

Beijer HMI RMC200 Connection with Modbus

This document demonstrates how to establish a basic ethernet connection between a Beijer X2 pro 7 HMI and Delta RMC200. The user will learn how to send data to RMC registers from the Beijer HMI in the form of command data, read register data from the RMC with the HMI, and graph basic RMC plot data on the HMI.

Equipment used:

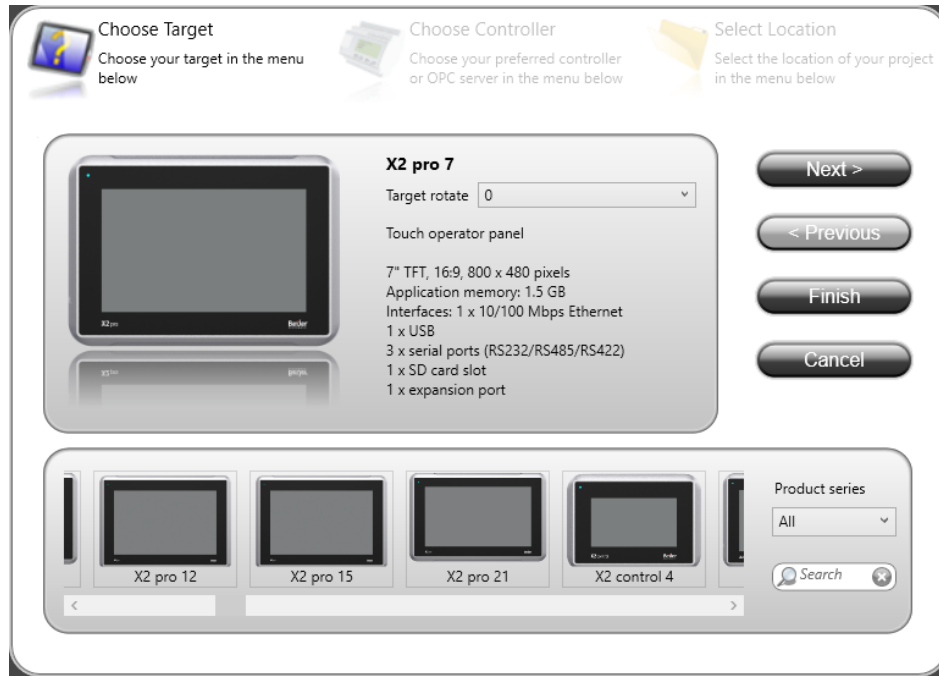
- Delta RMC200 Motion Controller
- Beijer X2 pro 7 HMI
- Ethernet switch
- PC connected to ethernet switch
- RMCTools Software
- Beijer iX Developer 2.40 Software

Contents

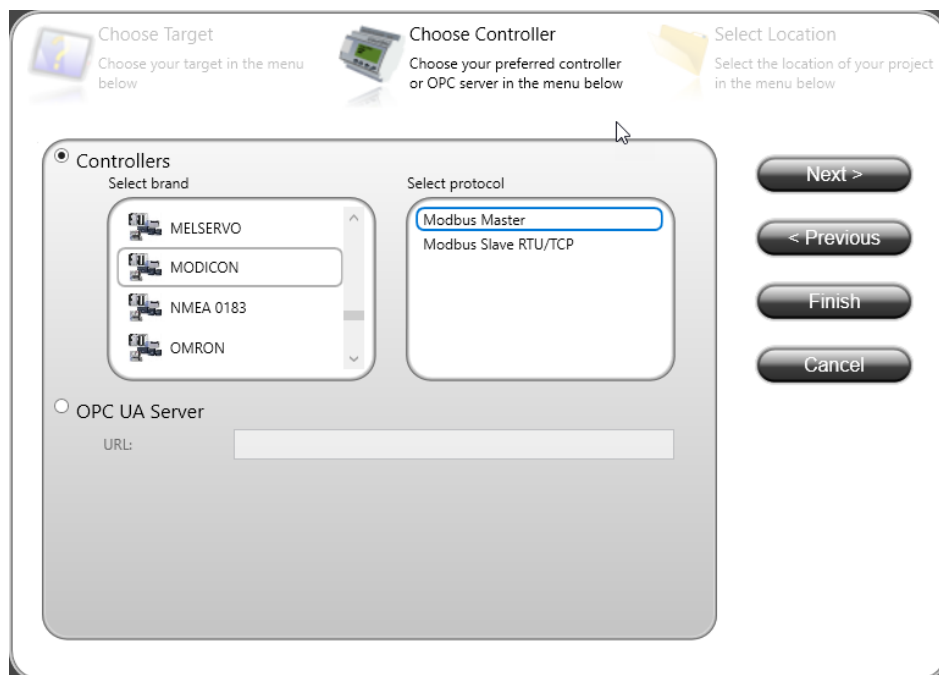
HMI Communication Settings	1
Setting the RMC IP Address	4
Reading From the RMC	5
Configuring Status Bits	7
Testing Communications for Reading	9
Writing to the RMC	12
Screen Configuration for Writes	13
Scripting for Writes	15
Testing Writes	16
Modbus Plot Capture	17
Testing Plot Capture.....	20

HMI Communication Settings

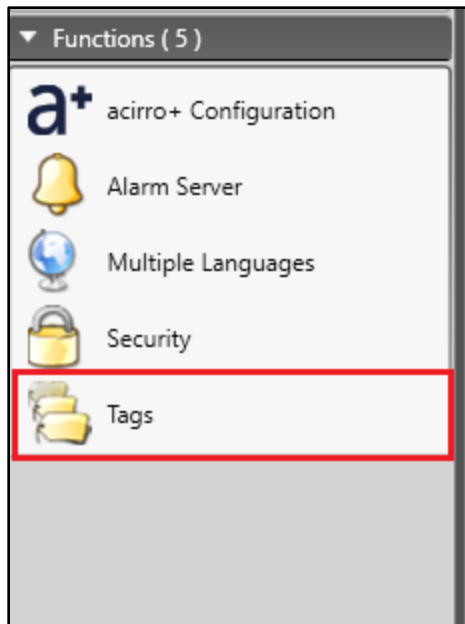
1. Connect the Delta RMC and Beijer HMI to an ethernet switch.
2. Open iX Developer 2.40. Create a **New Project**.
3. Choose a Target HMI (X2 Pro 7 in this example).



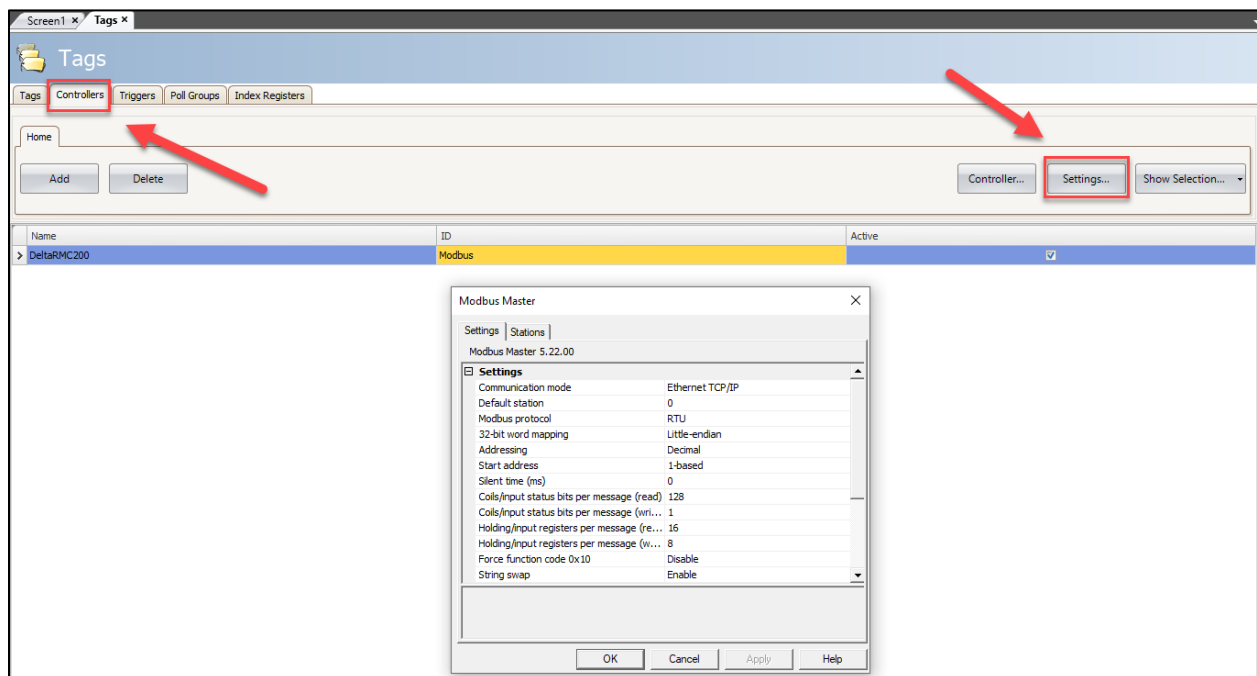
4. Choose the Controller (Modicon **Modbus Master** for this example) which the RMC will emulate.



5. Name the project and click **Finish**.
6. On the left side of the screen, in the **Functions** pane, click **Tags**.

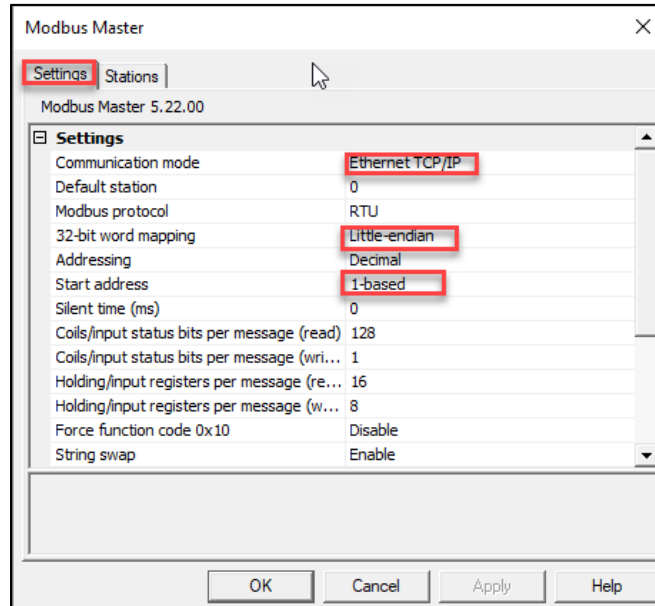


7. In the newly opened **Tags** panel, select the **Controllers** Ribbon.
8. With Controller1 selected, click **Settings** to open a dialog box.

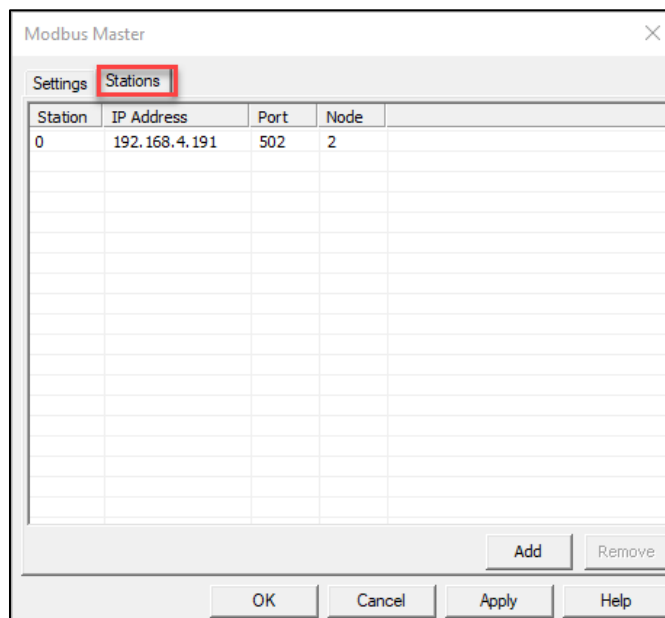


9. On the **Settings** tab, set the following:
 - a. **Communication Mode** to **Ethernet TCP/IP**
 - b. **32-bit Word Mapping** to **Little-endian**
 - c. **Start Address** to **1-based**

Make sure the rest of the options match the below images.



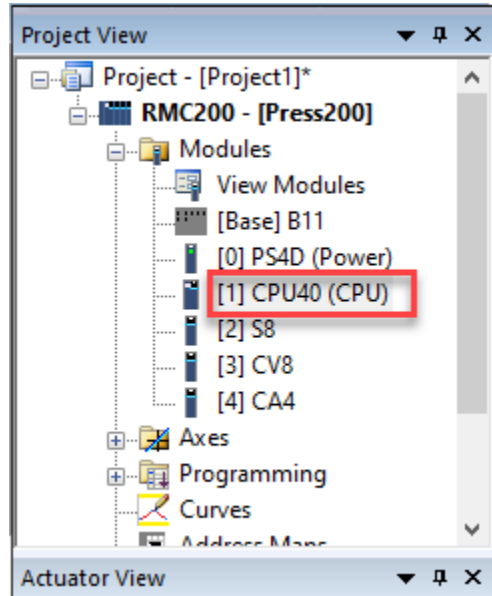
10. In the Modbus Master dialog box, on the **Stations** tab, set the **Station 0 IP Address** to the IP Address of the Delta RMC (Steps to find the RMC IP address are shown in *Setting the RMC IP Address* on pg. 4).



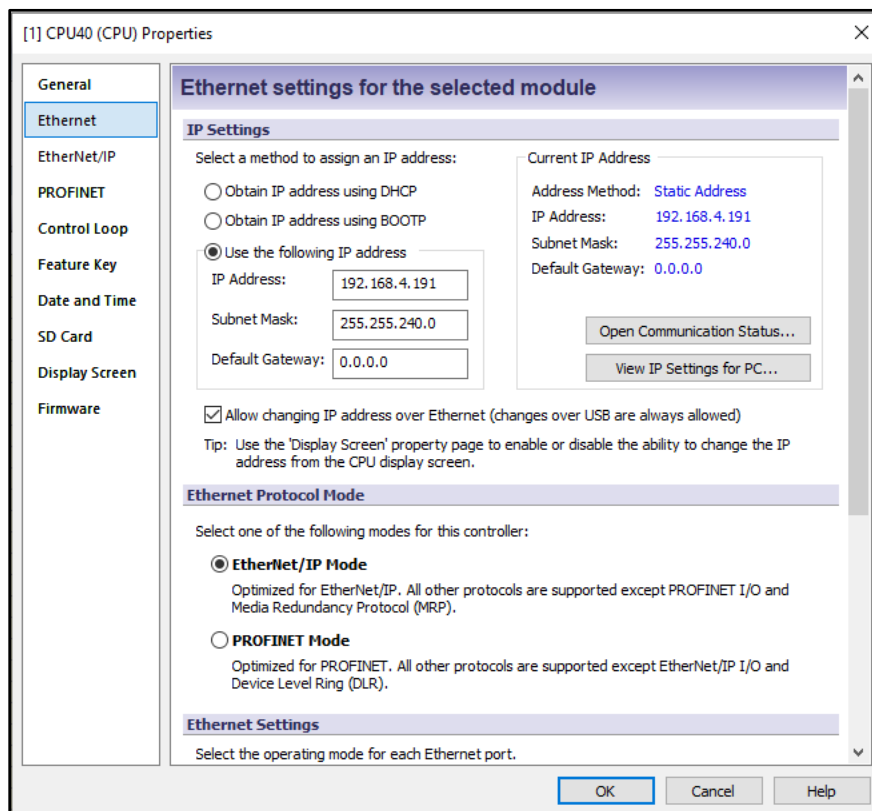
11. Click **OK**.
12. The Controller name can also be set, in this example we set the name to DeltaRMC200.

Setting the RMC IP Address

13. To find the IP Address of the RMC, connect the RMC to the ethernet switch and go online with it in RMCTools. On the left side of the screen, in the **Project View** pane, expand **Modules** and double-click the **CPU** module.



14. On the **Ethernet** page, set the RMC IP address settings.



Reading From the RMC

We will create HMI tags for reading from the RMC. The information we will be communicating are the actual position of Axis 0 along with the In Position and Halted status bits.

15. Return to the **Tags** panel. Click **Add** to add another Tag. Create a total of three Tags.
16. Rename the three Tags **PositionFeedback**, **InPosition**, and **Halted**.
17. Set the Controller Data Type for:
 - a. **InPosition** to **INT32**
 - b. **Halted** to **INT32**
 - c. **PositionFeedback** to **FLOAT**
18. Change the Access Right for each Tag to **Read**.

Screen1 x

Tags x

Tags

Tags

Controllers

Triggers

Poll Groups

Index Registers

Home

Add

Delete

Columns Visible

☒ Scaling

☐ Others

☐ Data Exchange


Filter

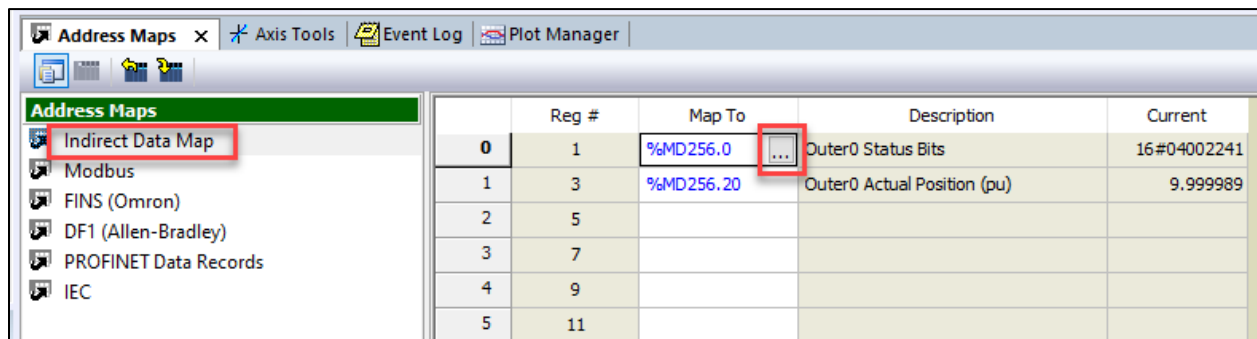
Tag			Controllers		Scaling				
Name	Data Type	Access Right	Data Type	DeltaRM...	Offset	Gain	Read Expression	Write Expression	
InPosition	DEFAULT	Read	INT32	400001	0	1			
Halted	DEFAULT	Read	INT32	400001	0	1			
PositionFeedback	DEFAULT	Read	FLOAT	400003	0	1			

19. To correctly assign Tags to addresses in the RMC use the **Indirect Data Map** in RMCTools.
20. In RMCTools, in the **Project View** pane, double-click **Address Maps** and choose the **Indirect Data Map**. Right click on the **Reg #** column, point to **Address Format** and choose **Modbus**.

	Reg #	Map To	Description	Current
0	1	RM00000	Output Status Bits	16#04002241
1	3		Actual Position (pu)	9.999959
2	5			
3	7			
4	9			
5	11			
6	13			
7	15			
8	17			
9	19			

21. The Indirect Data Map allows the user to map and write/read specific registers. The Register number is listed in the **Reg #** column. For this example, we will use Registers **1** and **3** (The actual internal register values are 400001 and 400003 but are truncated for readability).

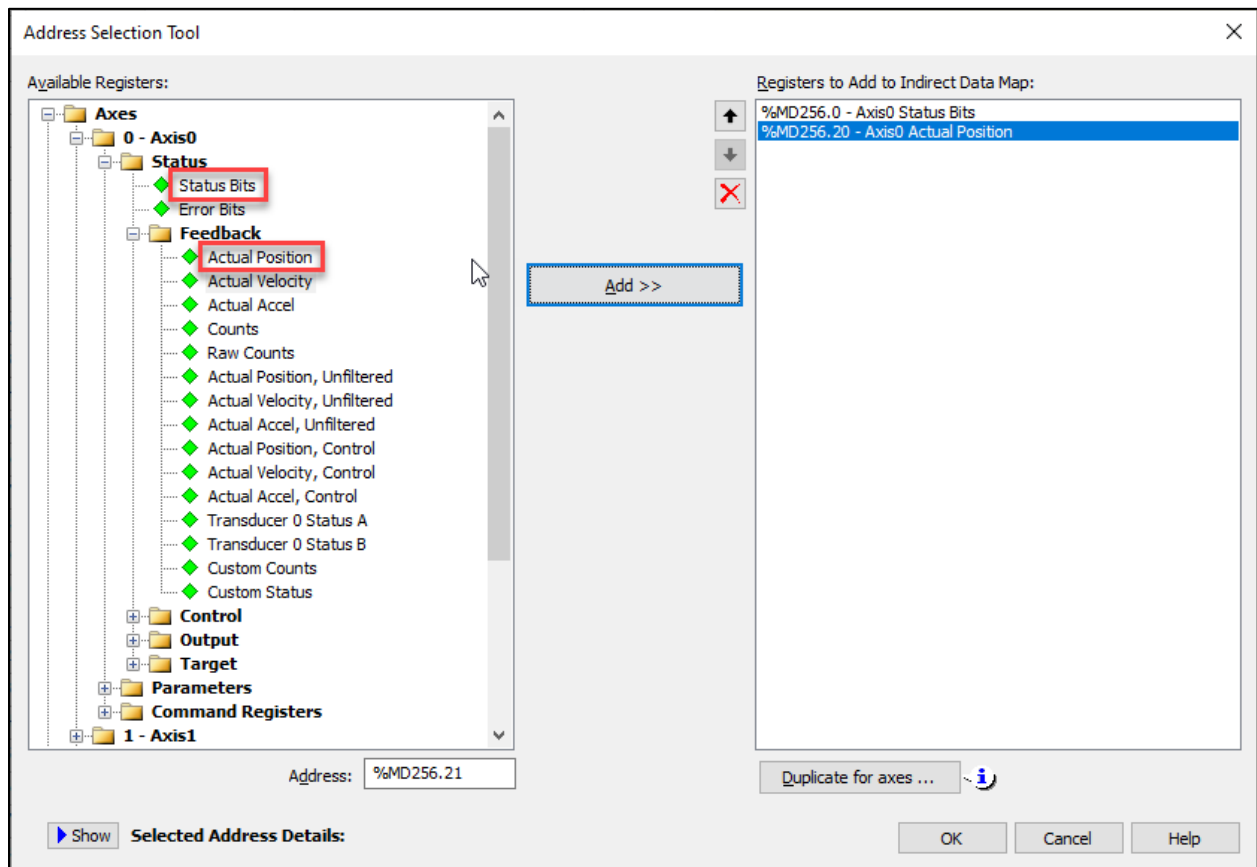
22. In the **Map To** box, click the  button to open the **Address Selection Tool** window.




23. The **Address Selection Tool** window allows the user to easily find and select the necessary register. In this case select:

- a. **Axis0 Status Bits**
- b. **Axis0 Actual Position**

Add both, click **OK**.



24. Click the **Download to Controller**  button on the **Address Maps** toolbar to download to the controller.

25. Return to iX Developer. On the **Tags** tab, set the register addresses for the **InPosition**, **Halted**, and **PositionFeedback** Tags to match the register addresses in the **Indirect Data Map**. In this case, **400001** and **400003**, as shown below.

Home

Add

Delete

Columns Visible

☒ Scaling
☐ Others
☐ Data Exchange

Filter

Tag	Controllers			Scaling				
Name	Data Type	Access Right	Data Type	DeltaRM...	Offset	Gain	Read Expression	Write Expression
InPosition	DEFAULT	Read	INT32	400001	0	1		
Halted	DEFAULT	Read	INT32	400001	0	1		
PositionFeedback	DEFAULT	Read	FLOAT	400003	0	1		

Configuring Status Bits

The **InPosition** and **Halted** Tags in their current configuration will print all status bits in hexadecimal. We will modify the Tags so that they mask and only print the **InPosition** and **Halted** status bits, respectively.

26. In the **Tags** pane, in the **Columns Visible** section, check **Scaling** to reveal more options.

Home

Add

Delete

Columns Visible

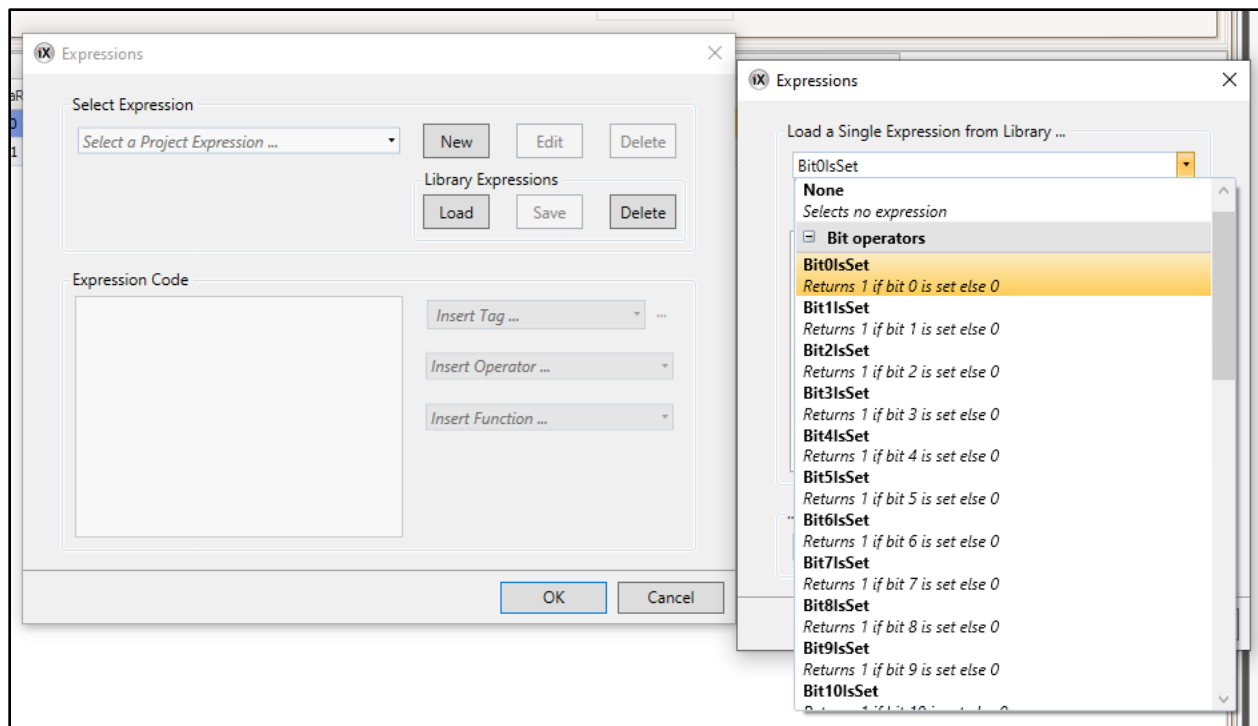
☒ Scaling
 ☐ Others

☐ Data Exchange

Filter

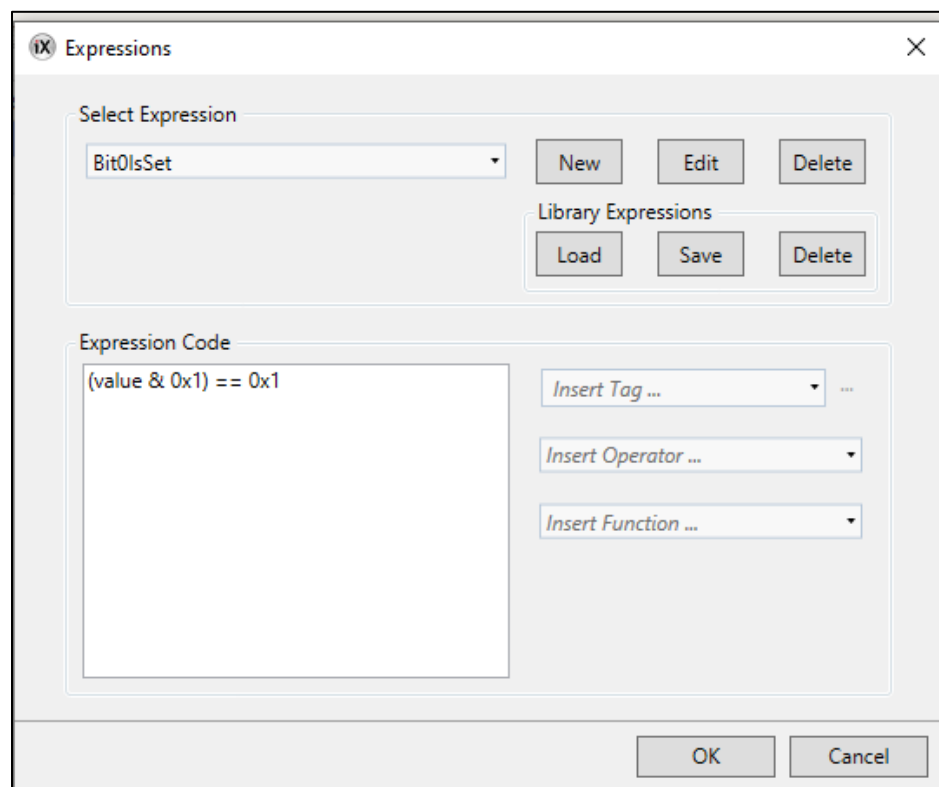
Tag				Controllers	Scaling			
Name	Data Type	Access Right	Data Type	DeltaRM...	Offset	Gain	Read Expression	Write Expression
InPosition	DEFAULT	Read	INT32	400001	0	1		
Halted	DEFAULT	Read	INT32	400001	0	1		
PositionFeedback	DEFAULT	Read	FLOAT	400003	0	1		

27. In the **InPosition** row, click the cell in the **Read Expression** column and click the **three-dot ellipsis**. A dialog box will appear.
28. Click **Load** and another dialog box will appear. From the drop-down list, expand **Bit Operators** and choose **Bit0IsSet**, then click **OK**.



29. Name the expression if prompted and click **OK**.

30. The **Expression** should look like the following.



31. Click **OK** to close the dialog box.

32. Repeat steps 24-28 for the **Halted** tag. Use **Bit15IsSet** for the expression.

33. The Tags should now look like this:

Home

Add

Delete


Columns Visible

☒ Scaling

☐ Others

☐ Data Exchange

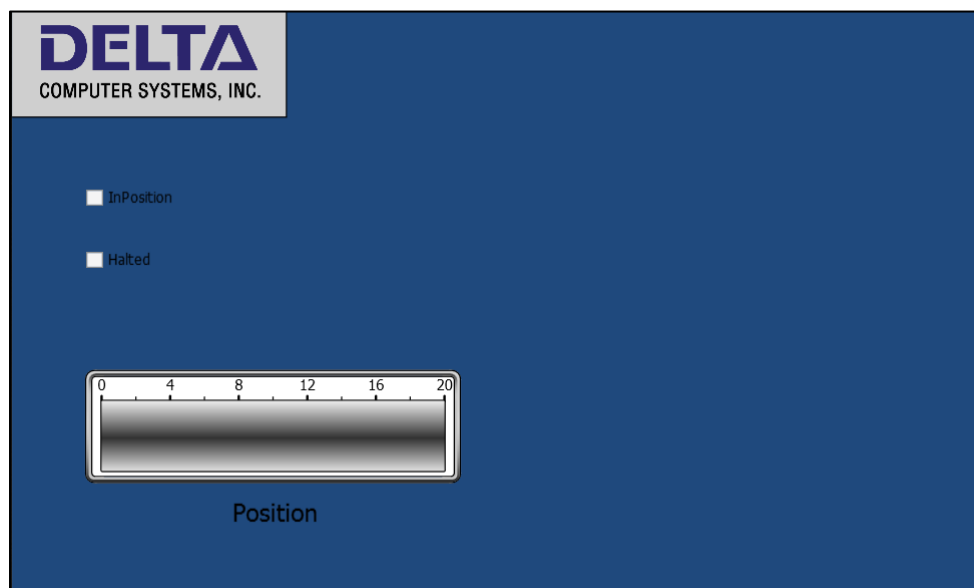
Filter

Tag			Controllers		Scaling			
Name	Data Type	Access Right	Data Type	DeltaRM...	Offset	Gain	Read Expression	Write Expression
InPosition	DEFAULT	Read	INT32	400001	0	1	Bit0IsSet	
 Halted	DEFAULT	Read	INT32	400001	0	1	Bit15IsSet ...	
PositionFeedback	DEFAULT	Read	FLOAT	400003	0	1		

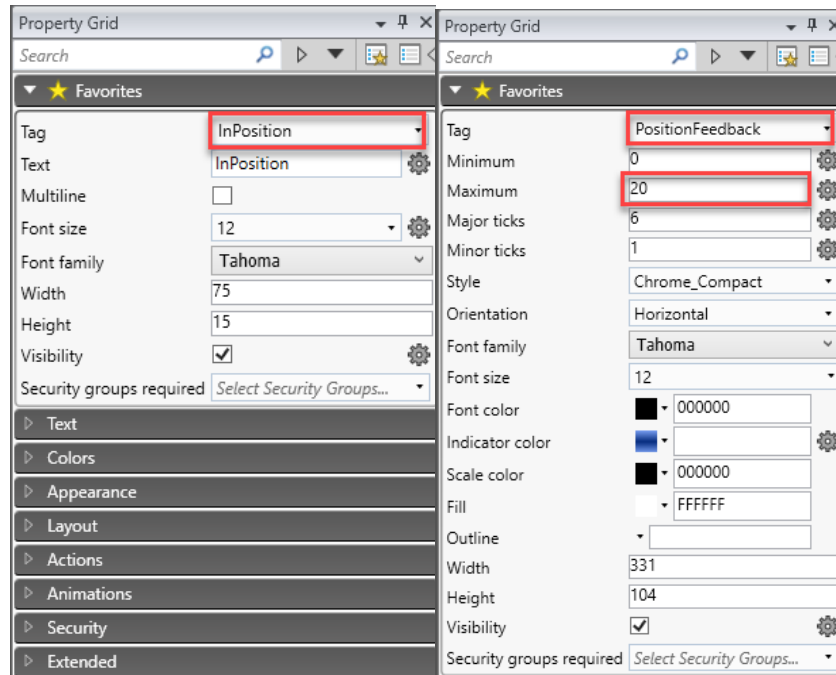
The connections for RMC reads from the HMI have now been set up. Next, we will connect our created Tags to objects on the screen.

Testing Communications for Reading

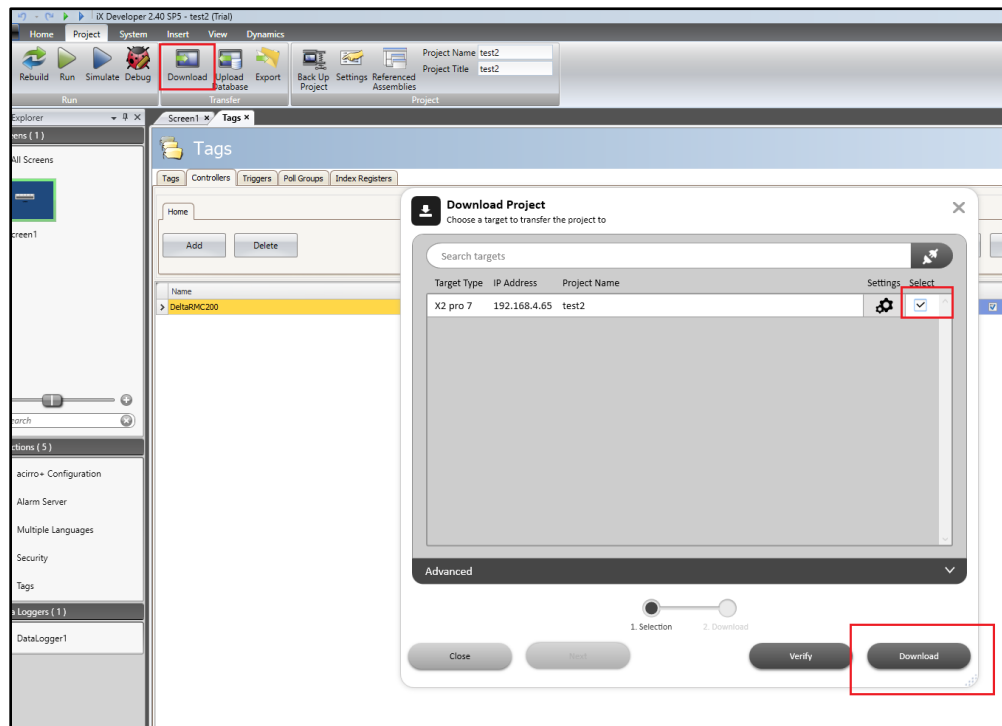
An example screen shown below has been created using **Check Boxes** and a **Linear Meter**. The Linear Meter has been set to have a minimum value of **0** and maximum value of **20** to match the minimum and maximum travel in RMCTools. The Check Boxes are unmodified.



34. Assign one **Check Box** to the **InPosition** Tag and the second **Check Box** to the **Halted** Tag by selecting each object and using the **Property Grid**. Assign the **Linear Meter** to the **PositionFeedback** Tag.



35. Now we can test the communications. In iX Developer, on the **Project** tab, click **Download**, select the target device in the dialog box, and click **Download**.



36. Test the RMC reads from the HMI by sending simple move commands to Axis0 in RMCTools. The Position meter should move in real-time and accurately represent the position of the Axis0 cylinder. The In-Position check box should be checked when the axis has finished moving and should be unchecked while it is moving. The Halted Check Box should remain unchecked unless an axis halt occurs.

After confirming that reading these two objects works, tags will now be created for writing to the RMC.

Writing to the RMC

Due to the Beijer HMI's constant polling, scripting must be used to create functional objects and connections for writing to the RMC. First, we will create tags and link them to the RMC Address Map for the writes to the RMC.

37. To determine the register addresses for these tags, return to the **Address Maps** in RMCTools.
38. Choose the **Modbus Address Map** and expand the **402049** register address for **Command Area Axis 0** as shown.

	Start Address	End Address	Registers	Map To	Description
0	400001	402048	1024	%MD8.0	Indirect Data Map, registers 0-1023
1	402049	402688	320	%MD16.0	Command Area, Axes 0-31
2	402049	402068	10	%MD16.0	Command Area, Axis 0 (Outer0)
3	402049	402050	1	%MD16.0	Outer0 Command
4	402051	402052	1	%MD16.1	Outer0 Command Parameter 1
5	402053	402054	1	%MD16.2	Outer0 Command Parameter 2
6	402055	402056	1	%MD16.3	Outer0 Command Parameter 3
7	402057	402058	1	%MD16.4	Outer0 Command Parameter 4
8	402059	402060	1	%MD16.5	Outer0 Command Parameter 5
9	402061	402062	1	%MD16.6	Outer0 Command Parameter 6
10	402063	402064	1	%MD16.7	Outer0 Command Parameter 7
11	402065	402066	1	%MD16.8	Outer0 Command Parameter 8
12	402067	402068	1	%MD16.9	Outer0 Command Parameter 9
13	402069	402088	10	%MD16.10	Command Area, Axis 1 (Outer1)

39. The Axis0 Command parameters are listed from 402049 to 402067. Each Command Parameter corresponds in order as shown in the Command Tool. For a Move Absolute (20), Axis0 Command (**403049**) is the **Command Number** (20), Axis0 Command Parameter 1 (**402051**) is the **Position** value, Command Parameter 2 (**402053**) is **Speed**, and so on (Direction, 402059, is not necessary for linear axes like in this example).

Command Tool - Press200

Axis0

Cmd: Move Absolute (20)

Position (pu): 15

Speed (pu/s): 5

Accel Rate (pu/s²): 5

Decel Rate (pu/s²): 5

Direction: Nearest (0)

Send

Axis1

Cmd: No-op (0)

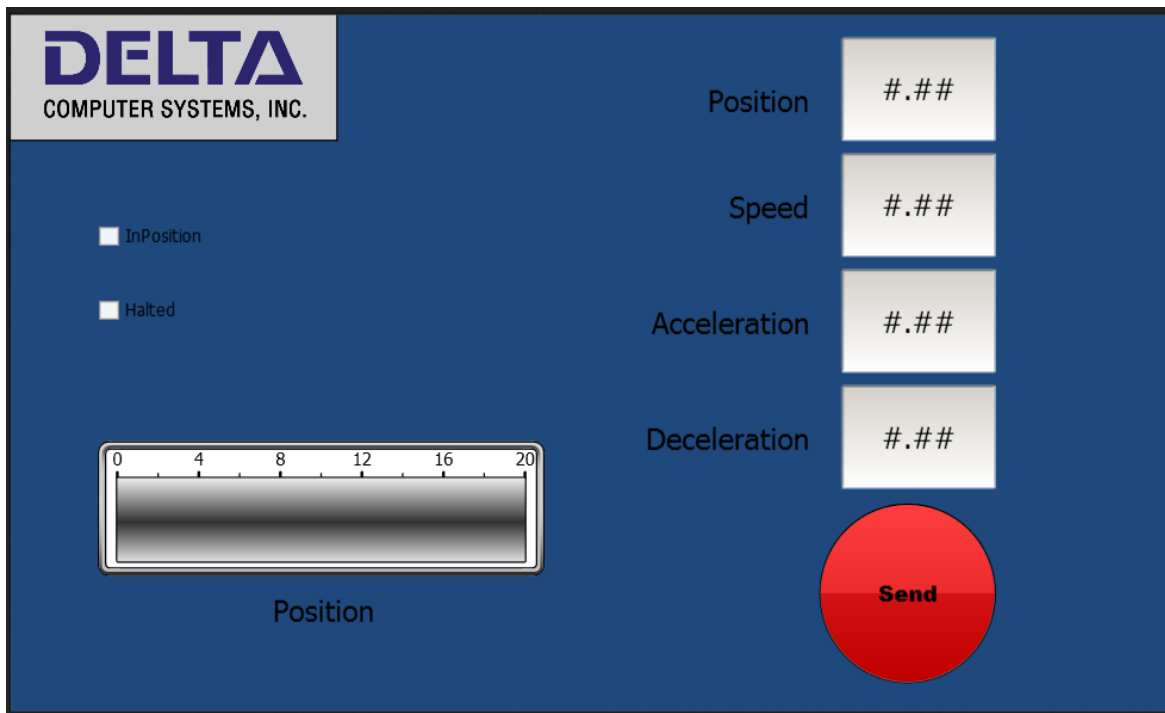
40. Return to iX Developer and the Tags pane. Create 9 more Tags and name them like in the image below. Set the **Access Right** for each to **Write** and set the **Data Type** for each to **FLOAT**.
41. Assign the register addresses from RMCTools to their corresponding tag in iX Developer. Use the below image for reference. Note that **Command** is assigned an address directly whereas the remaining RMC addresses are assigned to Tags named **Parameter 1 - 4**.

Tag			Controllers		Scaling			
Name	Data Type	Access Right	Data Type	DeltaRM...	Offset	Gain	Read Expression	Write Expression
InPosition	DEFAULT	Read	INT32	400001	0	1	Bit0IsSet	
Halted	DEFAULT	Read	INT32	400001	0	1	Bit15IsSet	
PositionFeedback	DEFAULT	Read	FLOAT	400003	0	1		
Position	DEFAULT	Write	FLOAT		0	1		
Speed	DEFAULT	Write	FLOAT		0	1		
Acceleration	DEFAULT	Write	FLOAT		0	1		
Deceleration	DEFAULT	Write	FLOAT		0	1		
Command	DEFAULT	Write	FLOAT	402049	0	1		
Parameter 1	DEFAULT	Write	FLOAT	402051	0	1		
Parameter 2	DEFAULT	Write	FLOAT	402053	0	1		
Parameter 3	DEFAULT	Write	FLOAT	402055	0	1		
Parameter 4	DEFAULT	Write	FLOAT	402057	0	1		

Screen Configuration for Writes

42. Return to the **Screen** pane in iX Developer.

New Objects are added to link to the newly created Tags. **Analog Numeric** objects are used for **Position**, **Speed**, **Acceleration**, and **Deceleration** Tags and a **Button** object is used to send a command.



43. Each **Analog Numeric** is linked to its respective **Tag** (Position Numeric linked to the **Position** Tag, Speed Numeric linked to the **Speed** Tag, etc). The max value for each of these **Analog Numeric** objects is adjusted in the **Property Grid** to prevent sending a value outside the realm of each parameter (ex. Max Speed = 25, Max Position = 20...). The **Display Format** is set to **Decimal** and the **Number of decimals** set to 2. The **Validate value on input** box is also checked to send an error message if a value out of range is attempted. A Button object to send a command is also added but does not need to be linked to a Tag.

Tag	Acceleration	Tag	Position
Display format	Decimal	Display format	Decimal
Limit number of characters	<input type="checkbox"/>	Limit number of characters	<input type="checkbox"/>
Number of characters	10	Number of characters	10
Number of decimals	2	Number of decimals	2
Prefix		Prefix	
Suffix		Suffix	
Read only	<input type="checkbox"/>	Read only	<input type="checkbox"/>
Validate value on input	<input checked="" type="checkbox"/>	Validate value on input	<input checked="" type="checkbox"/>
Validate value on display	<input type="checkbox"/>	Validate value on display	<input type="checkbox"/>
Lower limit	0	Lower limit	0
Upper limit	50	Upper limit	20
Font size	20	Font size	20
Font family	Tahoma	Font family	Tahoma
Font color	000000	Font color	000000
Fill		Fill	
Outline	7F9DB9	Outline	7F9DB9
Width	107	Width	107
Height	72	Height	72
Visibility	<input checked="" type="checkbox"/>	Visibility	<input checked="" type="checkbox"/>
Security groups required	Select Security Groups...	Security groups required	Select Security Groups...

Scripting for Writes

Scripting is incorporated to allow for reliable sending of data from the HMI to RMC. Without scripting, the Beijer HMI will be unable to send a value of "0" for any of its parameters.

44. While on **Screen1**, click the **Script** button at the bottom of the screen to pull up the script page for **Screen1**.



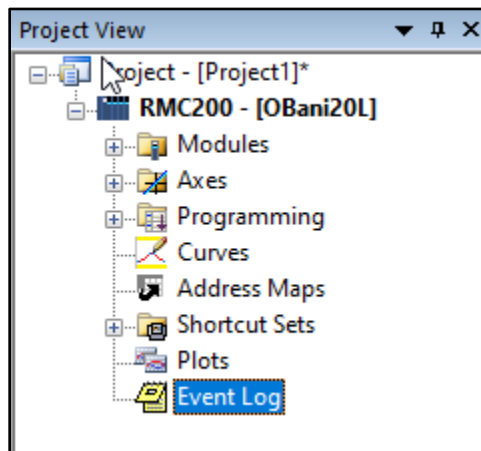
45. Add the following script shown in the red box.

```
1  //-----
2  // Press F1 to get help about using script.
3  // To access an object that is not located in the current class, start the call with Globals.
4  // When using events and timers be cautious not to generate memoryleaks,
5  // please see the help for more information.
6  //-----
7
8  namespace Neo.ApplicationFramework.Generated
9  {
10     using System.Windows.Forms;
11     using System;
12     using System.Drawing;
13     using Neo.ApplicationFramework.Tools;
14     using Neo.ApplicationFramework.Common.Graphics.Logic;
15     using Neo.ApplicationFramework.Controls;
16     using Neo.ApplicationFramework.Interfaces;
17
18
19     public partial class Screen1
20     {
21         void Button_Click(System.Object sender, System.EventArgs e)
22         {
23             float position = Globals.Tags.Position.Value;
24             float speed = Globals.Tags.Speed.Value;
25             float acceleration = Globals.Tags.Acceleration.Value;
26             float deceleration = Globals.Tags.Deceleration.Value;
27
28             Globals.Tags.Parameter1.SetAnalog(position);
29             Globals.Tags.Parameter2.SetAnalog(speed);
30             Globals.Tags.Parameter3.SetAnalog(acceleration);
31             Globals.Tags.Parameter4.SetAnalog(deceleration);
32
33             Globals.Tags.Command.SetAnalog(20);
34
35         }
36     }
37 }
38
```


The script makes it so, that upon clicking the Button object, the user selected values for Position, Speed, Acceleration, and Deceleration are assigned to their respective Tag which is then sent to the RMC. The final line of the script sends a value of 20 to the RMC Command register which executes a Move Absolute with the entered parameter values.

Testing Writes

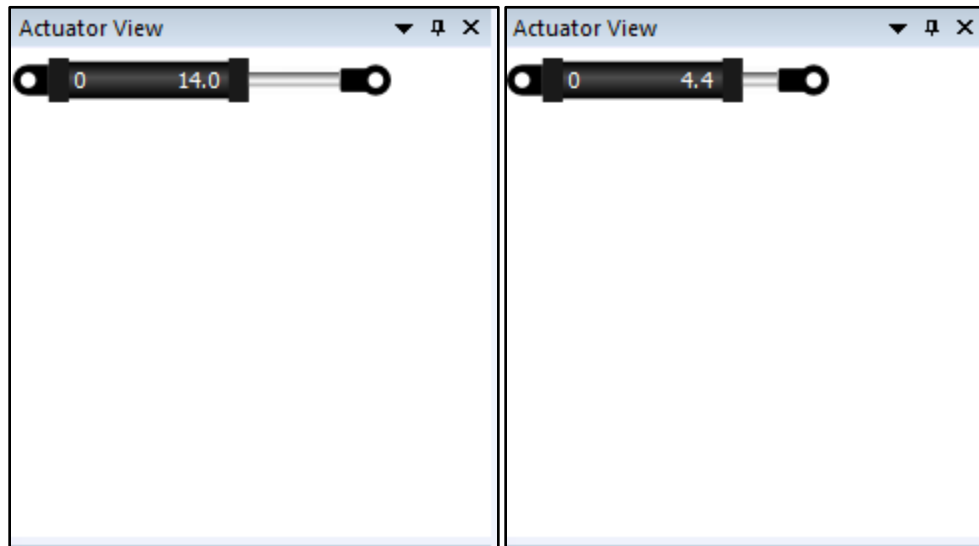
46. Download to the HMI and test the functionality. The **In-Position** box, **Halted** box, and **Position Meter** should still be accurate. To test the newly added Command Parameters, use the **Event Log** in RMCTools.
47. To access the **Event Log**, return to RMCTools. In the **Project View** pane, double-click the **Event Log** at the bottom of the project tree.



48. Upon sending a move from the HMI, the command and parameter values should appear in the Event Log in RMCTools as shown.

Time	Event	Details
1d 18:00:31.603	Event Log last updated	
1d 18:00:26.263	Command received.	Source: Intern
	Move Absolute (20) [Axis0]	
	Position (pu):	15
	Speed (pu/s):	10
	Accel Rate (pu/s ²):	17
	Decel Rate (pu/s ²):	16
	Direction:	Nearest (0)
1d 18:00:13.993	Command received.	Source: Intern
	Move Absolute (20) [Axis0]	
	Position (pu):	0
	Speed (pu/s):	10
	Accel Rate (pu/s ²):	25
	Decel Rate (pu/s ²):	25
	Direction:	Nearest (0)
1d 17:59:57.102	Command received.	Source: Intern
	Move Absolute (20) [Axis0]	
	Position (pu):	10
	Speed (pu/s):	10
	Accel Rate (pu/s ²):	25
	Decel Rate (pu/s ²):	25
	Direction:	Nearest (0)

49. The **Actuator View** can also be used to help test in RMCTools. The Actuator View can be accessed under **View -> Actuator View**.



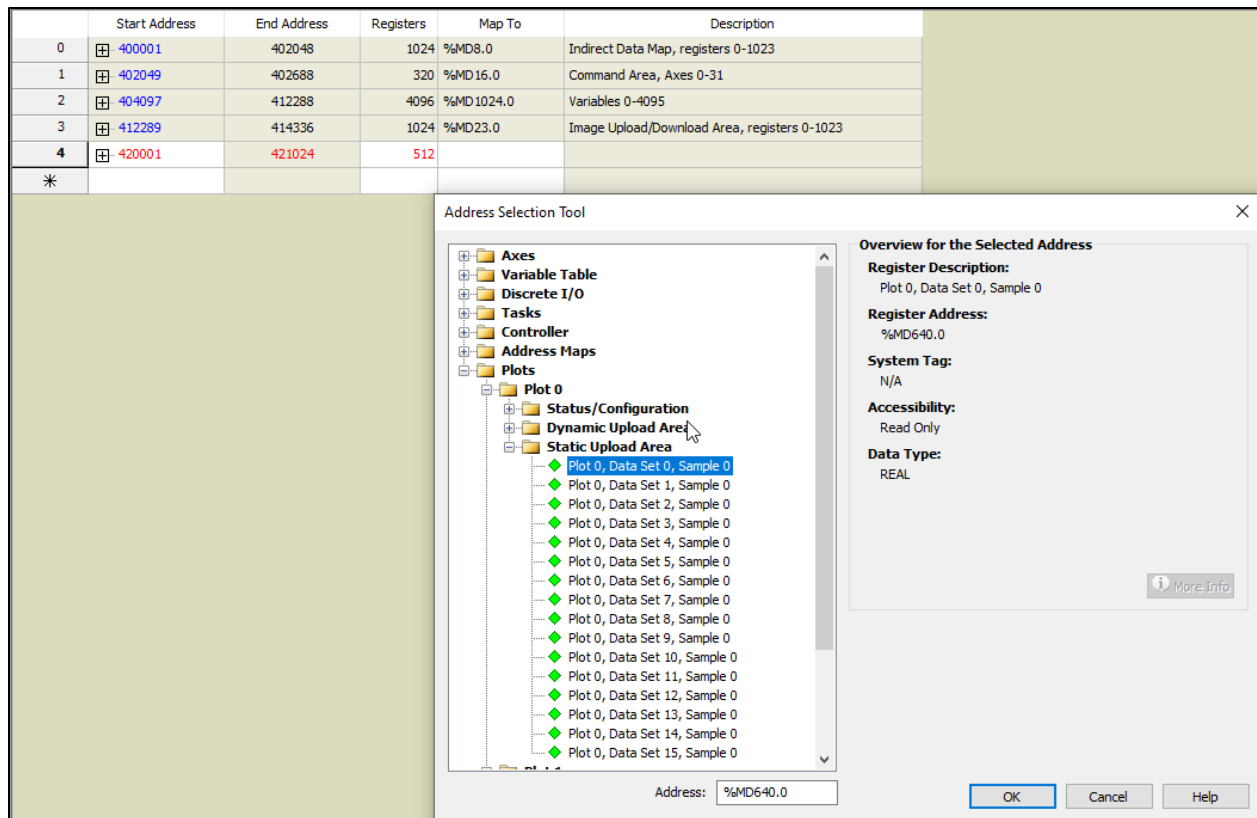
Modbus Plot Capture

The Beijer X2 Pro 7 can display short plots by reading the RMC plot data registers. With Modbus the plots can be up to 512 data points in length.

50. In RMCTools, open the **Address Maps** and select **Modbus**. Add a Configurable Address with a start address of **420001** and **512** registers.

	Start Address	End Address	Registers	Map To	Description
0	400001	402048	1024	%MD8.0	Indirect Data Map, registers 0-1023
1	402049	402688	320	%MD16.0	Command Area, Axes 0-31
2	404097	412288	4096	%MD1024.0	Variables 0-4095
3	412289	414336	1024	%MD23.0	Image Upload/Download Area, registers 0-1023
4	420001	421024	512	%MD640.0	
*					

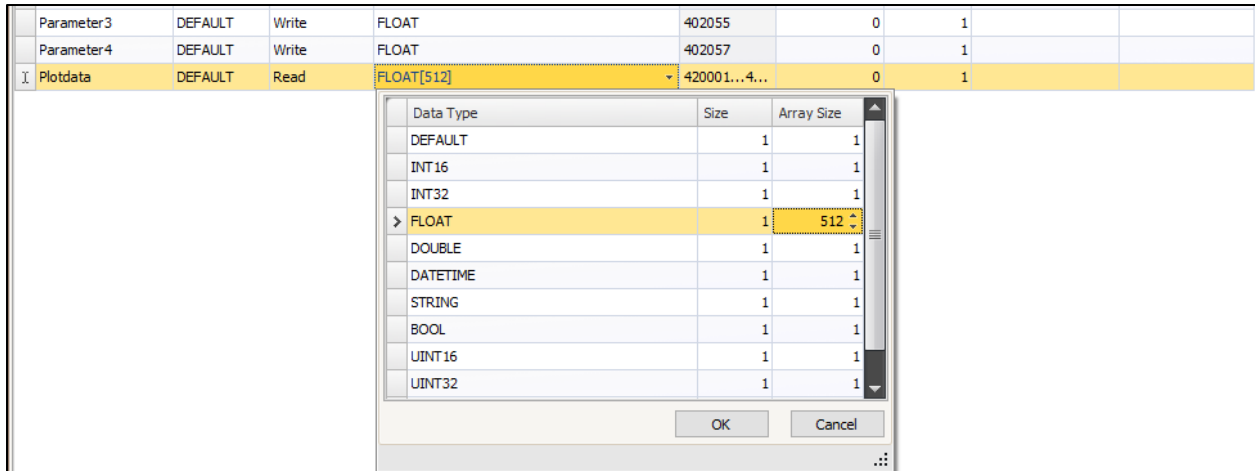
51. To select the register address to map to, click **...** in the respective **Map To** box. Browse to Plots -> Plot 0 -> Static Upload Area and choose **Data Set 0 (%MD640.0 for the RMC200)**.



52. The Configurable Address should now look the same as the image below.

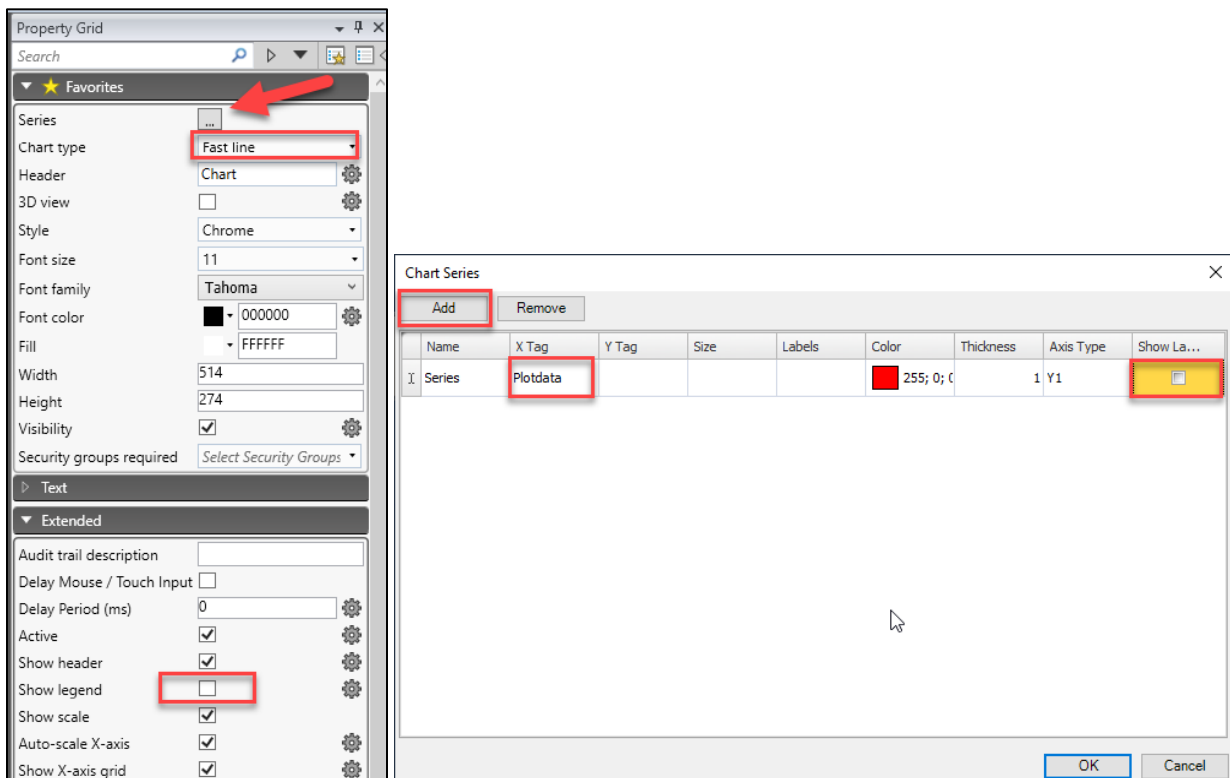
	Start Address	End Address	Registers	Map To	Description
0	400001	402048	1024	%MD8.0	Indirect Data Map, registers 0-1023
1	402049	402688	320	%MD16.0	Command Area, Axes 0-31
2	404097	412288	4096	%MD1024.0	Variables 0-4095
3	412289	414336	1024	%MD23.0	Image Upload/Download Area, registers 0-1023
4	420001	421024	512	%MD640.0	Plot 0 Static Upload Area (Data Set 0), registers 0-511
*					

53. Return to iX Developer and open the **Tags** pane. Create a new Tag and set the following parameters:
- Name the tag **Plotdata**
 - Set the **Read Access** to **Read**
 - Set the **Data Type** to **FLOAT**
 - Set the FLOAT variable to an array by increasing the **Array Size** to 512 in the pop-up box as seen below



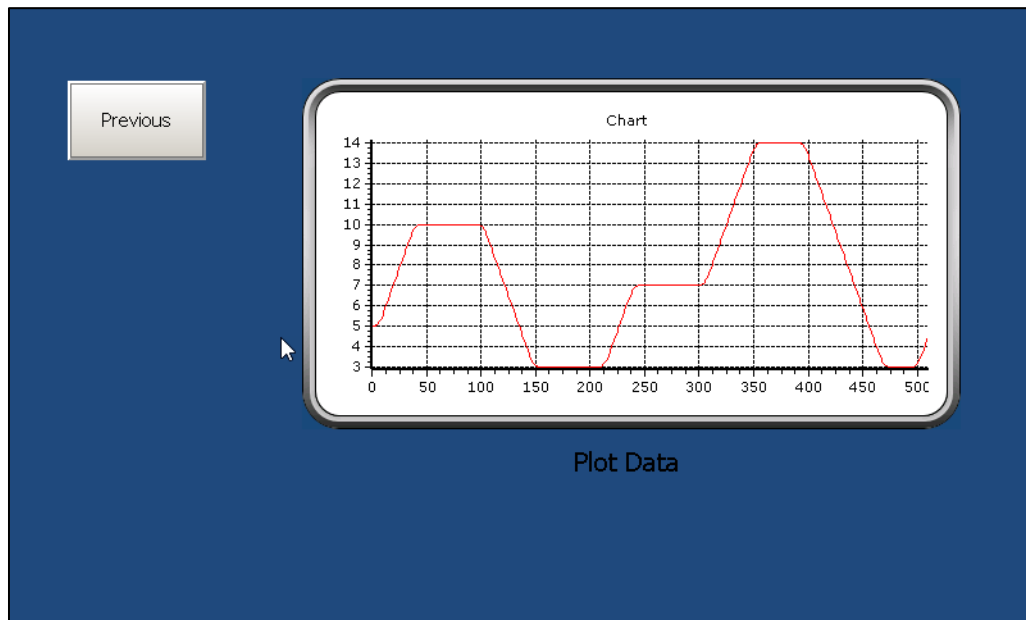
54. Create a second screen in iX Developer and add a **Chart** object. Change the **Chart Type** to **Fast line**. Expand the **Extended** drop down and uncheck **Show legend** to make the plot easier to read.

55. Click the three-dot ellipsis next to **Series** to open a dialog box. Click **Add** to add a new data series and set the **X Tag** to the **Plotdata** array. Uncheck the **Show Labels** box to make the graph easier to read.

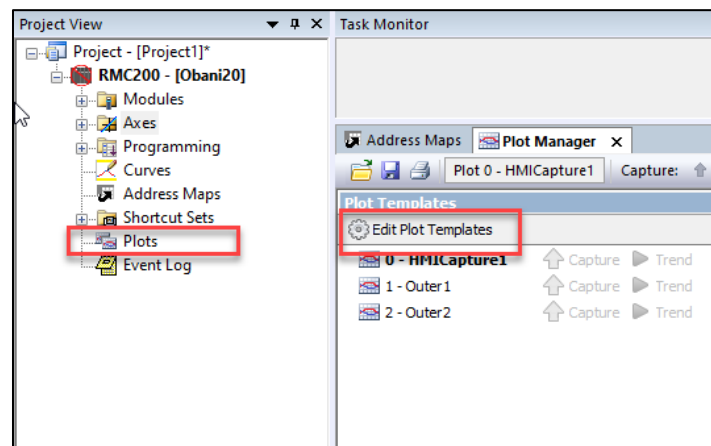


Testing Plot Capture

Buttons were added on both Screen 1 and Screen 2 to allow the user to change between screens.



- 56. Download and run the project. The Chart will show plot data and will update real-time. Like the Capture button in RMCTools, it will not overwrite after the chart is filled until new position data is received.
- 57. The sample rate of the plot in the RMC can be reduced to increase the plot duration at the expense of resolution.
- 58. To change the sample rate return to RMCTools and open the **Plot Manager**. In the **Plot Templates** pane, click the **Edit Plot Templates** button to open the Plot Template Editor.



- 59. In the Plot Template Editor, the **Sample Interval** controls the sample rate. Reducing the sample rate from 0.001 sec to 0.05 sec will multiply the plot length by 50.

Plot Template Editor

Plot Templates

Add
Up
Down
Delete

0 - HMICapture1

Plot Template 0 - HMICapture1

Plot Name: HMICapture1

Sample Interval: 0.05 sec

Capture Duration: 5 sec

Trend Duration: 30 sec

☒ Default Plot for Axis: 0 - Axis0

RMCTools will manage which registers are plotted.

☐ Custom Plot

The user will manage this plot's settings.

Plotted Data Items

Register/Value	Pen Color
Axis0 Target Position	
Axis0 Actual Position	
Axis0 Target Velocity	
Axis0 Actual Velocity	
Axis0 Control Output	
Axis0 Status Bits	
Axis0 Position Mean Squared Err	

Trigger Settings

Automatic Trigger: Motion Commands issued to Axis 0

Rearm Mode: Automatic

Pre-Trigger Percent: 0%

Plot Templates Used: 1 of 8
Template Memory Used: < 0.1%
Total Data Items Used: 5 of 128

OK
Cancel
Help