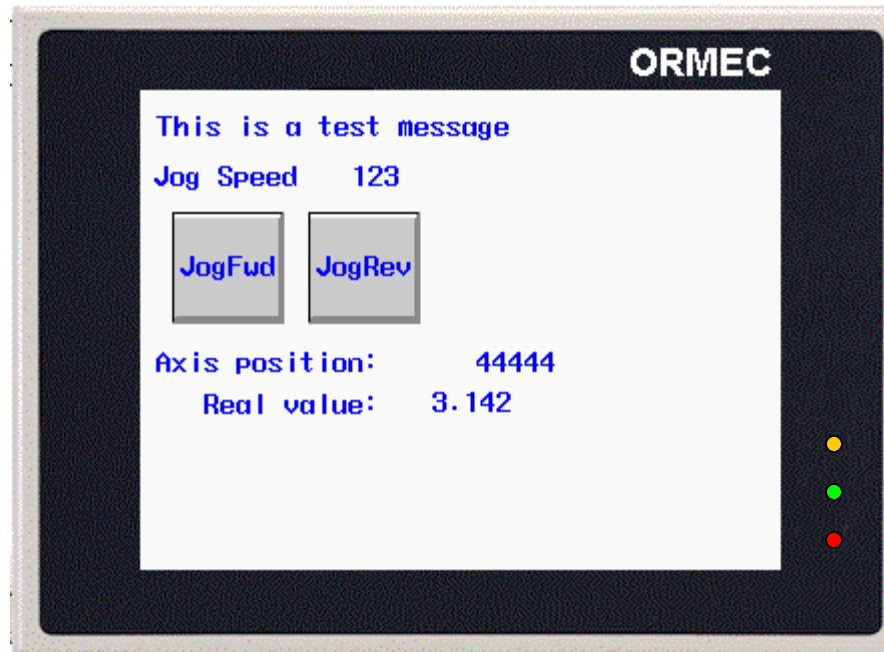


SMLC HMI Programming Tutorial

- In this tutorial:
 - We will configure the SMLC for an external HMI.
 - We will then create an HMI screen that accesses bits, integers, double integers, reals and strings in the SMLC program.
 - We will download and test the screen
- The resultant screen will look like this:

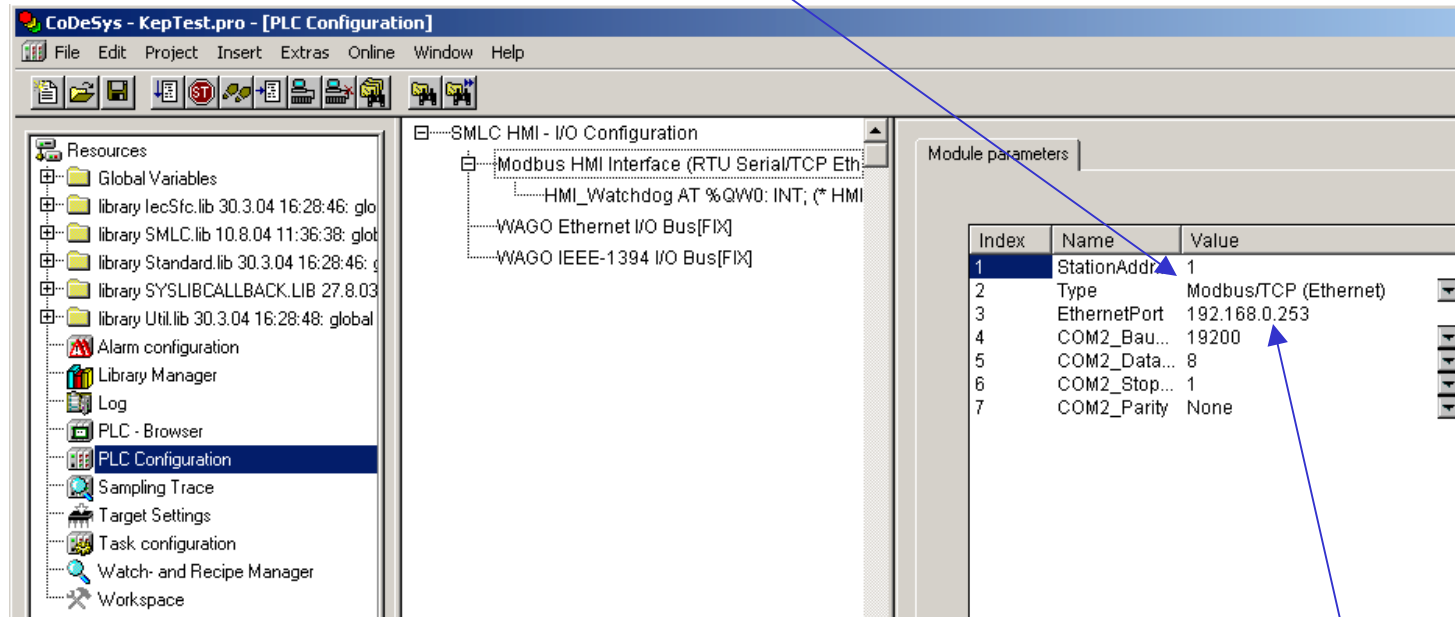


SMLC HMI Configuration

- The SMLC can function as a Modbus RTU slave as well as a Modbus/TCP Server.
- Any Modbus HMI device that can act as a Modbus RTU master or Modbus/TCP Client can communicate with the SMLC.
- Inside the SMLC there is a bank of 2048 “virtual” Modbus registers.
- These registers are like “holding” registers in a Modicon PLC (4xxxx).
- These virtual Modbus registers can be mapped to IEC addresses in blocks of 40 at a time.
- Each 40 registers can be mapped to either:
 - 40 INTs
 - 20 DINTs
 - 20 REALs
 - 1 STRING (80 characters)
- The IEC address corresponds to the Modbus register number (e.g. %QW1 = Register 1, %QW200 = Register 200, etc.).
- If you want to access a bit from the HMI you access a bit within a register (e.g. bit 0 of register 1).
- The HMI cannot directly access the SMLC’s physical I/O.
- HMI variables are initialized to 0 at SMLC powerup so if you want HMI values to be retain/persistent you need to copy the retain/persistent values to the HMI once after powerup and then copy the HMI values back to the retain/persistent variables every scan after that.

SMLC - Configuring the HMI for Modbus/TCP (Ethernet)

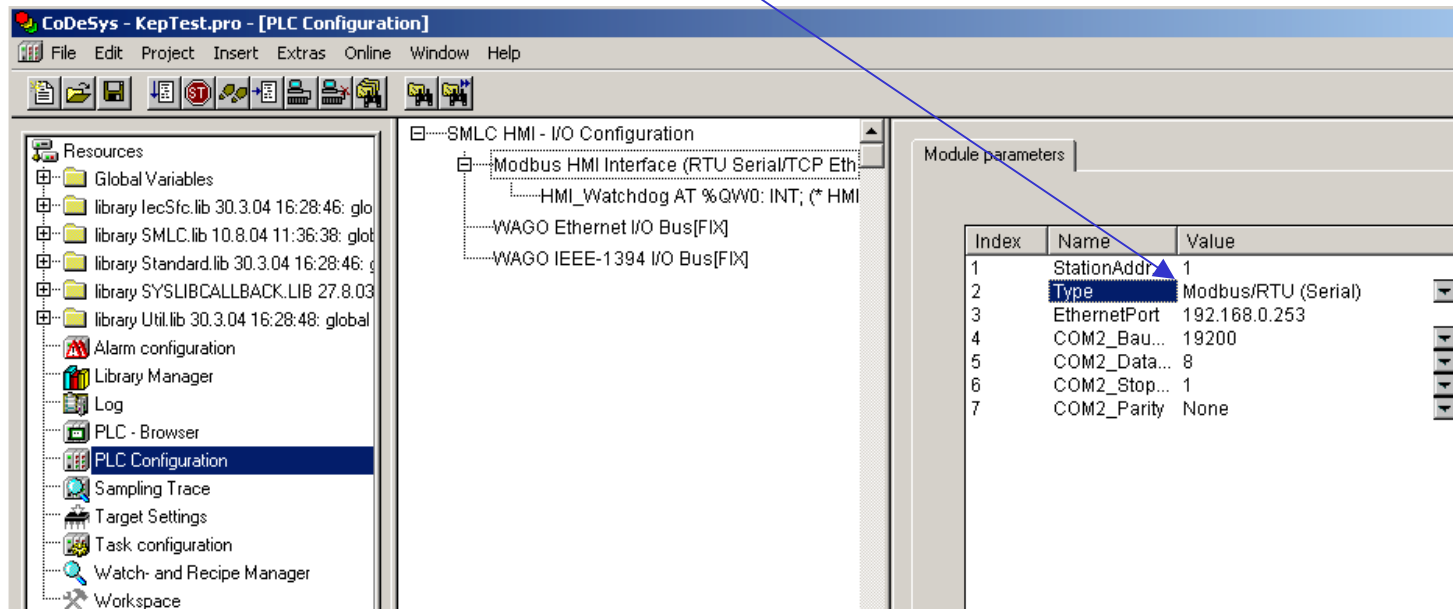
- Select the PLC Configuration Resource on the Resources tab
- Set the Type to Modbus/TCP (Ethernet)



- Set the EthernetPort field to the IP address of the port **on the SMLC** that you want to connect the HMI to.

--OR-- SMLC - Configuring the HMI for serial Modbus/RTU

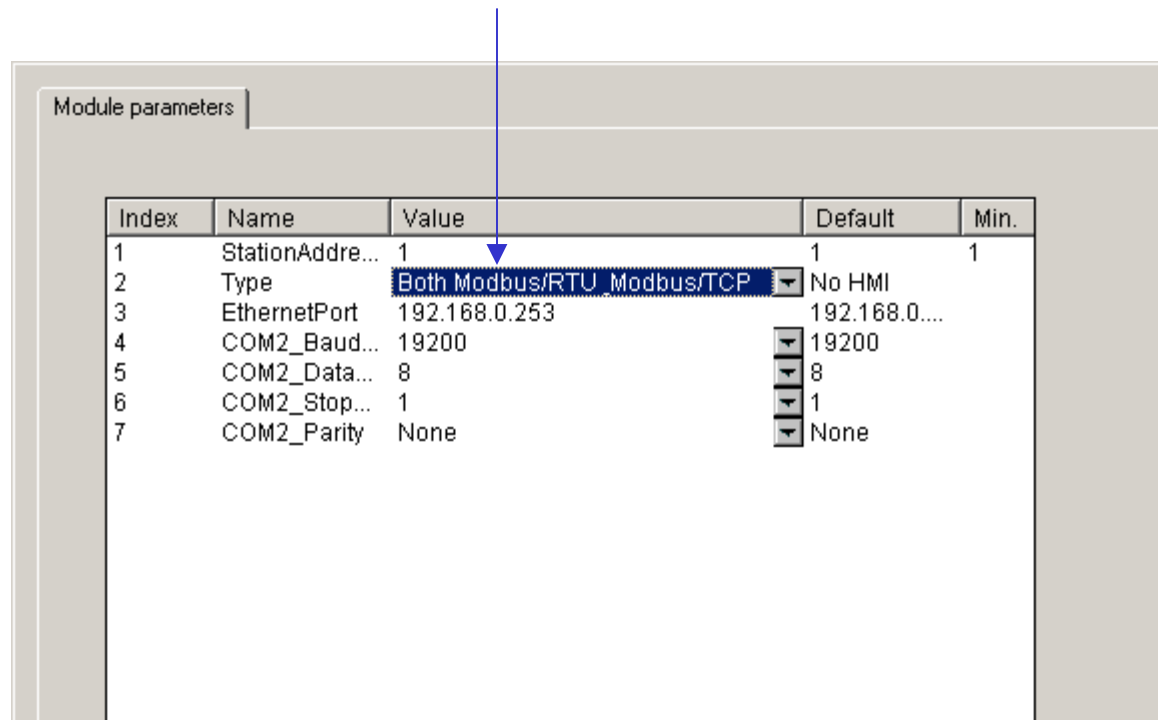
- Select the PLC Configuration Resource on the Resources tab
- Set the Type to Modbus/RTU (Serial)



- The serial Modbus server is on COM2 and uses the baud rate, data bits, stop bits and parity settings above. These must match the settings on your Modbus master device!

--OR-- SMLC - Configuring the HMI for both

- You can also configure the SMLC for both Modbus/RTU and Modbus/TCP simultaneously.



- This can be useful if you have a local, serial HMI as well as a remote, Ethernet based HMI or HMI package.

SMLC - Variable naming conventions

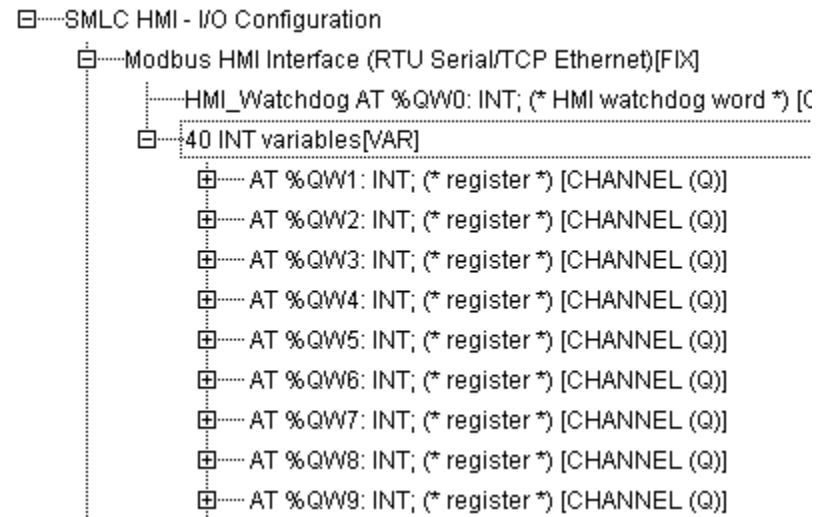
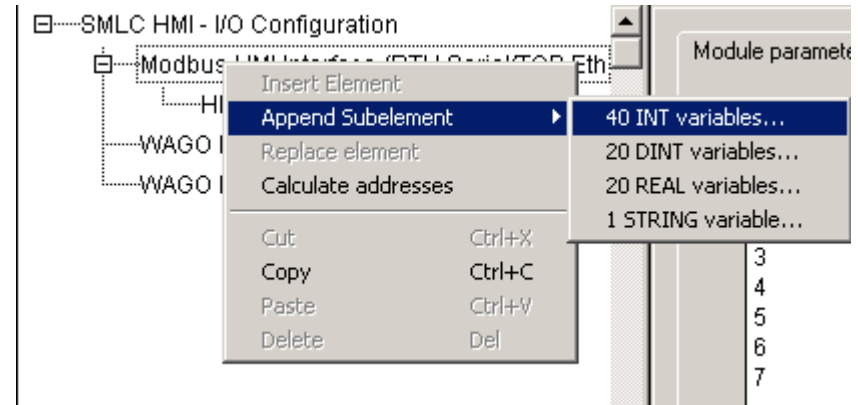
- Because IEC-61131-3 languages are strongly typed we use a variable naming convention to indicate a variables type. We prefix variable names with a lower case letter or letters based on the following types:

Prefix	Type		Lower limit	Upper limit	Memory space
b	BOOL	Boolean	0	1	8 Bit
by	BYTE	Byte	0	255	8 Bit
w	WORD	Word	0	65535	16 Bit
dw	DWORD	Double word	0	4294967295	32 Bit
si	SINT:	Short integer	-128	127	8 Bit
usi	USINT:	Unsigned short integer	0	255	8 Bit
i	INT:	Integer	-32768	32767	16 Bit
ui	UINT:	Unsigned integer	0	65535	16 Bit
di	DINT:	Double integer	-2147483648	2147483647	32 Bit
udi	UDINT:	Unsigned double integer	0	4294967295	32 Bit
r	REAL	Real (float)	$\pm 3.4 \times 10^{-38}$	$\pm 3.4 \times 10^{38}$	32 Bit
lr	LREAL	Long real (float)	$\pm 1.07 \times 10^{-308}$	$\pm 1.07 \times 10^{308}$	64 Bit
t	TIME	Time			
d	DATE	Date			
tod	TIME_OF_DAY	Time of day			
dt	DATE_AND_TIME	Date and time			
s	STRING	String			

- We also prefix global variables with a lower case g before the type identifier.
- All HMI and I/O variables are global in CoDeSys.

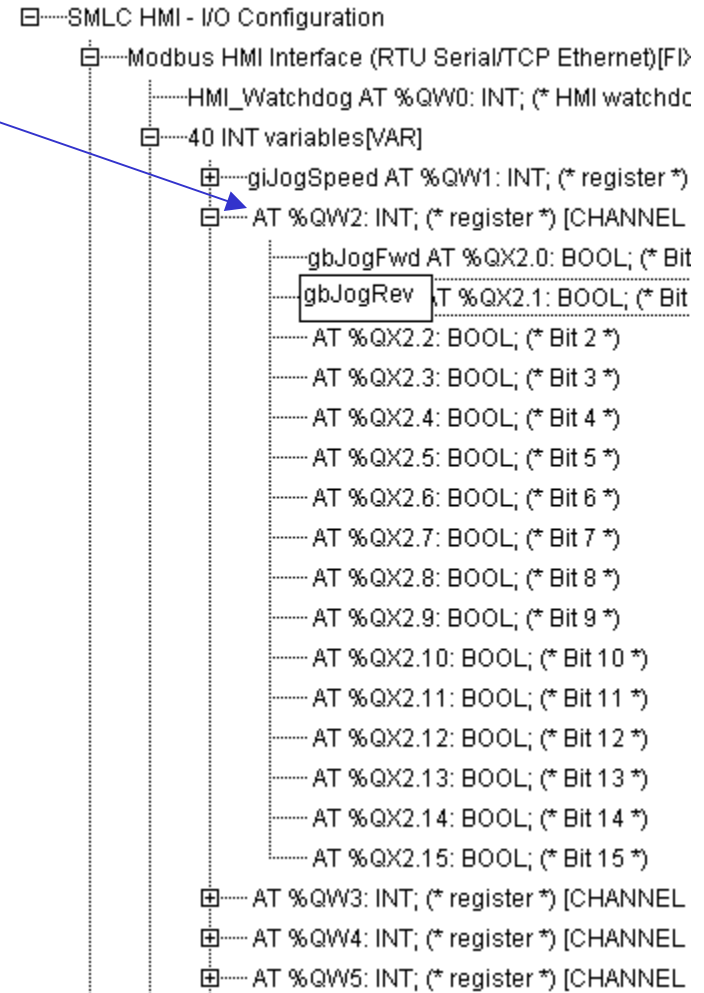
SMLC - Adding INT variables

- Right click on the Modbus HMI Interface branch of the tree and Append Subelement 40 INT variables.
- This maps 40 Integer variables to the Modbus registers in your SMLC.
- These variables are global variables in your SMLC program
- The Modbus register number is equivalent to the IEC address of the variable, e.g. %QW1 = Modbus register 1.



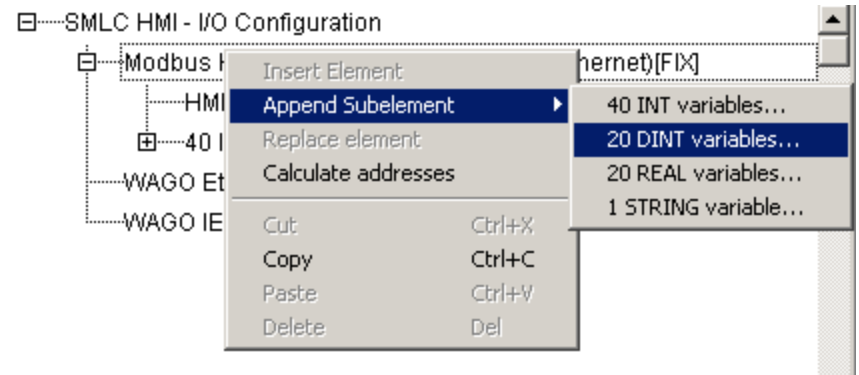
SMLC - Assigning names to the INT variables

- To assign a variable name to a register click on the word AT in front of the IEC address.
- Type in giJogSpeed as the name of Modbus register 1 (located at %QW1). We prefix the variable with a lower case g for global and i for INT
- Type in gbJogFwd as the name for bit 0 in Modbus register 2. We prefix the variable with a lower case g for global and b for BOOL.
- Type in gbJogRev as the name for bit 1 in Modbus register 2. We prefix the variable with a lower case g for global and b for BOOL.

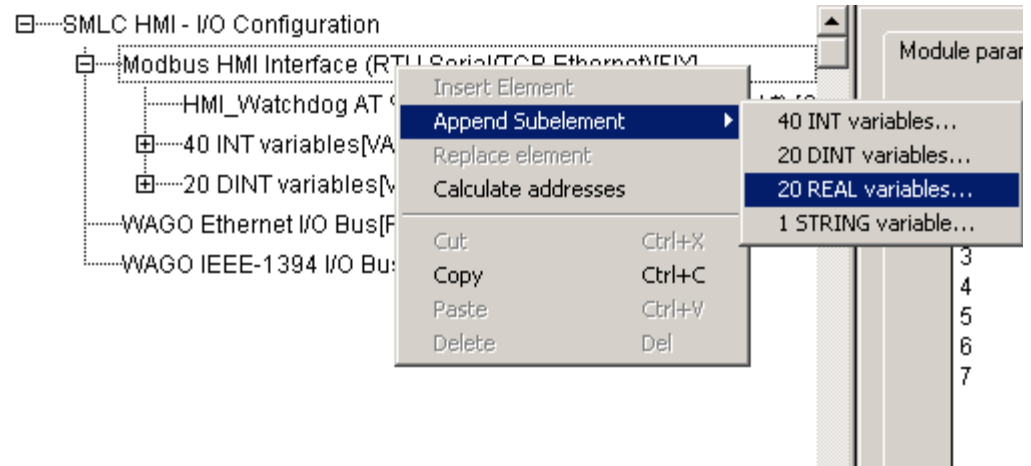


SMLC - Adding DINT and REAL variables

- Now add 20 DINT variables

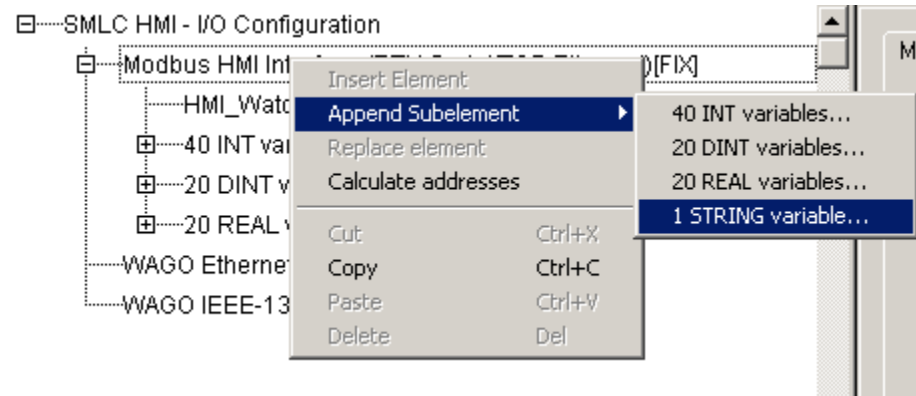


- Now add 20 REAL variables

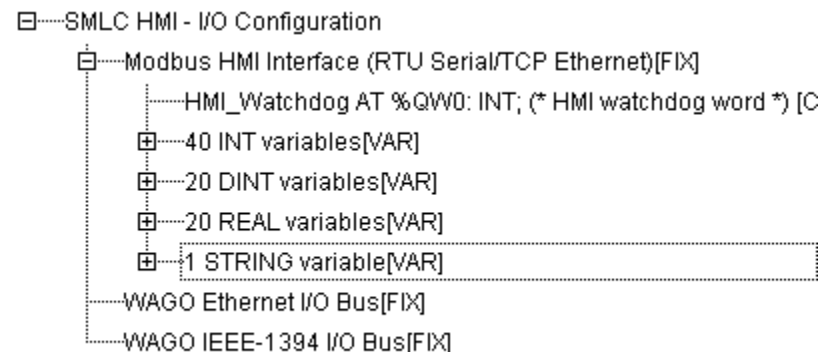


SMLC - Adding STRING variables

- Now add 1 STRING variable

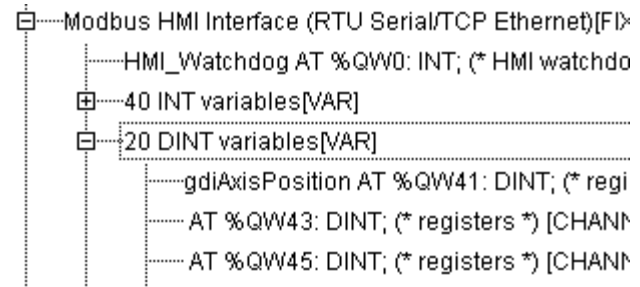


- Your configuration should look like this:

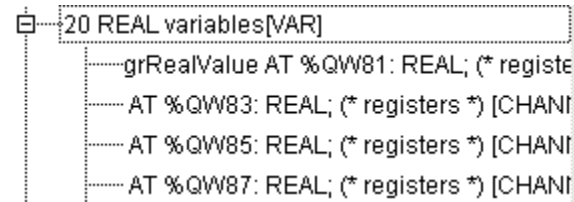


SMLC - Assigning names to the DINT, REAL and STRING variables

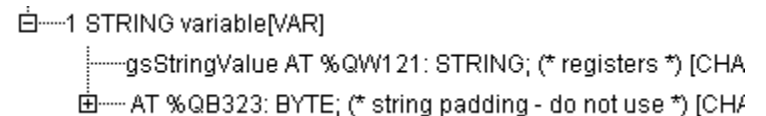
- Name the DINT variable at %QW41
gdiAxisPosition



- Name the REAL at %QW81
grRealValue



- Name the STRING at %QW121
gsStringValue



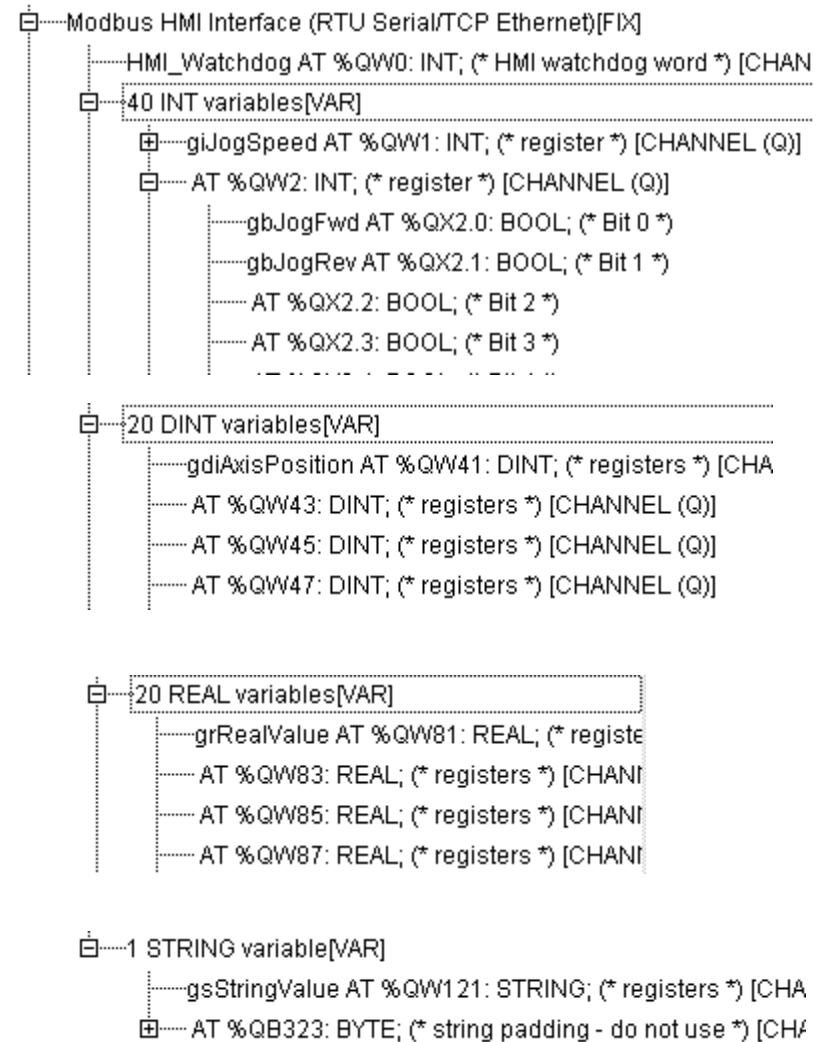
SMLC - Summary of Variable to Modbus register mapping

- giJogSpeed 1
- gbJogFwd 2/00 (word 2, bit 0)
- gbJogRev 2/01 (word 2, bit 1)

- gdiAxisPosition 43

- grRealValue 81

- gsStringValue 121



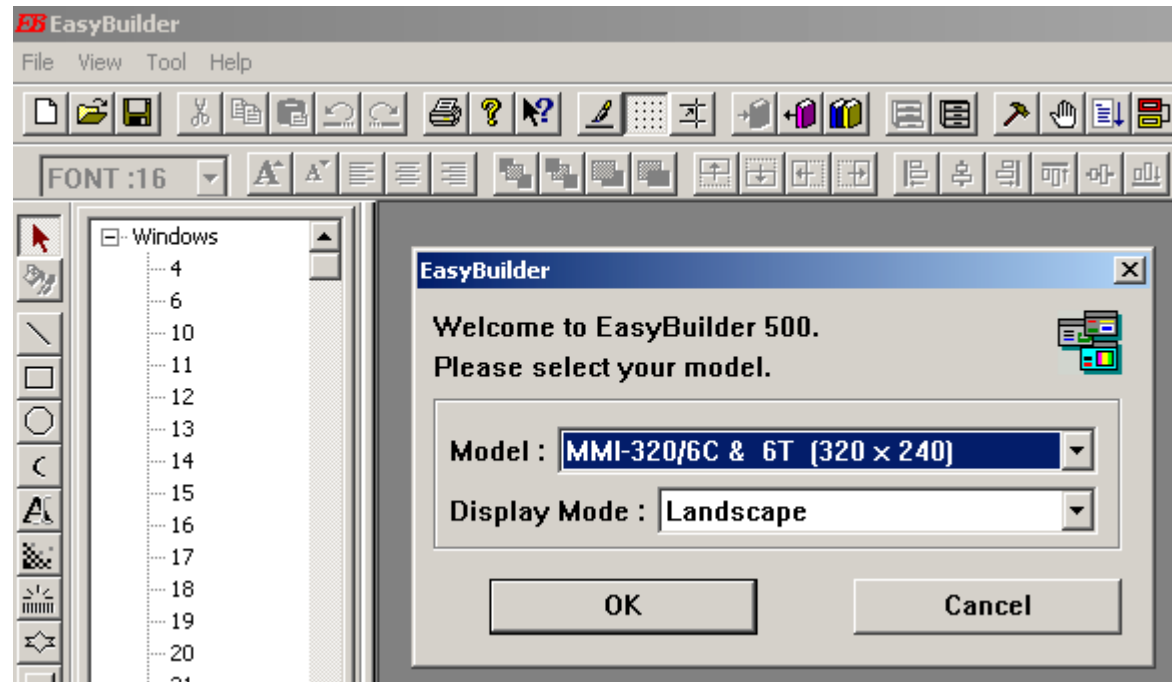
HMI - Creating a new project

- Run EasyManager
- Make sure that the COM port selection matches the comm port that is connected from your PC to the HMI. This is the port you will use to download the screen to the HMI
- Click on the EasyBuilder button.



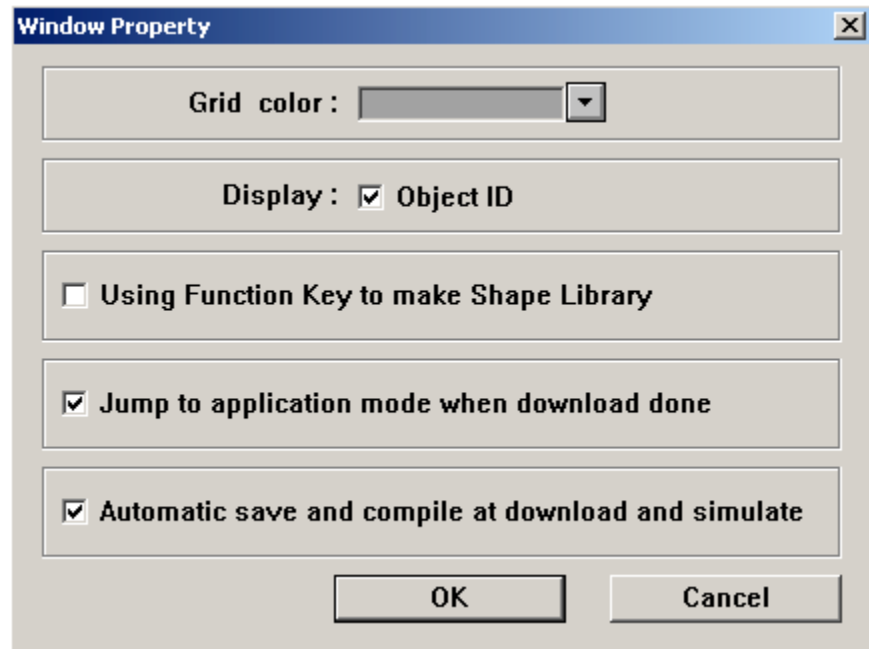
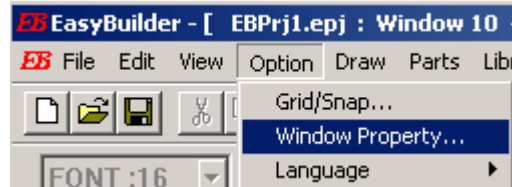
HMI - Creating a new project

- Select the appropriate display model



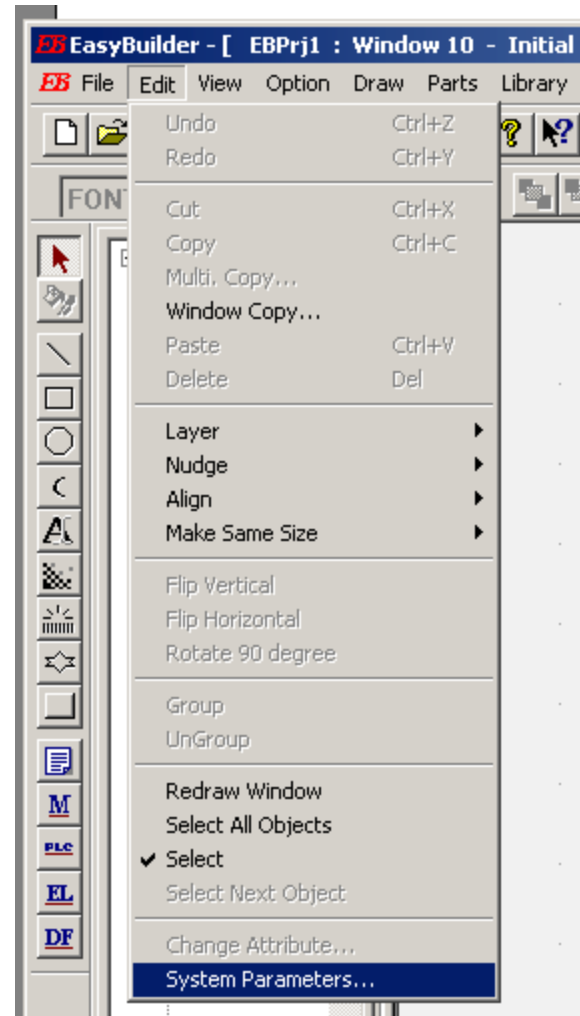
HMI - Useful EasyBuilder settings

- There are a couple of global settings in EasyBuilder that will make your life easier by reducing the number of acknowledgements you'll have to make when you compile and download a new HMI program.
- Go to Option | Window Property
- Check "Jump to application mode when download done"
- Check "Automatic save and compile at download and simulate"



HMI - Setting the System Parameters

- Go to Edit | System Parameters



HMI - Ethernet Modbus/TCP Configuration

- Select MODBUS RTU TCP/IP as the PLC type.
- Select Ethernet for the PLC I/F.
- Set the TCP/IP Delay to 5 (this is in 10's of msec and is the delay between transactions).
- Set Multiple HMI to Master.
- Set Connect I/F to Ethernet.
- Set the Local IP address to the desired IP address for the touchscreen.
- Set the Server IP address to the IP address of the SMLC.
- Set the Subnet mask.
- Set the default Router IP address to the same subnet as the touchscreen and the SMLC (first three octets match, fourth octec should be 0).

The screenshot shows the 'Set System Parameter' dialog box with the following configuration:

- PLC type: MODBUS RTU TCP/IP
- HMI model: MMI-320/6C & 6T (320 x 240)
- PLC I/F port: Ethernet
- Baud rate: 19200
- Data bits: 8 Bits
- Parity: None
- Stop bits: 1 Bit
- Comm Delay (mS): 20
- TCP/IP Delay: 5
- Parameter 3: 0
- Parameter 4: 0
- Parameter 5: 0
- Parameter 6: 0
- HMI station No.: 0
- PLC station No.: 1
- Multiple HMI: Master
- HMI-HMI link speed: 115200
- Connect I/F: Ethernet
- Local IP address: 192 . 168 . 0 . 250
- Server IP address: 192 . 168 . 0 . 253
- Subnetwork Mask: 255 . 255 . 255 . 0
- Default Route IP address: 192 . 168 . 0 . 0
- PLC time out constant (sec): 3.0
- PLC block pack: 0

HMI - Serial Modbus RTU configuration

- Select ORMEC SMLC Modbus RTU.
- Select RS-232 for the PLC I/F.
- The remaining default values should be ok.

The screenshot shows the 'Set System Parameter' dialog box with the following configuration:

- PLC type: ORMEC SMLC Modbus RTU
- HMI model: MMI-320/6C & 6T (320 x 240)
- PLC I/F port: RS-232
- Baud rate: 19200
- Data bits: 8 Bits
- Stop bits: 1 Bit
- Parity: None
- Comm Delay (mS): 20
- TCP/IP Delay: 0
- Parameter 3: 0
- Parameter 4: 0
- Parameter 5: 0
- Parameter 6: 0
- HMI station No.: 0
- PLC station No.: 1
- Multiple HMI: Disable
- HMI-HMI link speed: 115200
- Connect I/F: Serial
- Local IP address: 0 . 0 . 0 . 0
- Server IP address: 0 . 0 . 0 . 0
- Subnetwork Mask: 0 . 0 . 0 . 0
- Default Route IP address: 0 . 0 . 0 . 0
- PLC time out constant (sec): 3.0
- PLC block pack: 0

HMI - Adding a numeric input for the Jog Speed

- Select the numeric input object



Numeric Input Extend Object's Attribute

General Numeric Shape Font

Description :

Read address

Device type : Device address :

No. of words :

Trigger address :

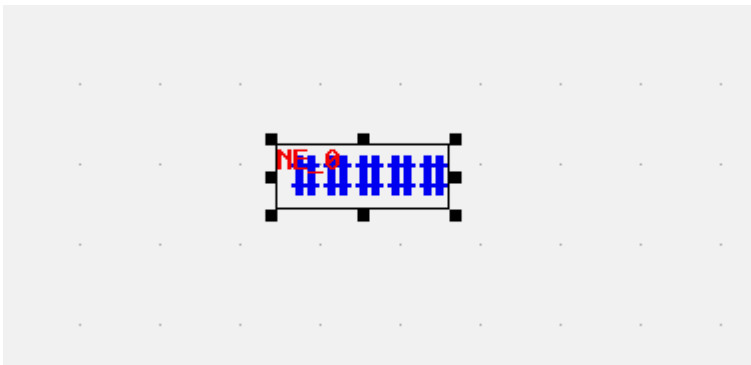
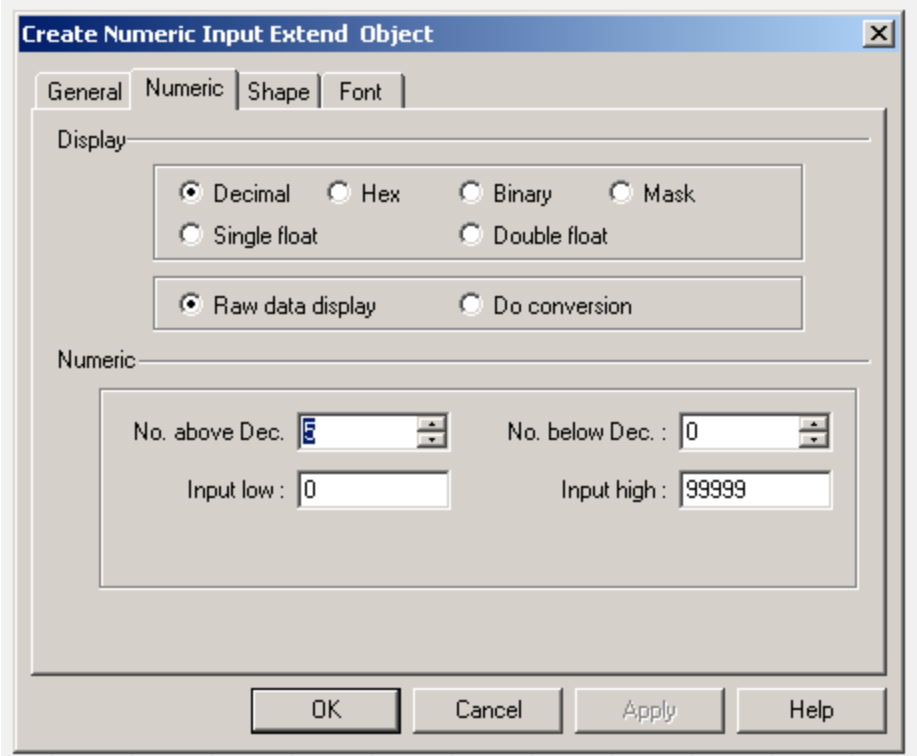
Device type : Device address :

OK Cancel Apply Help

- Select Hold reg(4x) as the Device type. Holding registers are the kind of Modbus registers that the SMLC has.
- Enter 1 for the Device address. This is Modbus register 1 or %QW1 in the SMLC.
- The trigger address is used to enable operator entry. Internal bit 9000 is always TRUE. Using this address will always allow operator entry of this value.

HMI - Adding a numeric input for the Jog Speed, part 2

- Go to the Numeric tab
- Change the No. above Dec. value to 5.
This controls the number of digits we will see on the screen.
- Press the OK button



- The screen should now look like this

HMI - Adding a pushbutton for JogFwd

- Select the toggle switch object



Toggle Switch Object's Attribute

General | Shape | Label | Profile

Description :

Read address :
Device type : Device address :

Write address :
Device type : Device address :

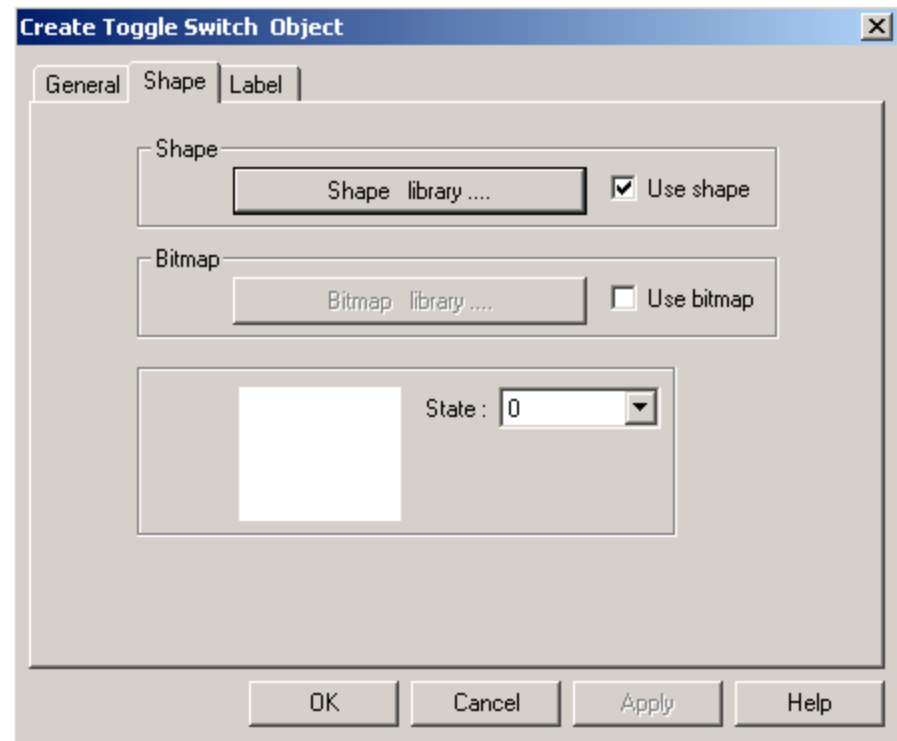
Attribute
Switch style :

OK Cancel Apply Help

- Select Hold reg/bit(4x/bb) as the device type for the Read address.
- Enter 2/00 as the Device address for the Read address.
- Select Hold reg/bit(4x/bb) as the device type for the Write address.
- Enter 2/00 as the Device address for the Write address.
- Select Momentary as the Switch style. This makes this a momentary pushbutton.

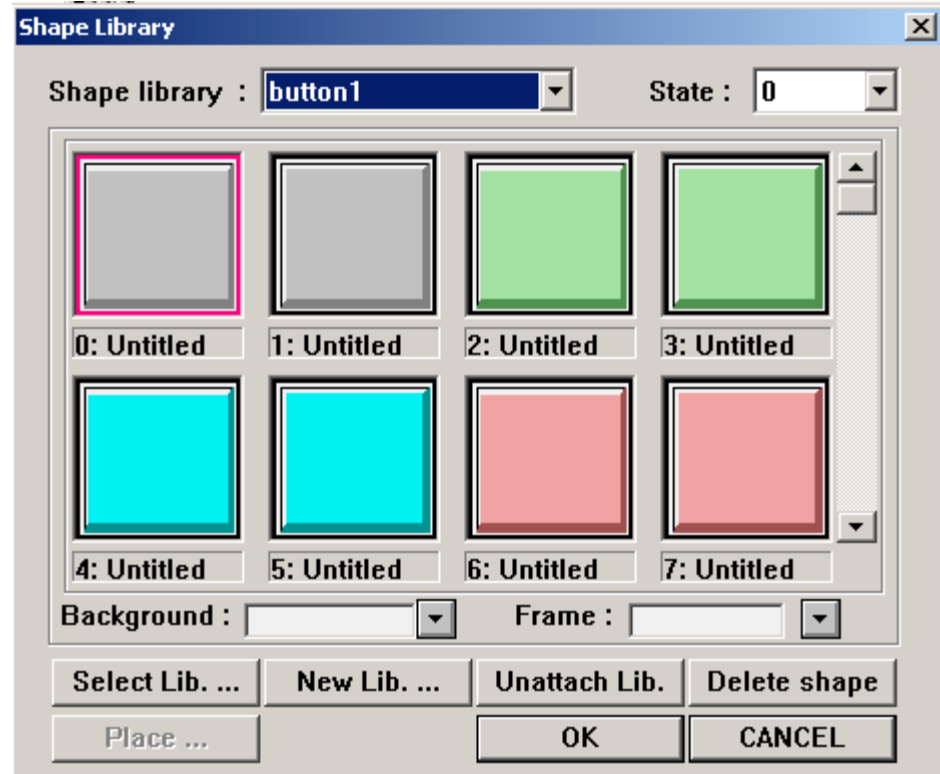
HMI - Adding a pushbutton for JogFwd, part 2

- Select the Shape tab
- Check the Use shape box
- Click on the Shape library button



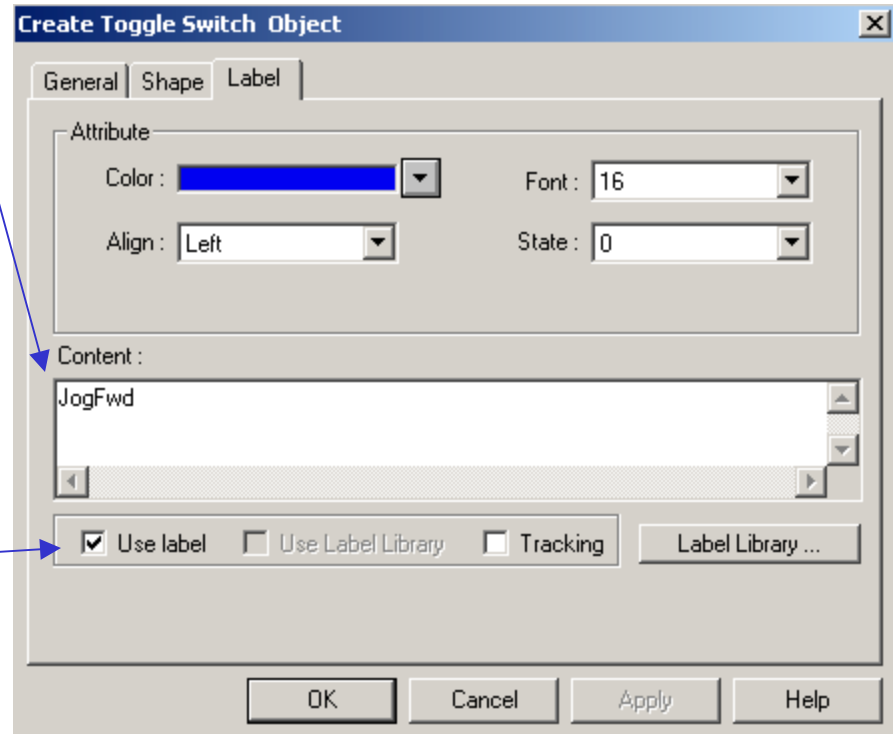
HMI - Adding a pushbutton for JogFwd, part 3

- Select an image from the Shape library. You can choose a different library from the drop down list



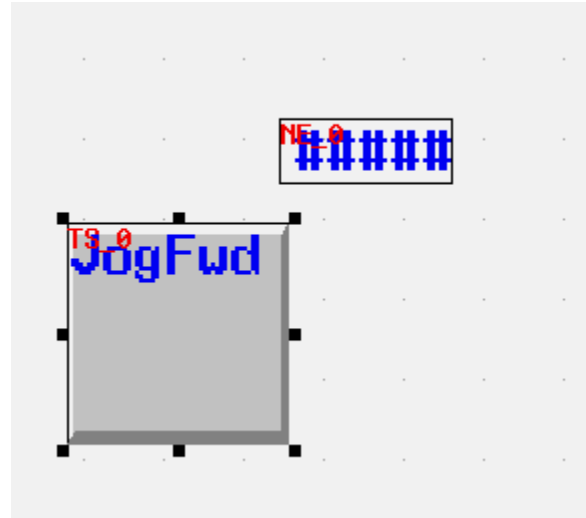
HMI - Adding a pushbutton for JogFwd, part 4

- Enter JogFwd as the label Content
- Note that this is the label for State 0. If we wished we could enter a different label for State 1 (when the button is pressed).
- Check the Use label box



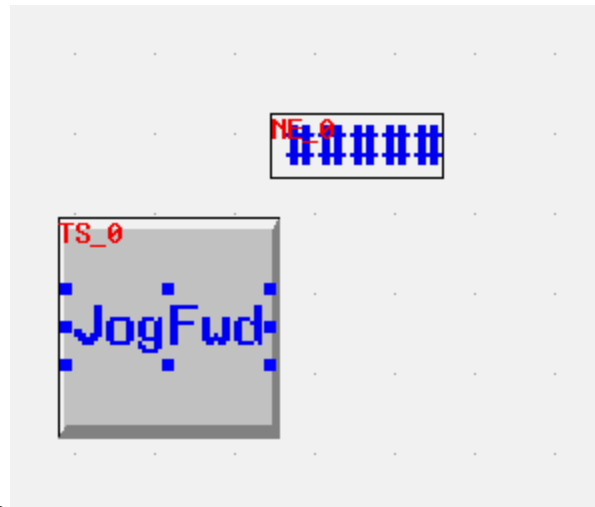
HMI - Adding a pushbutton for JogFwd, part 5

- The button should now appear on the screen.



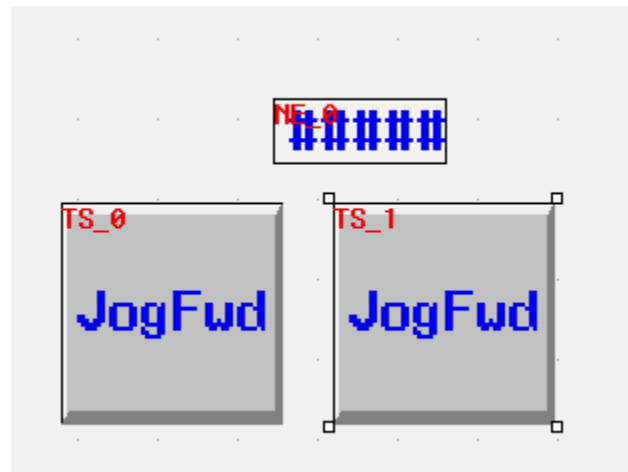
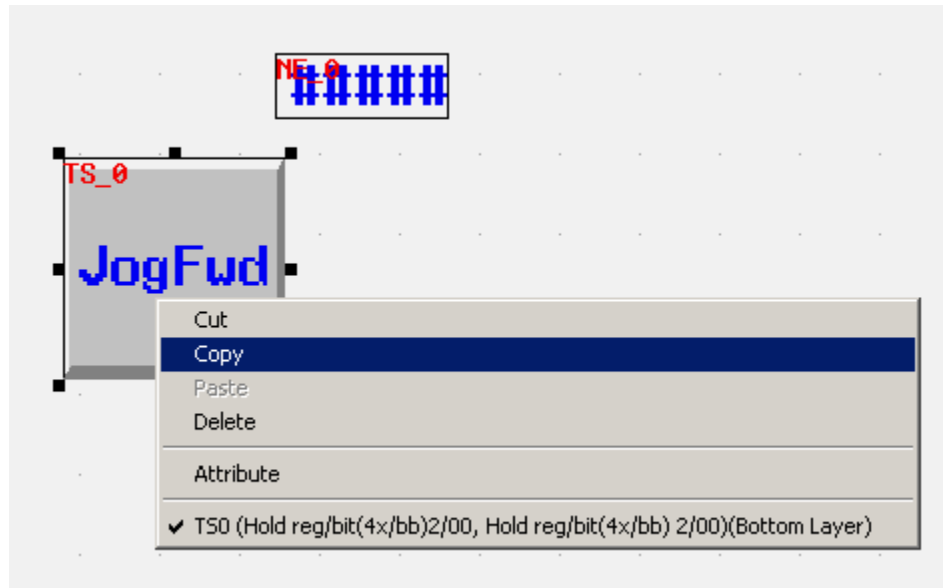
- Click once on the text on the button, the selection squares should appear.

- Drag the text to an appropriate location on the button.



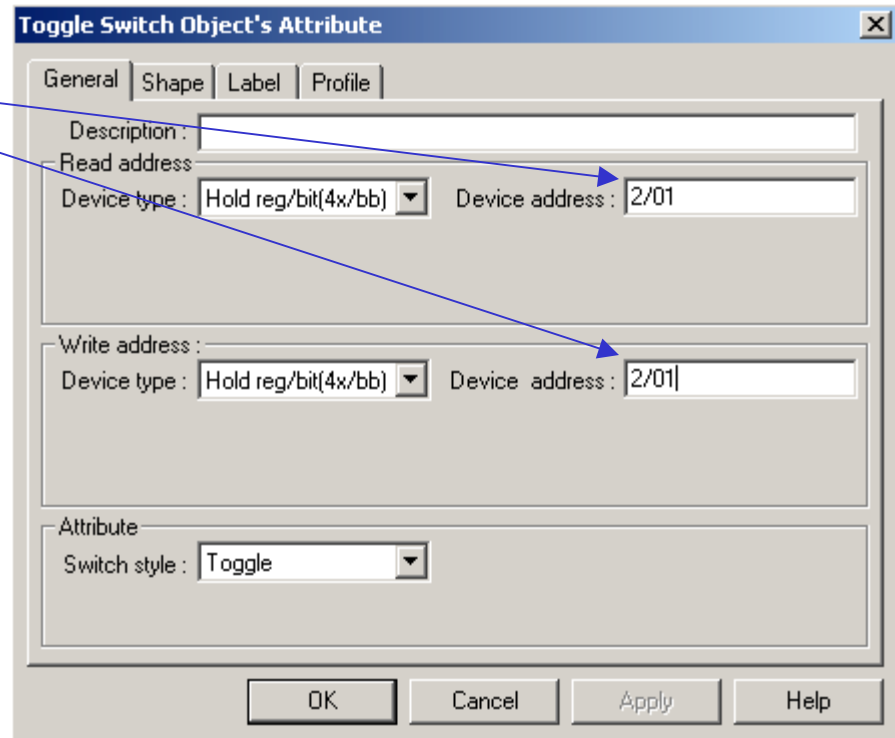
HMI Adding a pushbutton for JogRev

- To create the JogRev pushbutton lets copy, paste and edit the JogFwd button.
- Right-click on the JogFwd button and select Copy.
- Right click again and select Paste.
- Drag the new button to an appropriate location



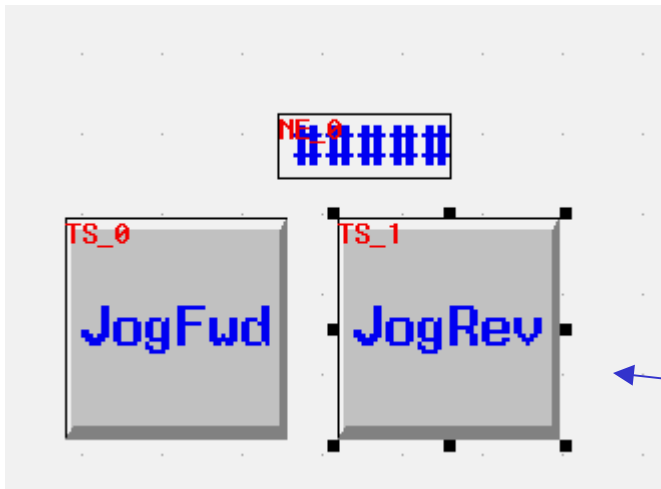
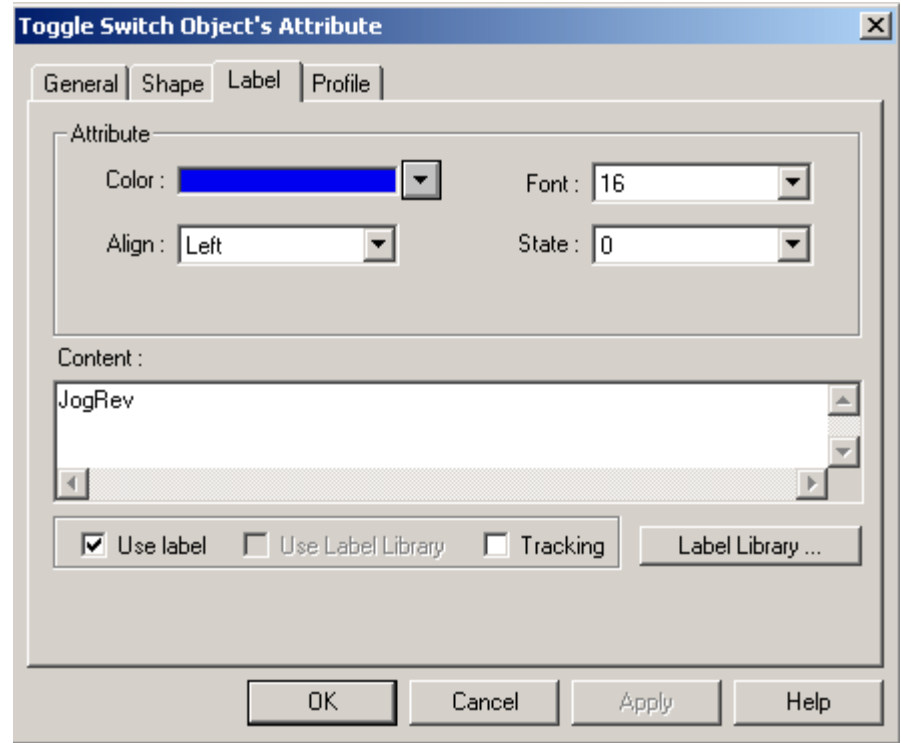
HMI - Adding a pushbutton for JogRev, part 2

- Double click on the new button to edit it.
- Change the Device Address of both the Read and Write to 2/01



HMI - Adding a pushbutton for JogRev, part 3

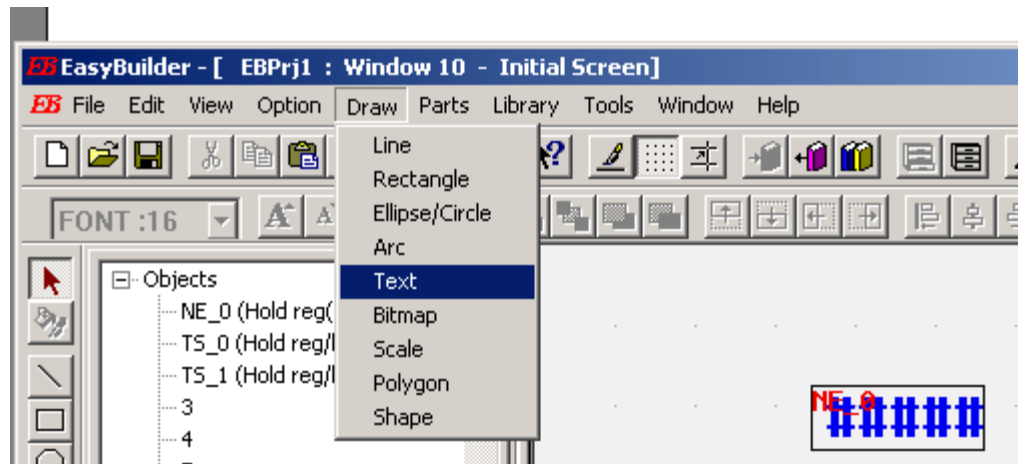
- Go to the label tab and change the text in the Content to “JogRev”
- Press OK when done



- The screen should now look like this

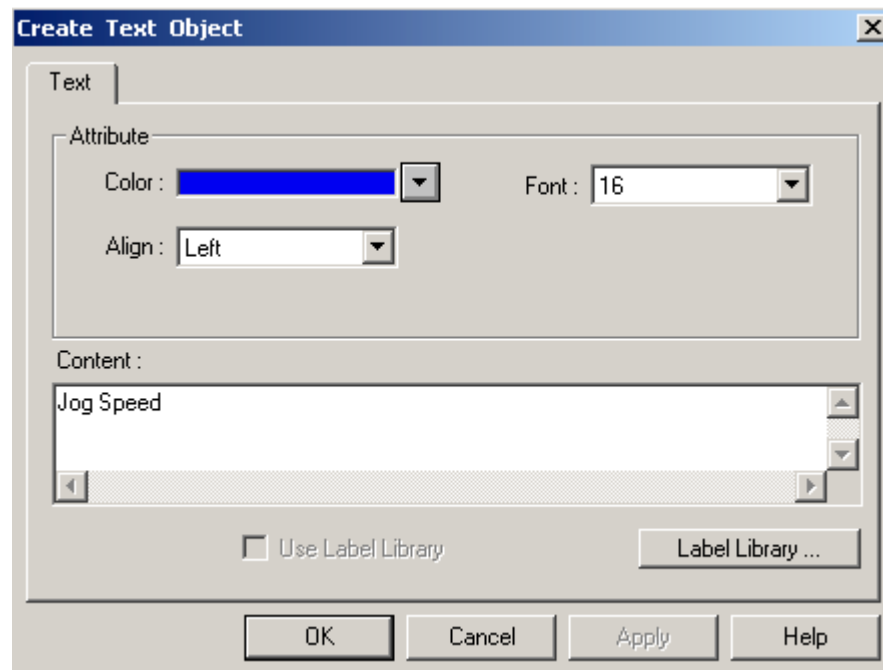
HMI - Adding a Text label for the Jog Speed

- Now let's add a text label to the left of the Jog Speed numeric entry.
- Go to the Draw | Text menu item.



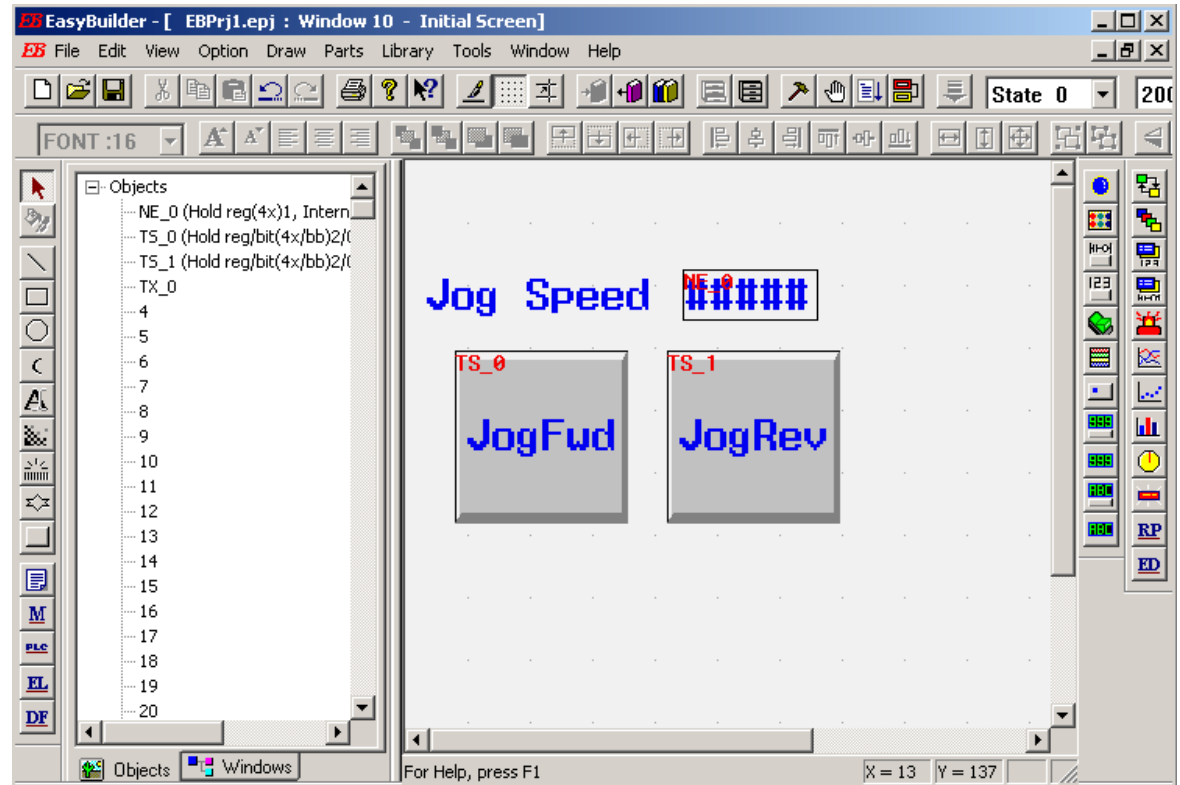
HMI - Adding a text label for the Jog Speed, part 2

- Enter Jog Speed as the Content.
- Press OK and drag the text to an appropriate location.



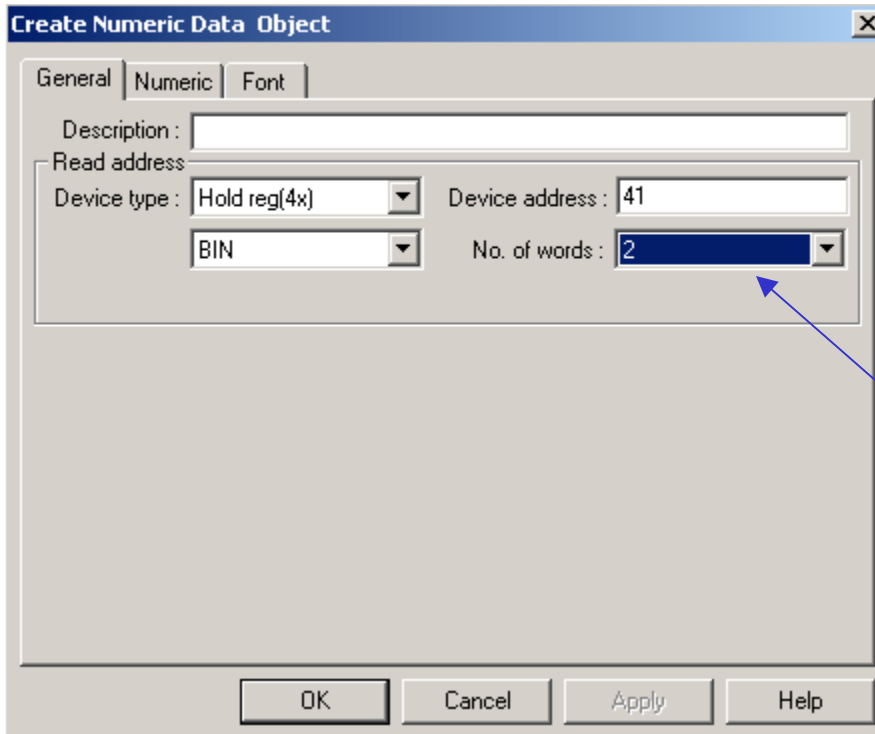
HMI - Adding a text label for the Jog Speed, part 3

- The screen should now look like this
- In the left hand pane select the Object tab. Note how you can use this as an alternate way to select individual screen objects for editing. This is useful when you build more complex screens that have objects stacked on top of each other.



HMI - Adding the axis position display (DINT)

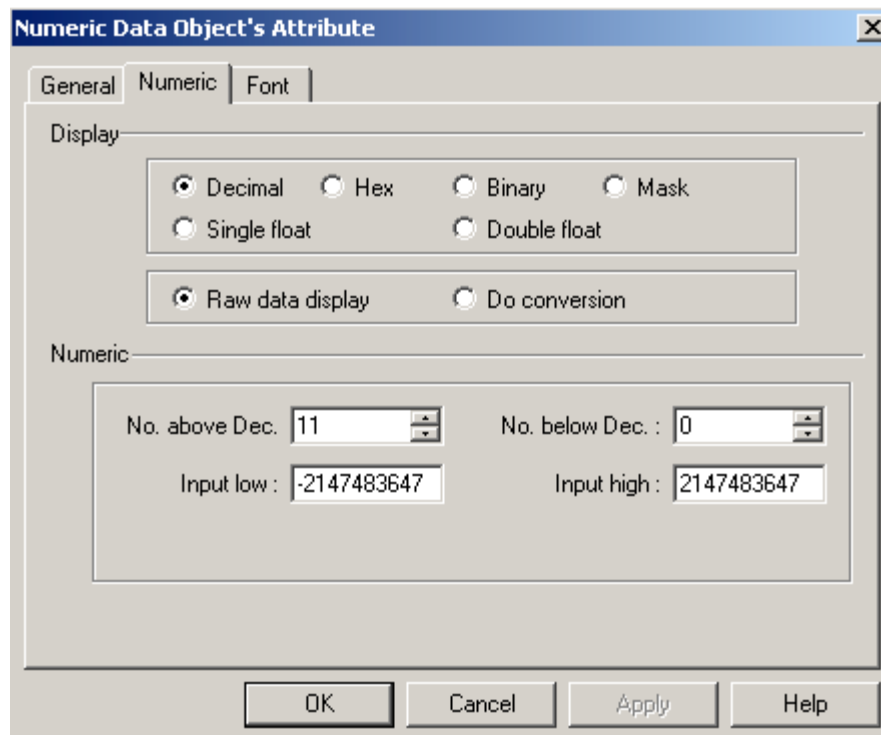
- Select the numeric data display object



- Select Hold reg(4x) as the Device type.
- Enter 41 for the Device address. This is Modbus register 41 or %QW41 in the SMLC.
- Set the No. of words to 2 because this is a DINT which is 32 bits.
- Select the Numeric tab

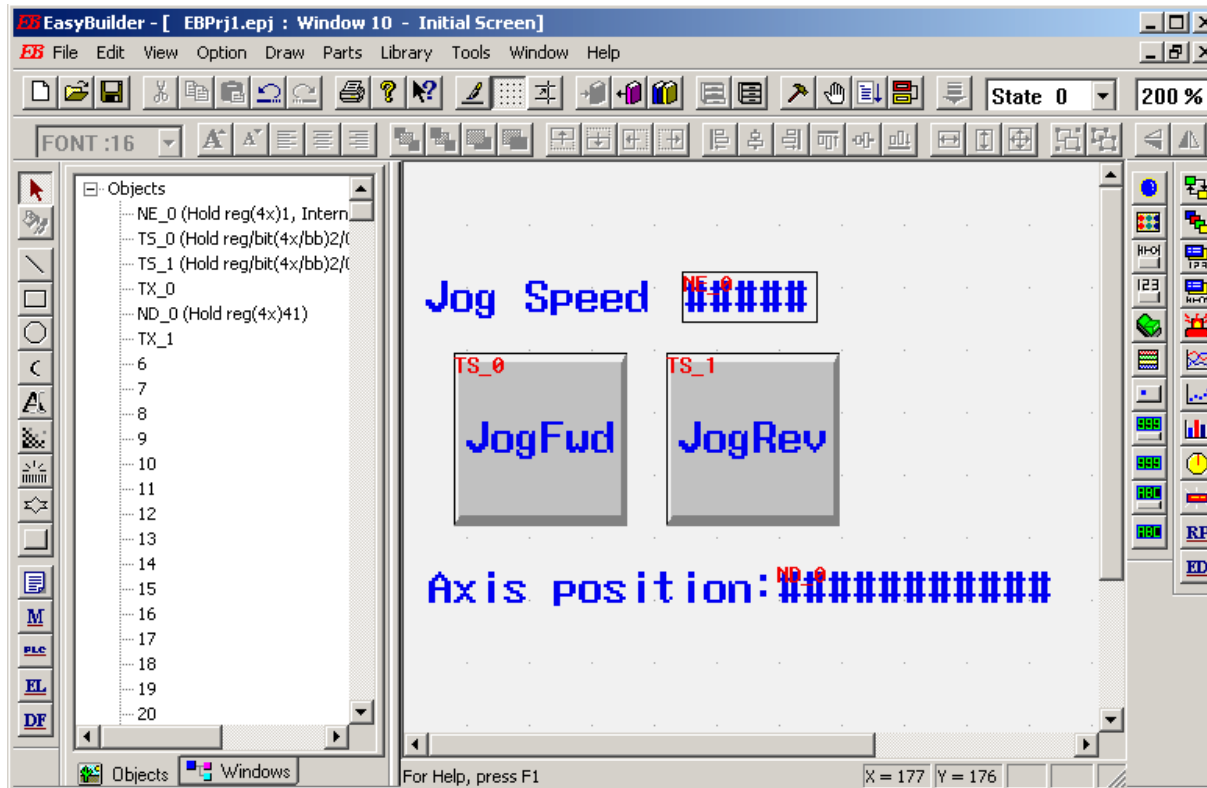
HMI - Adding the axis position display (DINT), part 2

- Set the No. above Dec. to 11.
- Note that the input high is automatically set as you increase the number of digits above the decimal.
- Set the Input low to the most negative DINT -2147483647.



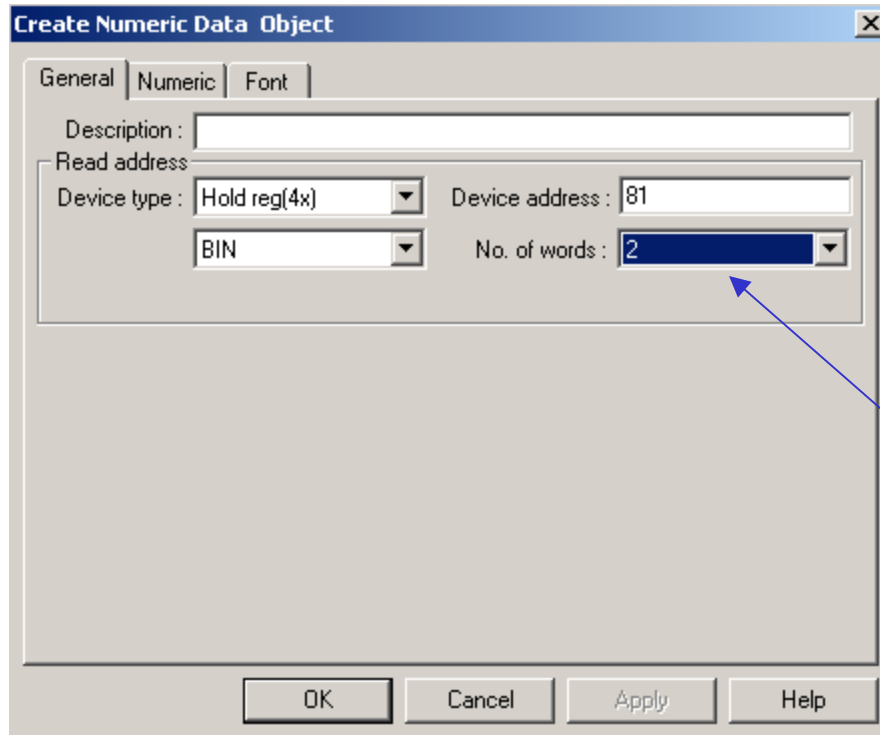
HMI - Adding the axis position display (DINT), part 3

- Add a text label for the axis position just like we did for the jog speed (use the Draw | Text menu).
- The screen should now look like this:



HMI - Adding a REAL display

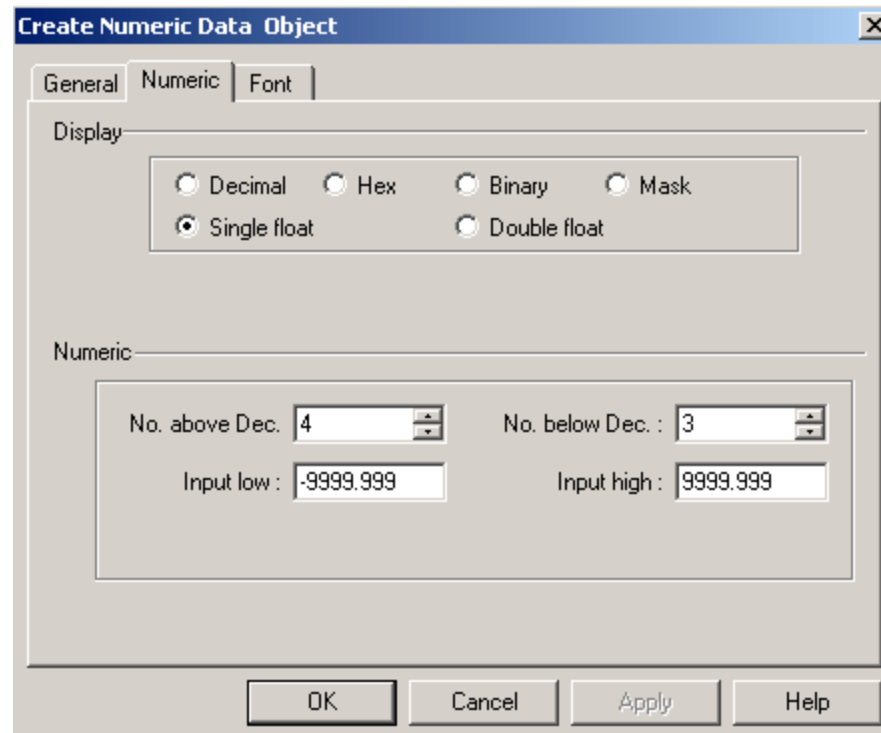
- Select the numeric data display object



- Select Hold reg(4x) as the Device type.
- Enter 81 for the Device address. This is Modbus register 81 or %QW81 in the SMLC.
- Set the No. of words to 2 because this is a REAL which is 32 bits.
- Select the Numeric tab

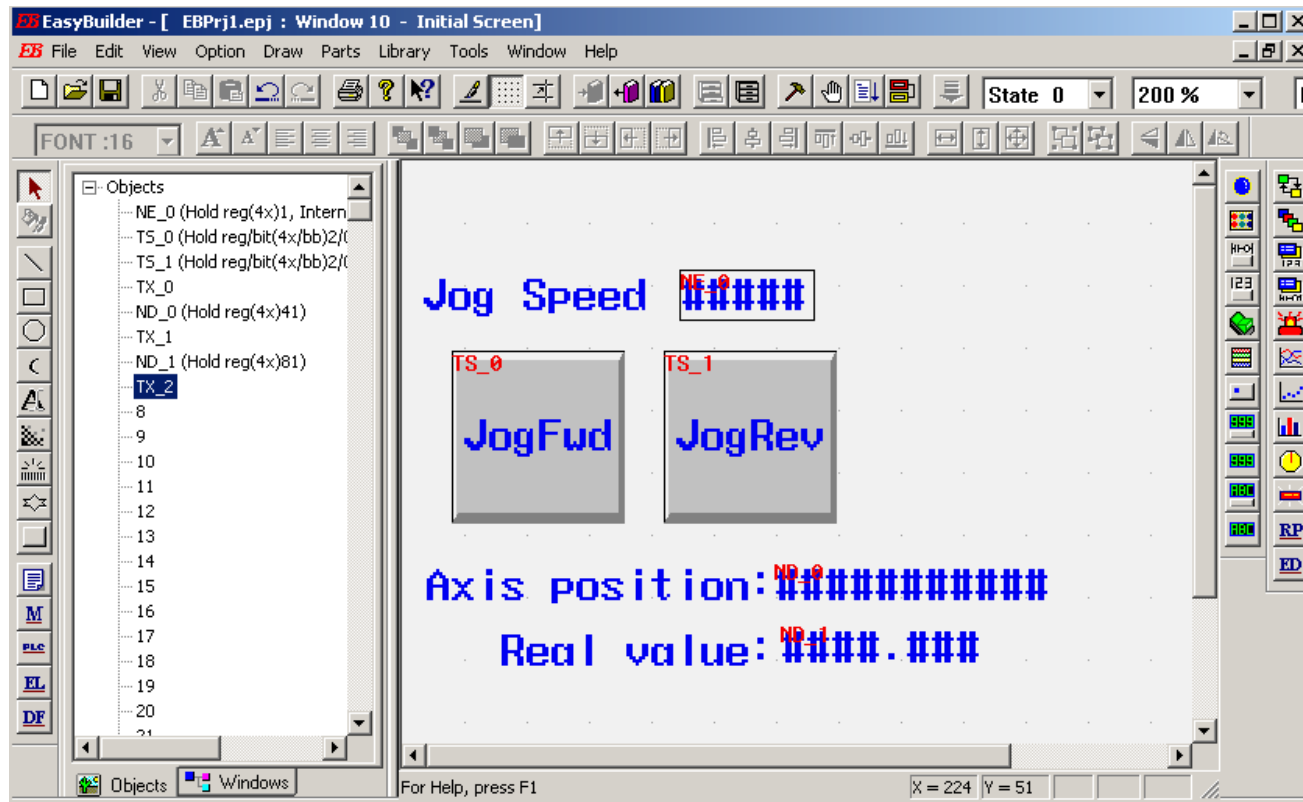
HMI - Adding a REAL display, part 2

- Set the No. above Dec. to 4 and No. below Dec. to 3
- Set the Input high to 9999.999
- Set the Input low to -9999.999



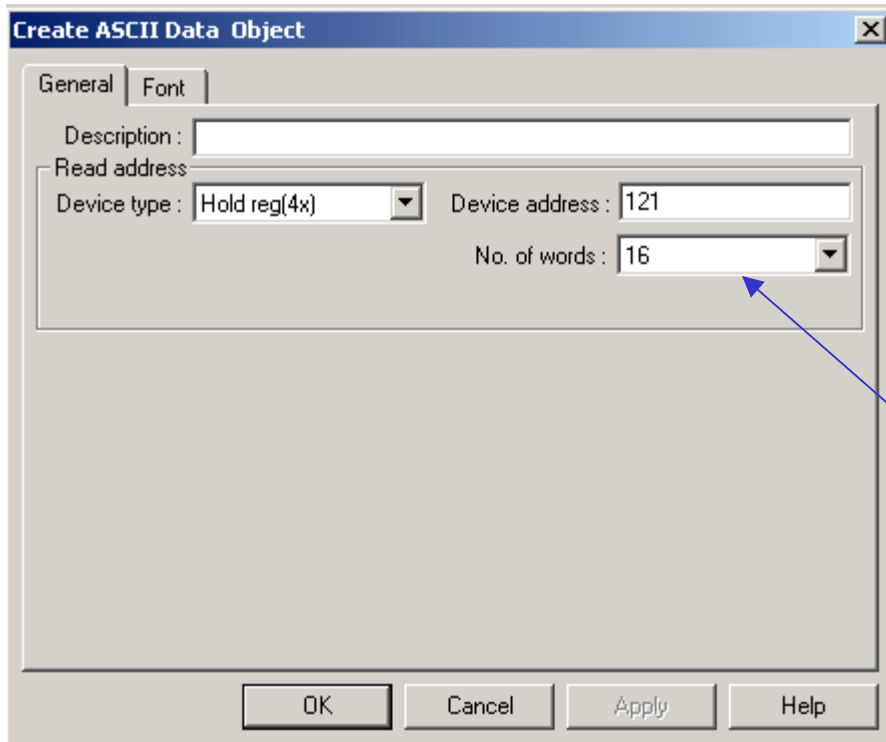
HMI - Adding a REAL display, part 3

- Add a text label for the REAL value just like we did for the Jog Speed and Axis position (use the Draw | Text menu).
- The screen should now look like this:



HMI - Adding a STRING display

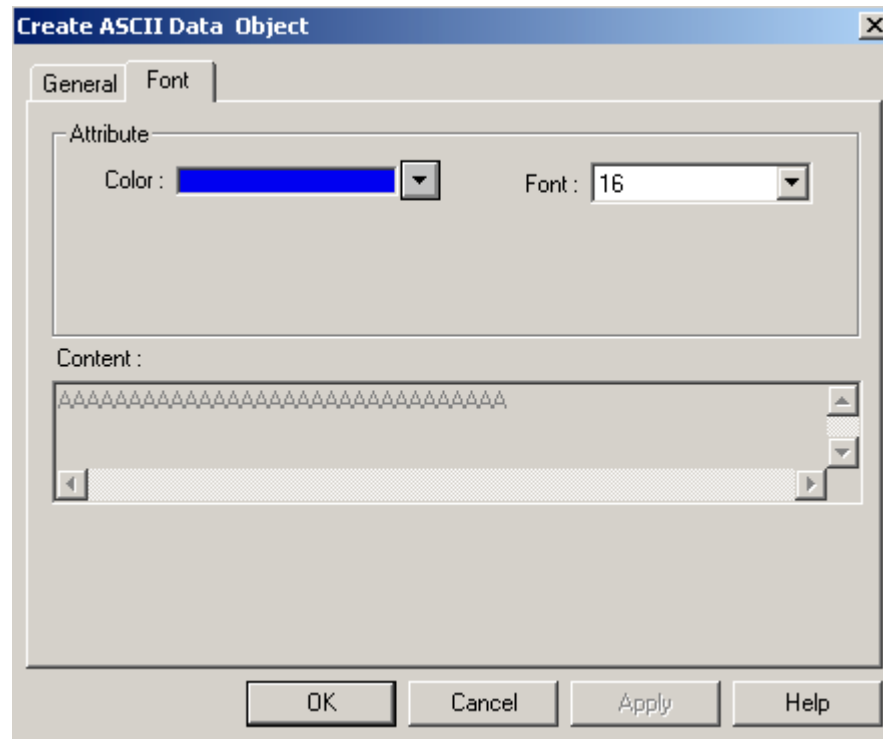
- Select the ASCII data display object



- Select Hold reg(4x) as the Device type.
- Enter 121 for the Device address. This is Modbus register 121 or %QW121 in the SMLC.
- Set the No. of words to 16 because this is the max length of a string in the HMI.

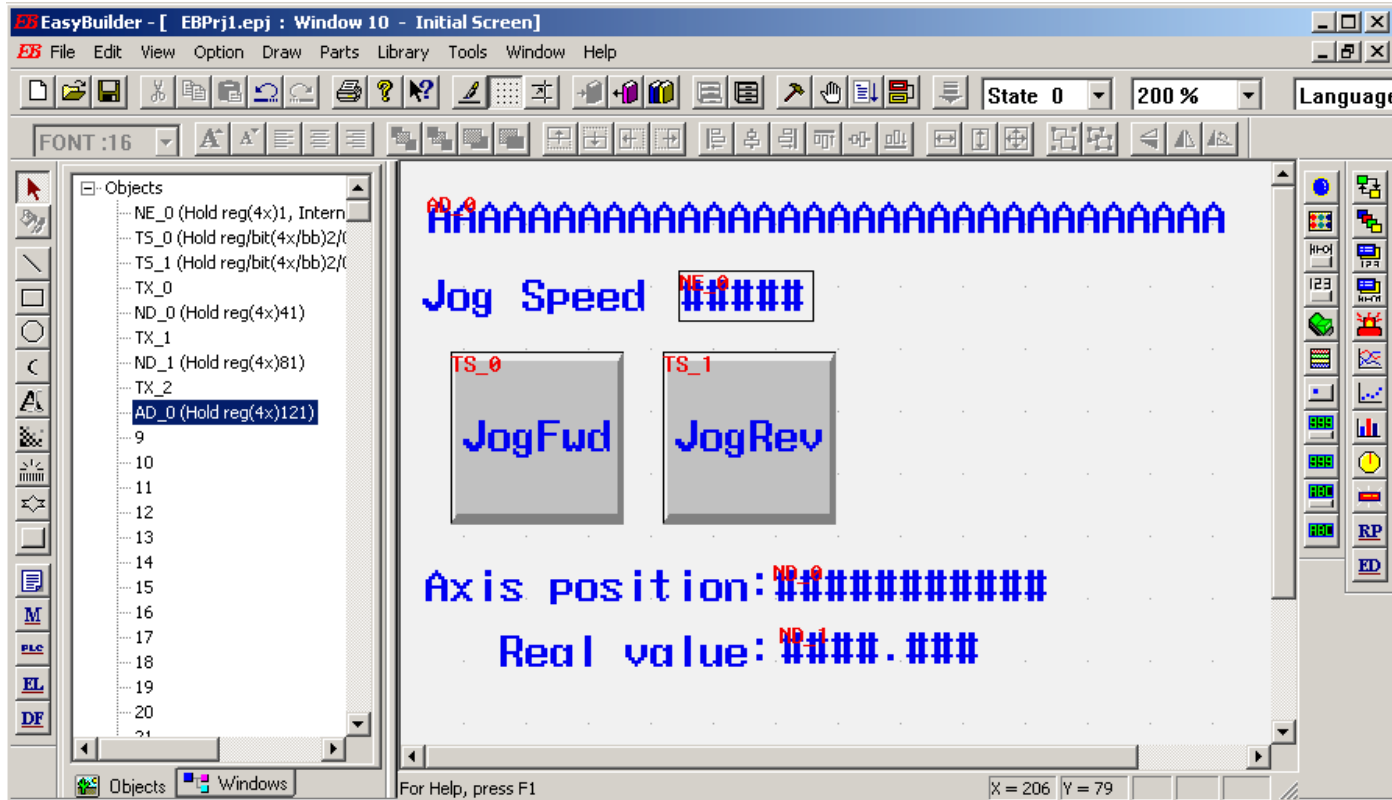
HMI - Adding a STRING display, part 2

- If you look at the Font tab you can see that there are 32 “placeholders” in the Content.



HMI - Adding a STRING display, part 3

- Place the text string at the top of the screen.
- The screen should now look like this:

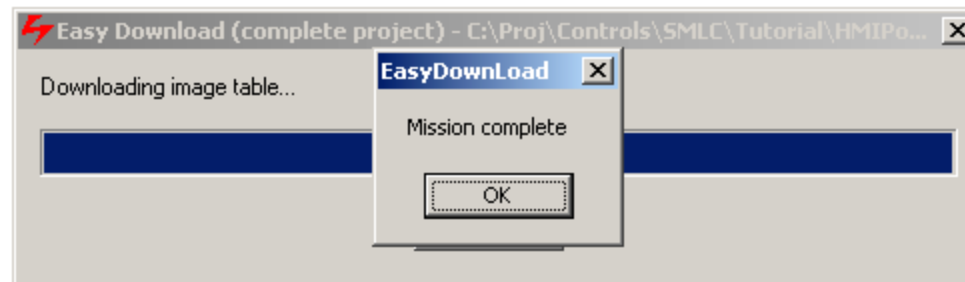


HMI - Downloading the project to the HMI

- Press the download button to send the project to the HMI

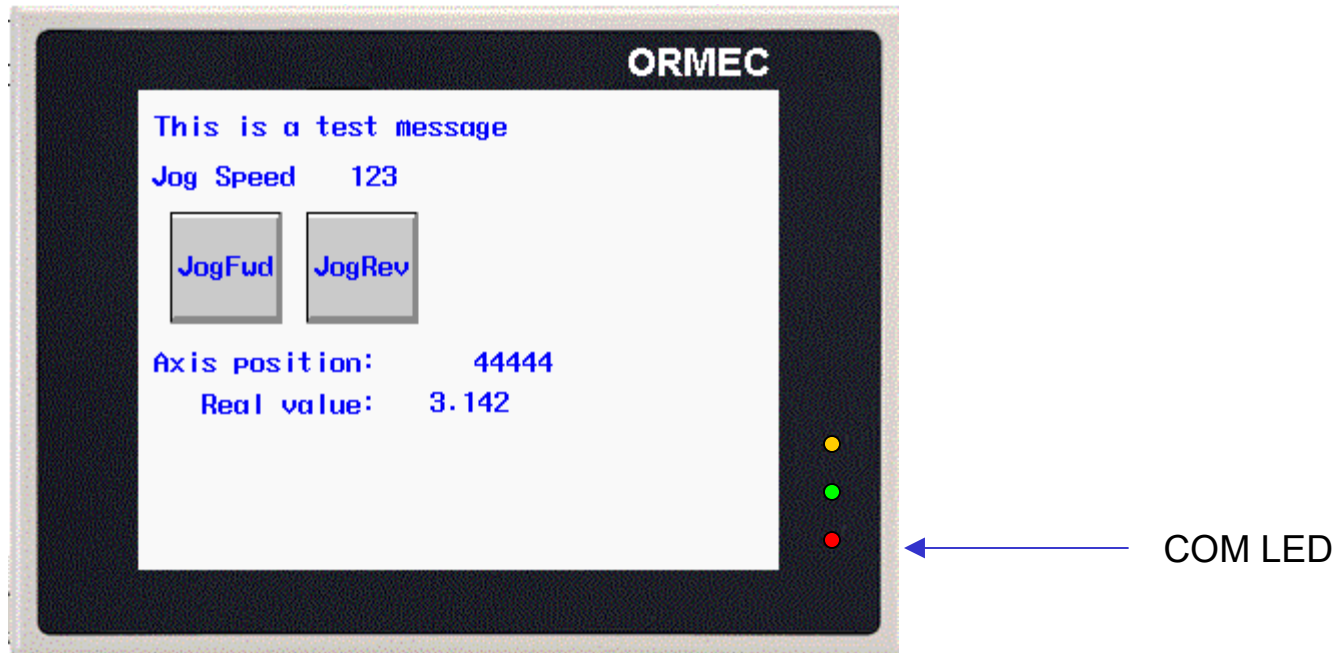


- While the program is downloading a progress bar is displayed
- When the download is finished the “Mission complete” dialog is displayed. The HMI will not display the first screen until you click OK.



HMI - Downloading the project to the HMI

- Press the download button to send the project to the HMI
- The screen should appear as below
- The red COM LED on the lower right indicates communication activity. If the HMI is successfully communicating with the SMLC the LED will appear solid or flashing rapidly. If there is a problem with the communications, the LED will flash briefly once every few seconds (based on what the timeout is set to).



HMI - Notes

- If you power cycle the SMLC with an Ethernet HMI connected to it you must power cycle the HMI as well. The HMI will not re-initiate communications until you do.

← COM LED