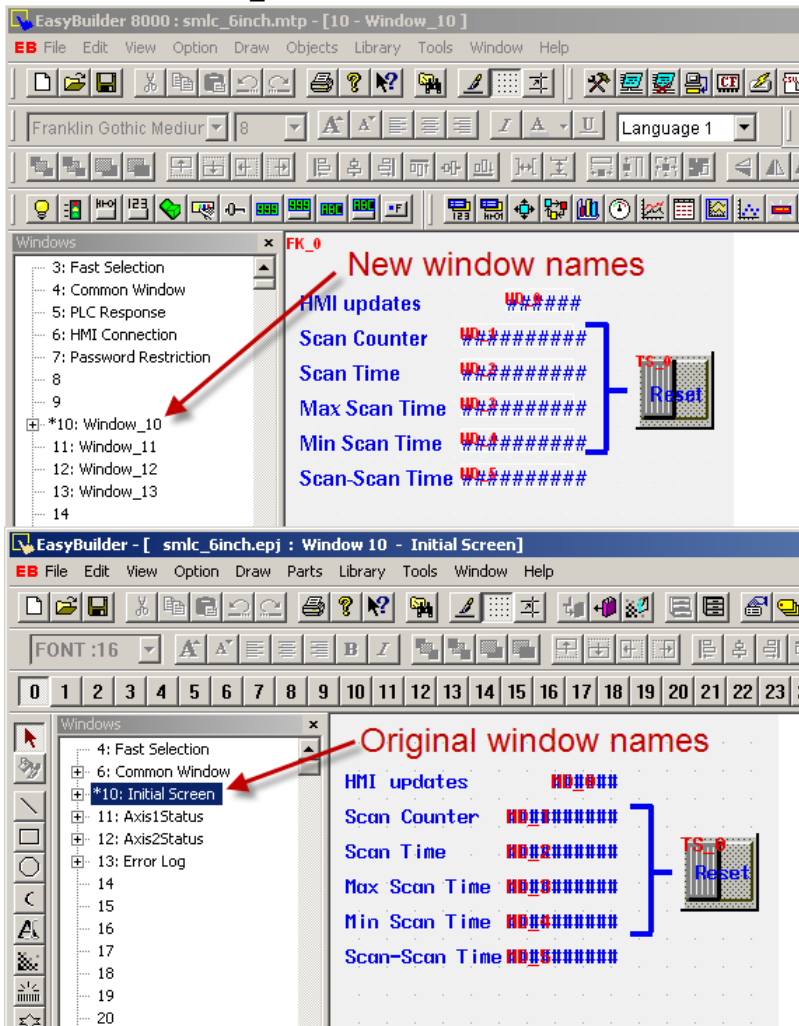


Figure 42 - Translate, restoring Window names

- Open the screen(s) with the keypads; locate and display the settings box for the ASCII Display Object's properties. Change the address from 9071 to 9150. Do this for all keypads in the project.

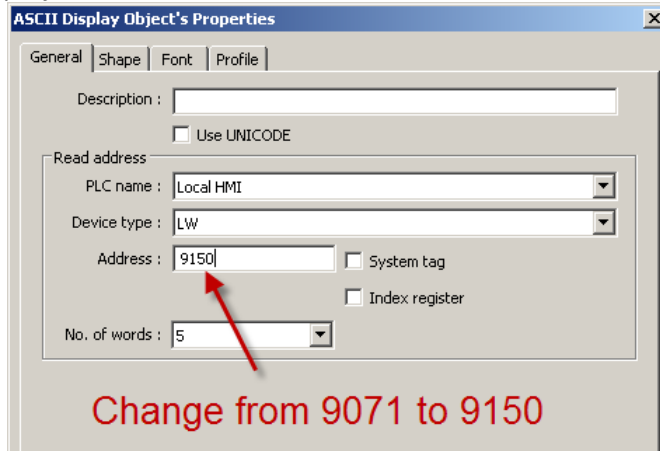


Figure 43 - Translate, changing the LW address for Keypads

- Open all windows and adjust the text and button placements as required.
- The **PLC no response** pop-up window (window #5 in EasyBuilder 8000 projects) may need to be resized and/or repositioned to show up properly on the selected HMI model.
- Compile the project and correct any errors, if they exist.
- Determine the IP address of the new terminal by pressing the icon in the lower right hand corner of the screen or drag the cursor to the lower right hand corner of the display if it is not present to pop up the built in utility display. If necessary, an external USB mouse can be connected and used. When it is displayed, press the blue monitor icon to view the System Information. Press the Wrench icon and enter the System Setting password, which is "111111" by default, to change the IP address.

Other Differences between EasyBuilder 500 and EasyBuilder 8000

- Screen project download is through Ethernet communications only. Previously it was done with a special serial cable and required a computer with an available serial port or a USB to Serial adapter.
- Online simulation no longer uses the HMI. The simulation's communications now occur between the PC and the PLC via Ethernet (or via the PC's COM1 if a serial protocol is selected). This has the advantage of allowing you to develop and test your screens without having the HMI. Note that the simulation connection times out after 20 minutes and must be restarted.
- Your computer should not be configured to use oversized fonts. The dots per inch (DPI) setting should be normal (96 dpi) not large (120 dpi). This will result in the screen layout to appear different on the computer screen than it does on the touch screen.
- If you develop your screens with a zoom level of 200% selected, center justified text may appear to be right justified, but will display properly in the touch screen.
- If a project is uploaded from a terminal, the terminal needs to be rebooted or power cycled after the upload is completed.

- When a project is uploaded and decompiled, the register values get converted to the format of station address # register number (Example: 1#55). This is OK, you can leave the station address prefix alone.
- **ORION only:** If the PLC no Response pop-up box appears when switching screens, it may be caused by the panel trigger tag not being grouped with other registers on the screen. At least one register after the panel trigger may need to be mapped.
- If a hyperlink in the help system generates a windows script error, make sure that fhpopup.ocx is installed and registered.
- To disable the beep on a global project basis see Local System Bit LB-9019, set the value to 0 to disable the beep.

Considerations when replacing an existing MMI-320/640 with an MMI-8xxx

- If the existing MMI-320/640 was using serial communications, note that the serial communications connectors have changed gender and different cables are needed. Use the new CBL-SMLC-MMI8 or CBL-ORM-MMI8 cables documented in this manual.
- The original EasyBuilder 500 Project is needed so that it can be compiled (to create the .EOB file) and imported into EasyBuilder 8000.
- EasyBuilder 8000 uses True Type Fonts, where the previous version used bit mapped fonts. After the project is imported, most text locations have to be adjusted.
- The Communications LED on the face of the terminal is functional for Ethernet only, not RS-232 or RS-485.
- Keypads have a new way of being displayed, but the previous method can also be used.
- Local Word for displaying digits as they are entered on keypads has changed from LW 9071 to LW 9150.
- When importing an existing EB500 project, the recommended font is Franklin Gothic Medium or Arial Black.
- If the first user screen number in the EB500 project was not 10, the screen numbers in the translated project will be shifted one way or the other.
- EB500 Window names are lost when translated to EB8000, they get renamed Window_##.

Appendix A

SMLC to MMI-8xxx Serial Cable

