Tutorial: Integrating Ignition with RMC Motion Controllers

Introduction

The popular Ignition SCADA/MES software development platform by Inductive Automation may be integrated with an RMC motion controller manufactured by Delta Computer Systems using Modbus TCP. Data may be exchanged bi-directionally to both control and monitor an RMC from an Ignition project. Tag history can be recorded to a database and charted within Ignition projects. Alarms can be configured for any tag. The licensing for Ignition allows use of full functionality on a trial basis for free, the only limitation being that the service needs to be reactivated after two hours.

This guide assumes that RMCTools is installed and connected to an RMC200 controller. Instructions for configuring Modbus registers in RMCTools are described as preparation for connecting to Ignition.

Preparing your RMCTools Project

Finding desired Modbus TCP addresses

RMC controllers ship with many register addresses pre-configured. A number of addressing formats are offered, one of which is Modbus. The default addressing format conforms to the IEC-61131 standard, but display in alternate formats is offered.

In an RMCTools project, Modbus addresses may be viewed in the **Address Maps** tool. After opening this tool, select **Modbus** in the Address Maps list. For an RMC200, four ranges of addresses are preconfigured. These provide access to the Indirect Data Map, the command area, variables, and the image upload/download area.

Some of the addresses are multi-level having addresses listed in a tree format, with ranges expandable using the '+' and '-' icons. For example, to find the registers for the Axis 0 command area, click the '+' icon next to 402049 to expand the address map and view the addresses for the Axis0 Command register and the nine Axis0 Command Parameter registers in the range 402049 through 402068.

Setting up the Indirect Data Map

As described in the RMCTools help, "the Indirect Data Map maps data registers from anywhere in the RMC to a single area. This allows scattered data items to be consolidated for efficient communications."
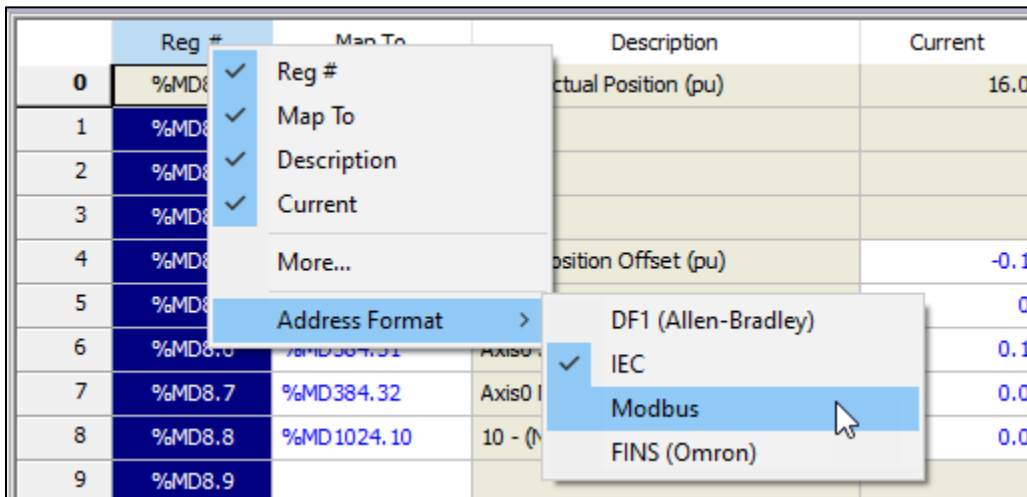
Initially the Indirect Data Map is empty, allowing each project to hold a custom configuration of desired addresses from various areas of the RMC controller. Use of this tool is optional but may be required depending on the registers that you may wish to access. For example, we will access the current position of an axis in this tutorial and will configure the Indirect Data Map to do this.

To set up the Indirect Data Map:

1. Open the Address Maps editor and select the Indirect Data Map.
2. In the **Map To** column of an Indirect Data Map row, click the ellipsis ⬚ button to browse to the RMC register you want to map. You can map as many of the Indirect Data Map entries as you would like.
3. Click the Download button 📥 to download the changes to the RMC.
4. Now, with your PLC or other host controller, you can read or write your data directly from the Indirect Data registers. Notice that if a mapped register is read-only, you cannot write to it.

The addressing display for the Indirect Data Map within RMCTools may be changed to display addresses in Modbus format instead of the default IEC format. To change this display, in the Address Maps, right-click a cell in the **Reg #** column and use the menu to select **Address Format -> Modbus**.


With desired addresses identified in the Modbus area or configured in the Indirect Data Map, you are prepared to reference them in Ignition.

In this example, we will start by mapping the current position for Axis 0 to the Indirect Data Map and then reading the position data with Ignition.

To configure the current position for Axis 0, click the cell in the **Map To** column for the first line, then click the ellipsis button that appears. This will display the **Address Selection Tool** dialog box.
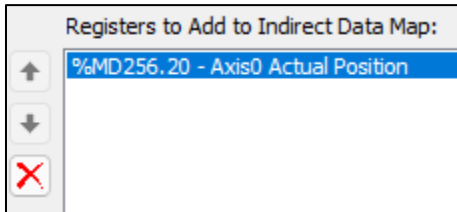
Click the '+' icons to open the tree to the Actual Position for Axis 0.
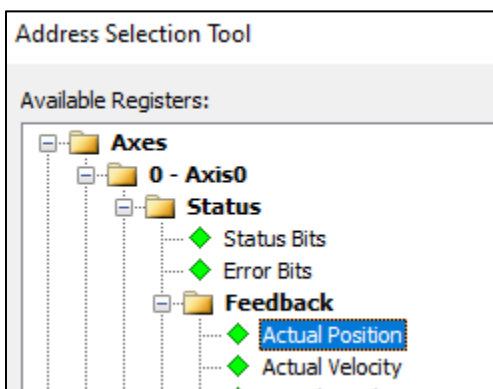
Below the tree, the value for this register is displayed: %MD256.20

Address: %MD256.20

Click the **Add >>** button to add this to the Indirect Data Map.

Registers to Add to Indirect Data Map:

%MD256.20 - Axis0 Actual Position

Click the **OK** button to save the changes.

Now the Indirect Data Map shows this register on line 0.

	Reg #	Map To	Description	Current
0	%MD8.0	%MD256.20	Axis0 Actual Position (pu)	N/A

Switching the address format to Modbus, the Reg # is shown as Modbus address 1.

	Reg #	Map To	Description	Current
0	1	%MD256.20	Axis0 Actual Position (pu)	N/A

Notice the prefix of digit 4 for Modbus is not shown. Now Modbus address 400001 is configured to communicate the actual position of Axis 0.

To save changes, ensure that the RMC is online. Use **Go Online** in the toolbar.

Offline

Go Online
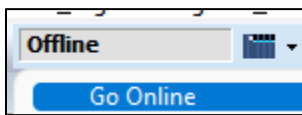
Click the Download button in the **Address Maps** area to download the changes to the RMC.

Installing Ignition

Downloading Ignition

The Ignition software may be downloaded at Inductive Automation's download site (https://inductiveautomation.com/downloads/). Installers for Windows, MacOS and Linux are offered.

Installing Ignition

This guide will address installing and configuring for Windows. After downloading the Ignition installer, launch the executable (ex. ignition-8.1.15-windows-64-installer.exe).

Specify an installation directory on the computer and assign a name for the Ignition service (a default is provided).

Select an installation option. For integration using Modbus, the **Typical** option is sufficient.

Now click the **Install** button.

Once the installer finishes, you may indicate whether to start the Ignition service.

Click **Finish**.

If you indicated to start Ignition, your default browser will open to a page prompting for the type of installation to establish. For a commercial usage scenario, accept the default initially highlighted by

clicking the **Ignition** button. Click to accept the terms and conditions. Next, provide your desired first (administrator) user account and password. Finally, review the TCP/IP port numbers that will be used, and change them here if you desire different ports. Click **Finish Setup**.

Ignition administration is performed in what is known as the Ignition Gateway. To start Ignition immediately, click **Start Gateway**. An opportunity to establish a demonstration "quick start" scenario is offered; this is optional.

For the purposes of this tutorial, select **Yes, Enable Quick Start**, as this will provision a sample database we can use for recording history.
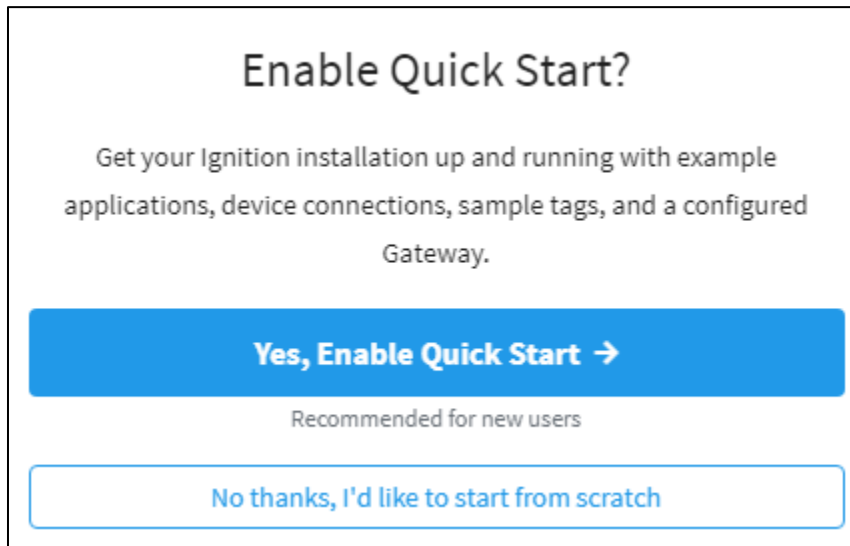
Confirm the administrator credentials and log in.

The Quick Start environment will be provisioned. You will now observe the Ignition Gateway administrative environment in your browser.

Preparing the Ignition Gateway

Adding a Modbus TCP device

For Ignition to communicate with an RMC using Modbus TCP, a "device" must be configured in the Ignition Gateway, and Modbus addresses configured for the device. With this completed, tag data can be exchanged bidirectionally (depending on the register, as some are read-only).

If you've followed the steps above, the Gateway administration screen will be visible. To launch this in the future, use the address https://hostname:8088/web/home (substitute a different port number if you configured one).

To add a device in the Gateway, click the **Config** icon in the menu at the left side of the Gateway administration screen in your browser.
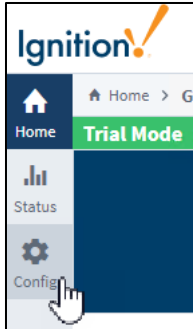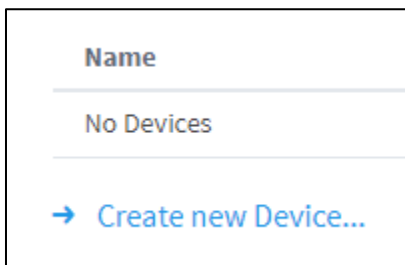
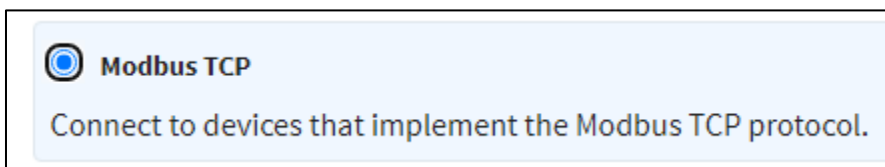Log in with your administrator credential established during setup.

In the menu at the left, scroll down to the OPC UA section and click **Device Connections**.
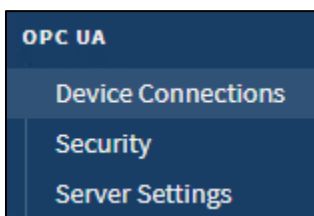
Click **Create new Device.**

Select **Modbus TCP** then scroll to the bottom and click Next.

Enter a **Name** for the Device and specify a **Hostname**. Other fields may be left alone.

Next you will configure an advanced property.

Setting advanced properties for Modbus TCP device

Click the check box labeled **Show advanced properties**. An option here is required for communicating with an RMC controller.

Click the check box for **Reverse Word Order**.

Leave the check box for **Zero-based addressing** unchecked. This will enable you to refer directly to Modbus address in RMCTools as you configure Ignition.

You may keep the defaults for other advanced settings.

Scroll to the bottom of the page and click **Create New Device**.

The new device will now appear in the device list. After a few moments, the Status should indicate "Connected". Ignition is now ready to communicate with the RMC but no addresses have been configured, so no data is exchanged.
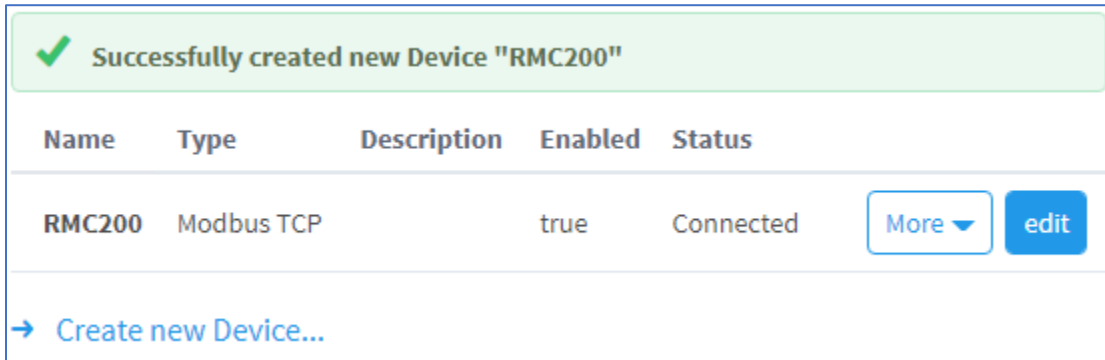
Next you will configure Modbus addresses for communicating with the RMC controller.

Adding Modbus TCP addresses

To configure some addresses from the RMC, click the **More** button for the device and select **Addresses**.
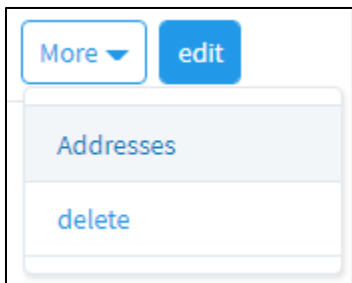
Now the Address Configuration screen will appear. Here you will add rows of addresses from the RMC.

Click **Add Row**. You may enter a single address or a range of addresses in one row.

To send commands an RMC, you will need to reference one or more Command Areas as seen in RMCTools. From Ignition you may write to these registers to issue a command to the RMC.

To add a range of Modbus addresses for the Command Area for Axis0 of your RMC, observe the list of addresses in the Address Map, Modbus area. Each register, being of type DWORD, consumes two addresses, for example 402049 and 402050 for Axis0 Command.

1	⊟ 402049	404088	1020	%MD16.0	Command Area, Axes 0-101
2	⊟ 402049	402068	10	%MD16.0	Command Area, Axis 0 (Axis0)
3	402049	402050	1	%MD16.0	Axis0 Command
4	402051	402052	1	%MD16.1	Axis0 Command Parameter 1
5	402053	402054	1	%MD16.2	Axis0 Command Parameter 2
6	402055	402056	1	%MD16.3	Axis0 Command Parameter 3
7	402057	402058	1	%MD16.4	Axis0 Command Parameter 4
8	402059	402060	1	%MD16.5	Axis0 Command Parameter 5
9	402061	402062	1	%MD16.6	Axis0 Command Parameter 6
10	402063	402064	1	%MD16.7	Axis0 Command Parameter 7
11	402065	402066	1	%MD16.8	Axis0 Command Parameter 8
12	402067	402068	1	%MD16.9	Axis0 Command Parameter 9

Follow the steps below to configure a range of addresses from the RMC to send commands to Axis 0.

1. In Ignition, to identify the registers, provide a prefix of one or multiple characters that will precede each register number. For this example, we will use 'M'. Enter the prefix in the first field of the row.
2. Next identify the starting and ending Modbus addresses. For the Axis0 Command Area range, you may enter any values in Ignition that have meaning (ex. could start with the digit 1). For this example, we will match the numbers in the RMC; enter 402049 in the **Start** field and 402068 in the **End** field.
3. For Ignition to treat two adjacent RMC registers as a single value, click the **Step** check box.
4. Being the first device attached, the **Unit ID** can remain at zero.
5. The **Modbus Type** indicates the type of data to exchange. For the Command Area, all registers are Floats, so select **Holding Register (Float)**.
6. Enter the starting **Modbus Address** for the range *without* the prefix of the digit 4: **2049** (this was obtained via RMCTools as shown above).
7. Leave the **Radix** value of 10 to use decimal integers.
8. Click **Save**.

Prefix	Start	End	Step	Unit ID	Modbus Type	Modbus Address
M	402049	402068	✓	0	Holding Register (Float) ▼	2049

Radix	10

Now Ignition has been prepared to send commands to Axis 0. These registers, however, don't retain their contents and reading them will typically return values of zero.

Let's establish an address to read the current position of Axis 0 for monitoring. This was prepared above in the RMCTools Indirect Data Map as indirect address 1 (400001).

1. In the **Address Configuration for 'RMC200'** screen, click **Add Row.**
2. Enter the fields as before, but this time specify the same address for **Start** and **End**.
3. Click the check box for **Step** to use both adjacent registers.
4. Specify the **Modbus Type** as **Holding Register (Float)**.
5. Specify the **Modbus Address** as **1** (again, the prefix of 4 for address 400001 is not entered).
6. Click **Save**.

Prefix	Start	End	Step	Unit ID	Modbus Type		Modbus Address
M	402049	402068	✓	0	Holding Register (Float)	▼	2049
M	400001	400001	✓	0	Holding Register (Float)	▼	1
Radix	10						

Now the current position should be observable in Ignition. We don't yet have an Ignition project to view the data, but we can use an administrative tool to view this register.

Testing connectivity with the OPC Quick Client

To view the current position of Axis 0 in Ignition, you may use the OPC Quick Client. In the left-hand menu of the Ignition Gateway, scroll to the **OPC Client** section and click **OPC Quick Client**.

OPC CLIENT

OPC Connections

OPC Quick Client
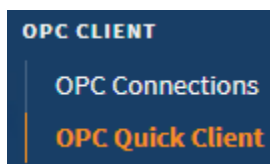
A tree of OPC tags is now available. From this tree it will be possible to directly view values from devices.

Click through the nodes of the tree until Tag M400001 is visible, as follows:

TYPE	ACTION	TITLE
Server	refresh	⊟ 📁 Ignition OPC UA Server
Object		⊟ 📁 Devices
Object		⊟ 📁 [RMC200]
Object		⊟ 📁 UnitId 0
Object		⊟ 📁 M400001-M400001
Tag	[s][r][w]	⊞ 📁 M400001
Object		⊞ 📁 M402049-M402068
Object		⊞ 📁 [Diagnostics]
Object		⊞ 📁 Server

To directly read the register, click **[r]**. The value will appear in a region above the tree. The Quality should indicate "Good". The [r] link may be clicked multiple times to display the most recent value.

✔ **Read completed.** `[Ignition OPC UA Server]ns=1;s=[RMC200]M400001`

Value: 0.0

Quality: Good

Timestamp: 3/16/22, 8:13:19 AM PDT

TYPE	ACTION	TITLE
Server	refresh	⊟ 📁 Ignition OPC UA Server

It is also possible with the Quick Client to watch the value over time without repeatedly clicking [r]. To do this, click **[s]** to "subscribe" to the tag for a while. This tag will be added to a list below the tree and the value will update every second. Presently, unless there is a program running in the RMC200, the value will remain constant.

Subscription 1 [x]   [Add]

Server	Address	Value	Quality	Timestamp	
Ignition OPC UA Server	ns=1;s=[RMC200]M400001	0	Good	3/16/22, 8:15:53 AM PDT	[x]

In addition to reading, it is possible to write to registers using the Quick Client using the **[w]** link. After clicking this link for a register, a dialog box will appear for entering a target value. Note that the data type, such as Float, should also be selected.
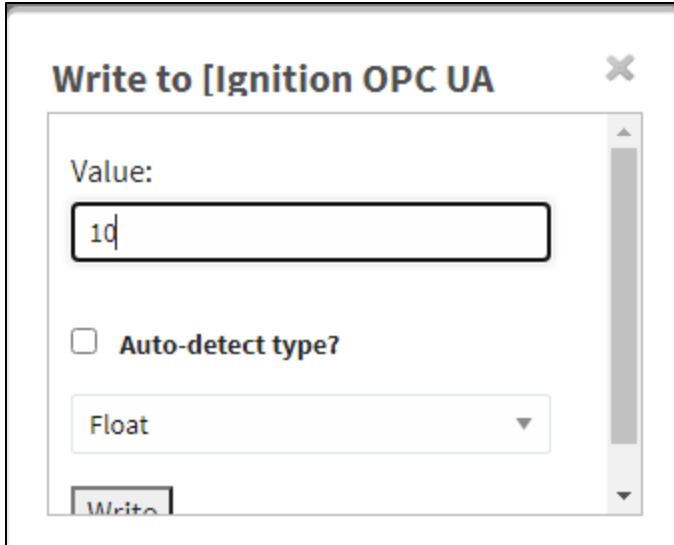
NOTE: An RMC will process a command as soon as a value is written to the Axis Command register, so the registers should be written in reverse order, parameter registers first, and command register last.

IMPORTANT: Prior to sending a command such as Move Absolute (20) to the RMC, be sure to check the Axis Tools and read for Axis0 the values for Positive Travel Limit and Negative Travel Limit. Use position values within this range.

To send a Move Absolute (20) command to Axis 0 using Quick Client, write values from to M402057 to M402049 in the following order:

-    M402057: 10   [decel rate]
-    M402055: 10   [accel rate]
-    M402053: 5   [speed]
-    M402051: 15   [position] (use a value that is in range for Axis0 on your RMC)
-    M402049: 20   [command]

Now the subscribed actual position register should update to display the current position commanded, in this case 15 position units.


With this configuration, Ignition is now prepared to send commands to Axis 0 of the RMC200 and monitor the current position of the axis. Repeat these steps to configure additional addresses.

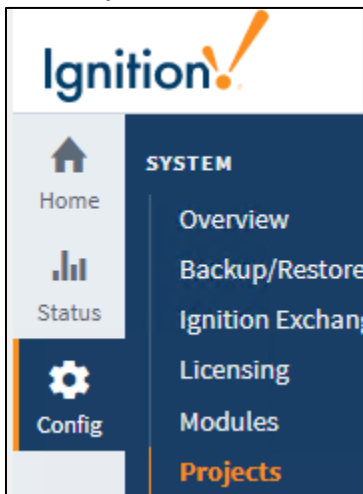# Configuring an Ignition Vision Project for an RMC200

Creating a new project

Open the Ignition Gateway in your browser (http://localhost:8088). Follow the steps below to establish a new project.

1. Click **Config**
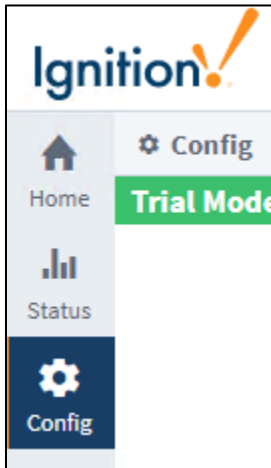
2. Click **Projects**

3. Click **Create new project**

    → Create new project...

4. Fill out the **Project Settings** and **Connections** as follows. You may use your own name and title, however, note the Connections properties.

Project Settings

Name *	RMC200Demo
Description	
Title	RMC200 Demo The title for the project. This can contain more characters than the name (space, etc), and will be used to represent the project to users. If empty, the name will be used.
Enabled	☑ A disabled project will not be active on the Gateway, but will remain editable in the Designer.
Inheritable	☐ Inheritable projects are not runnable as a stand-alone project, but are intended to provide shared resources to one or more child projects.
Parent Project	-None- ▼

Connections

User Source	default ▼
Default Database	Sample_SQLite_Database ▼
Default Tag Provider	default ▼

5. Click **Save**. Observe the confirmation message.
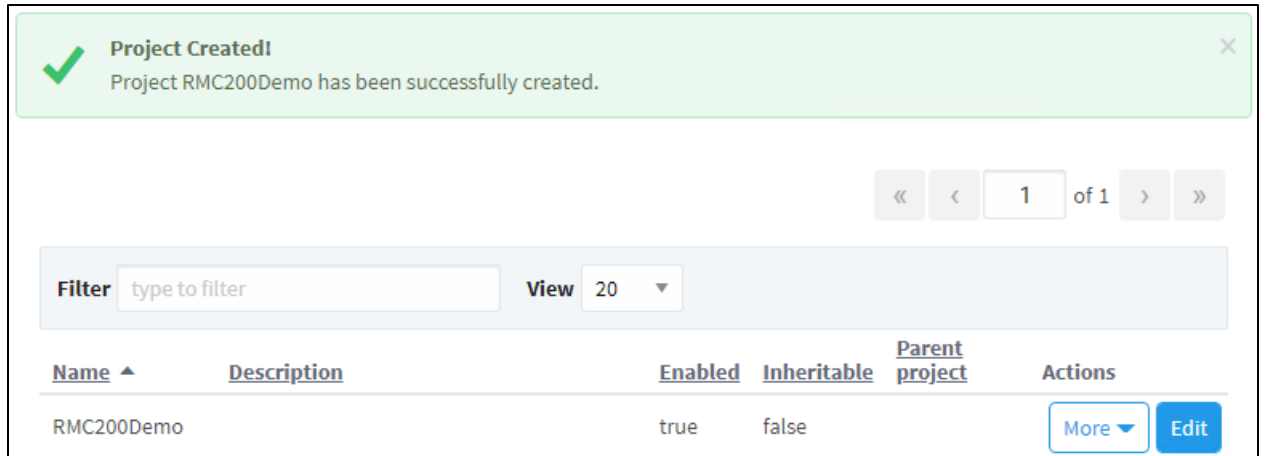
Your project is now ready for development in Ignition's Designer.

Installing and Launching the Designer

The Designer is an additional piece of software that can be installed from within the Gateway. Use the following steps to install and launch the Designer.

1. Click **Get Designer** button in the Ignition Gateway administration tool.
2. Review the 3-step instructions.
3. Click **Download for Windows.**
4. Run the Designer Launcher installer (designerlauncher.exe).
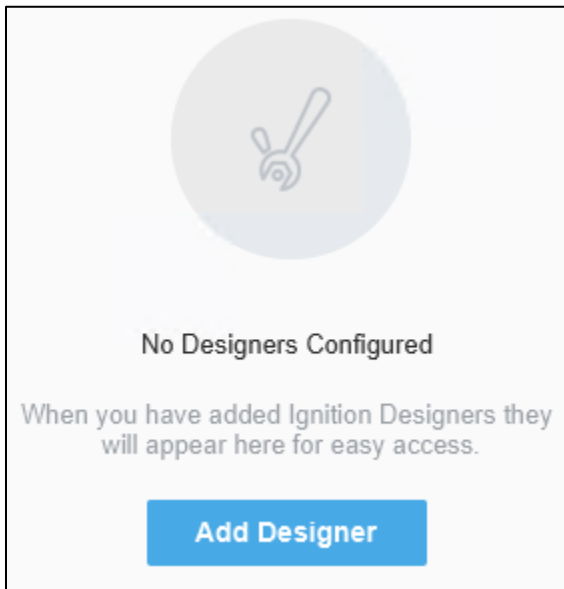5. Select **Install for all users.**
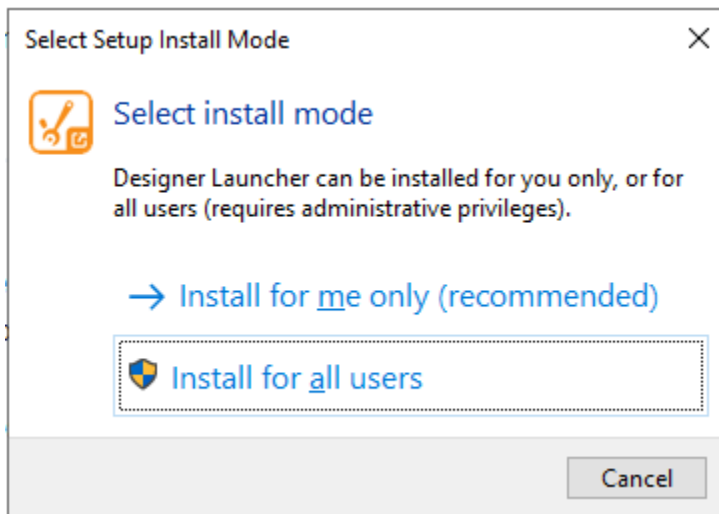
6. Accept the default values and click **Install.**
7. Leave the following check box enabled: **Launch Designer Launcher.**
8. Click **Finish.**

9. When prompted, allow access for firewall for Domain networks and Private networks.
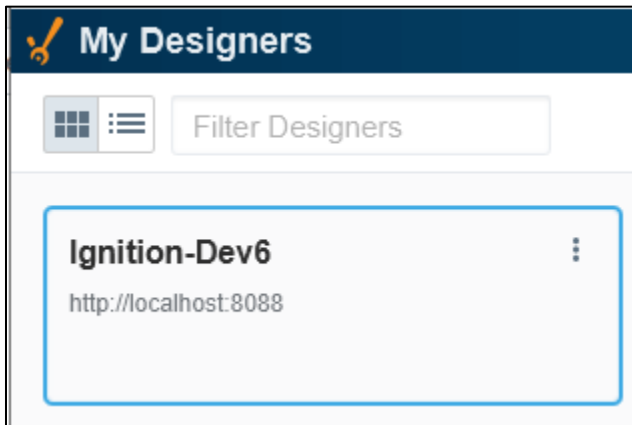

10. Click **Allow access.**
11. Click **Add Designer.**

12. Click the entry in the table where the URL is localhost:8088.
13. Click **Add Designer.**

14. Observe the rectangle with title "Ignition-*your-host-name*" indicating localhost:8088.

15. To launch the designer, click the rectangle and then click the **Open Designer** button (lower right).
16. Log in to the Designer using the administrator credentials configured earlier.

Observe two projects, the "RMC200Demo" created above and "samplequickstart".

To open the Designer for our tutorial project, click the **Open** button on the line indicating **RMC200Demo**.


Configuring tags

With a new project, before starting user interface design, an important first step is to configure tags in the Ignition design environment for devices with which you wish to communicate. These correspond to the addresses configured for a device in the Gateway administration.
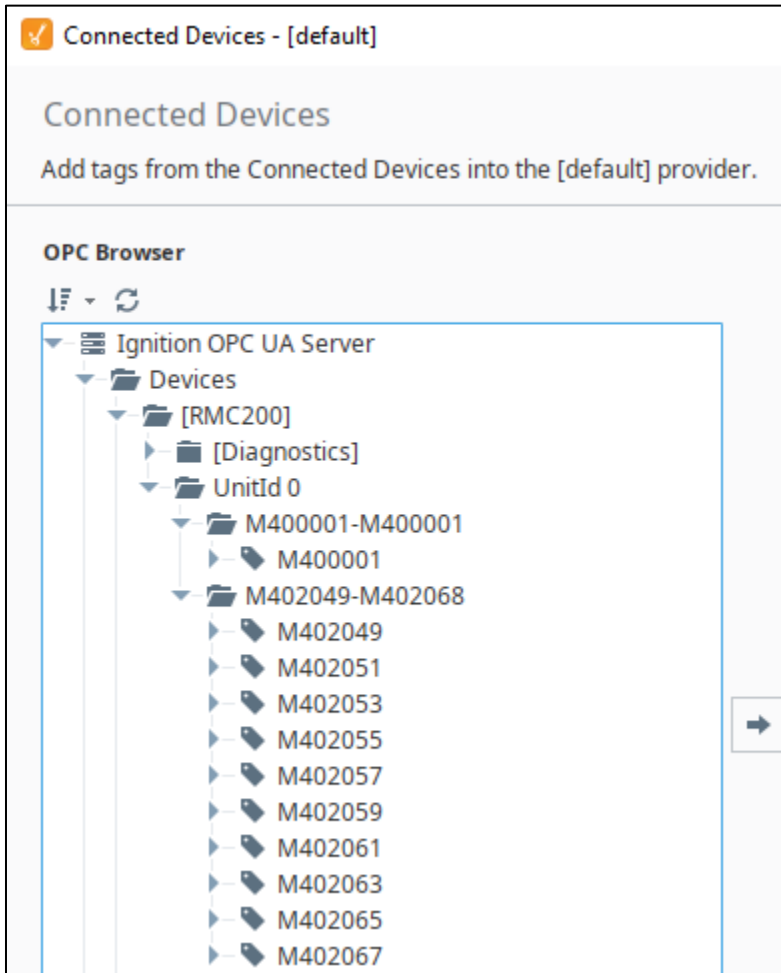
Tags are configured in the Designer in the Tag Browser panel positioned at the left of the window in the vertical center.
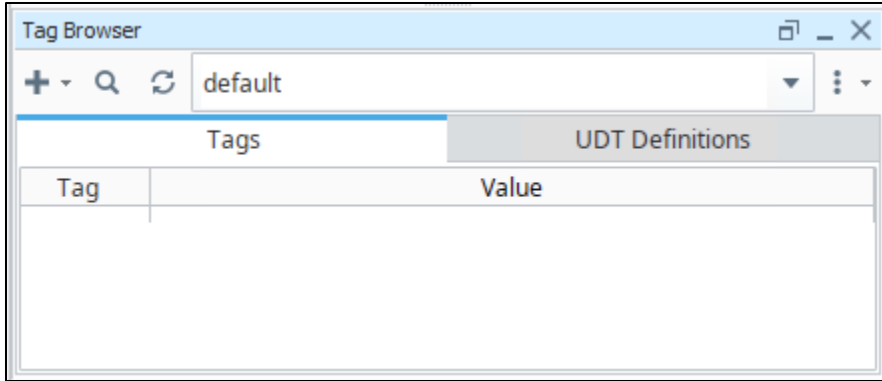
To easily establish the tags that were configured earlier for the RMC200, click the '+' menu and select **Browse Devices**.
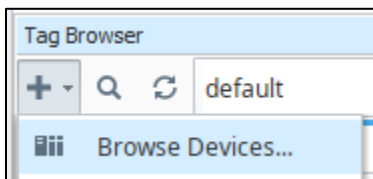
In the **Connected Devices – [default]** dialog box, expand the tree in the OPC Browser list to reveal the tags for the RMC200.


Organization of tags should be considered because text references are made throughout the project to the full path of a tag. Rearranging or renaming tags later does not update these references. An example of a full tag path is "[default]RMC200/M400001".

To establish a structure, at the right side of the dialog box, click the '+' menu and select **New Folder**. When prompted for the name of the folder, enter something like 'RMC200'. If you expect to have many RMC devices of the same type, using a suffix like '-1' may be a good choice (ex. RMC200-1). The folder will appear in the **For Provider [default] – Tags** list in the same dialog box.
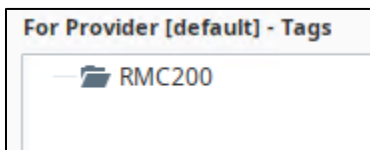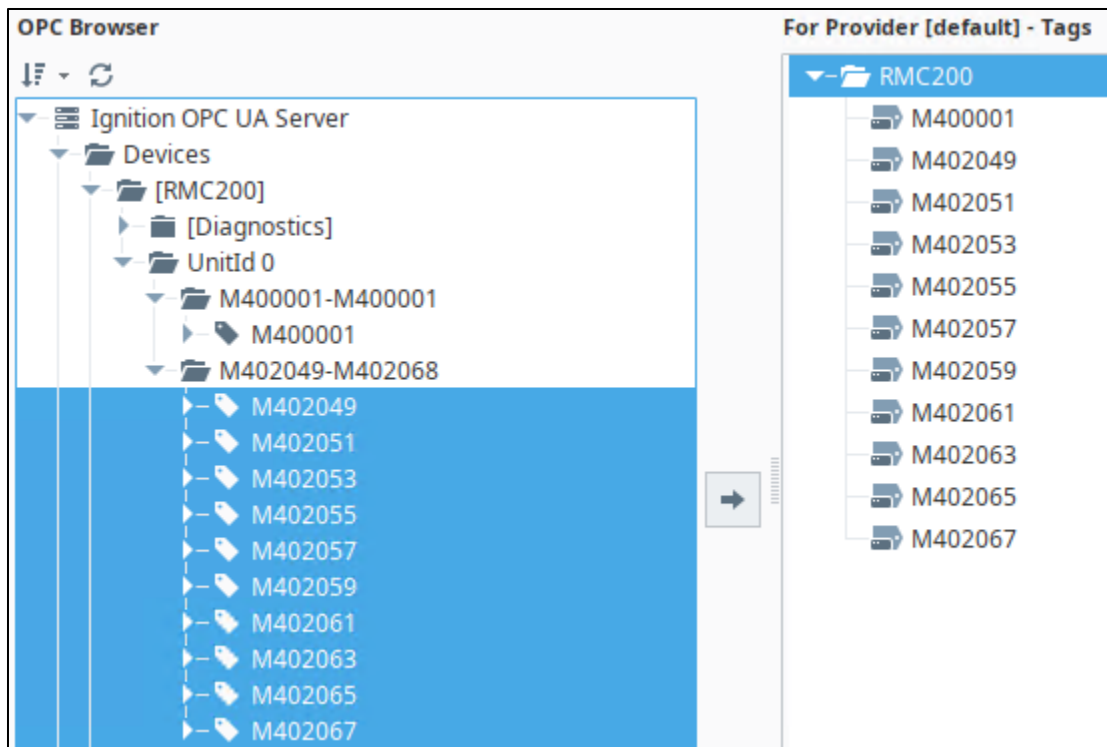
With the folder established, click the folder name once to highlight it, then click a tag name in the left **OPC Browser** list and click the right-arrow button between the lists.
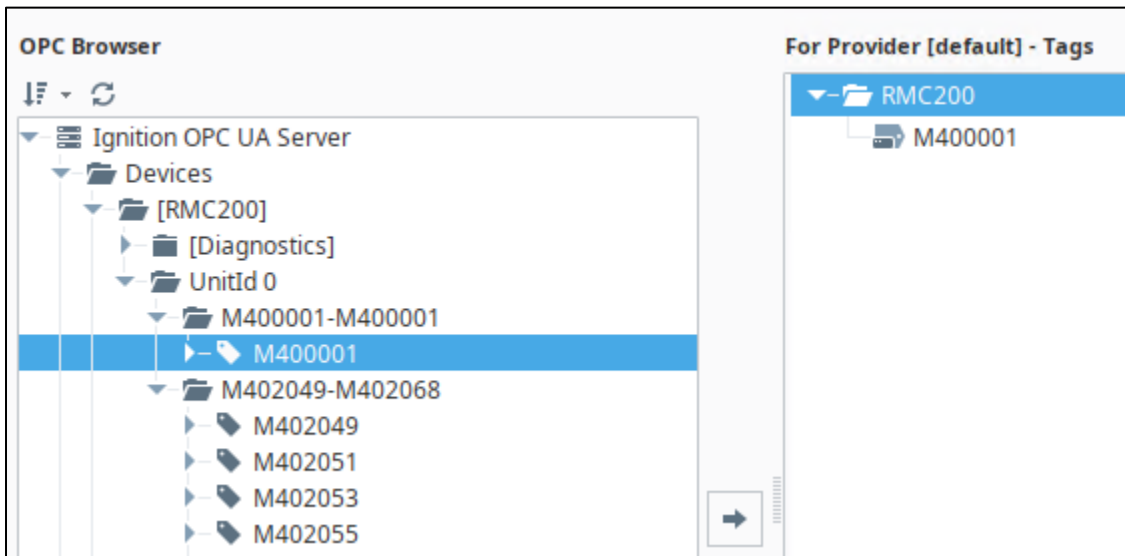
Multiple tags may be selected in the OPC Browser list, and upon clicking the right-arrow button, all will be copied into the selected folder.
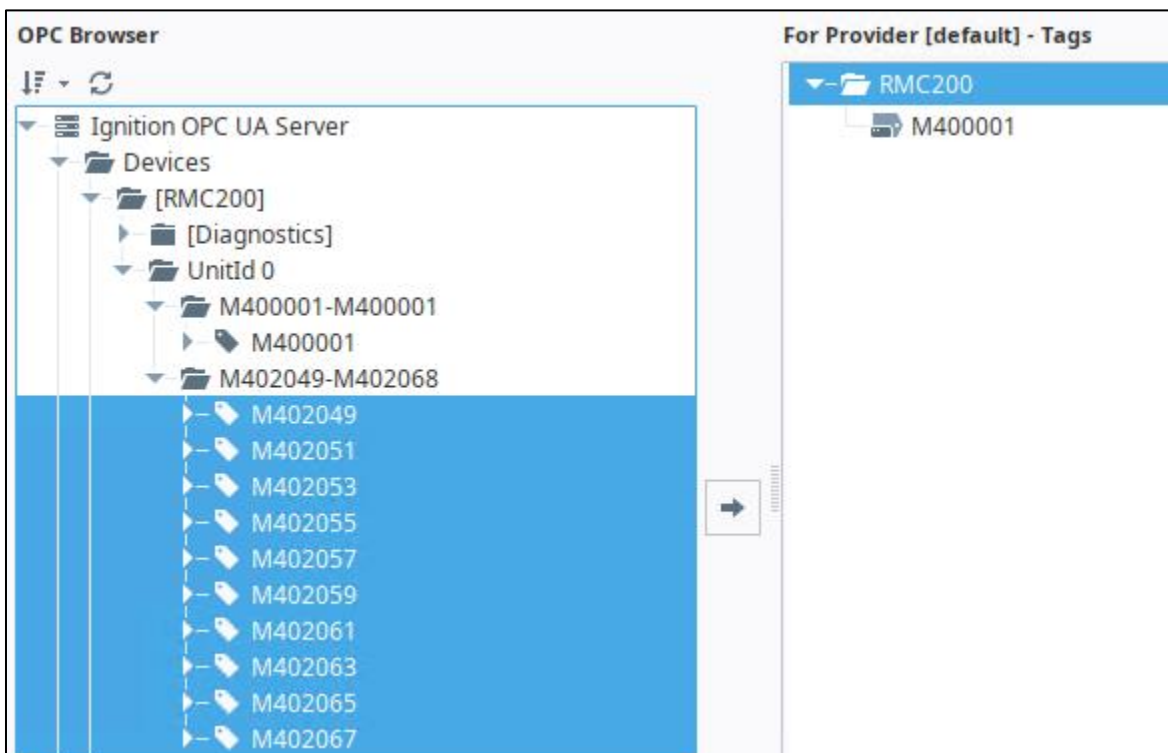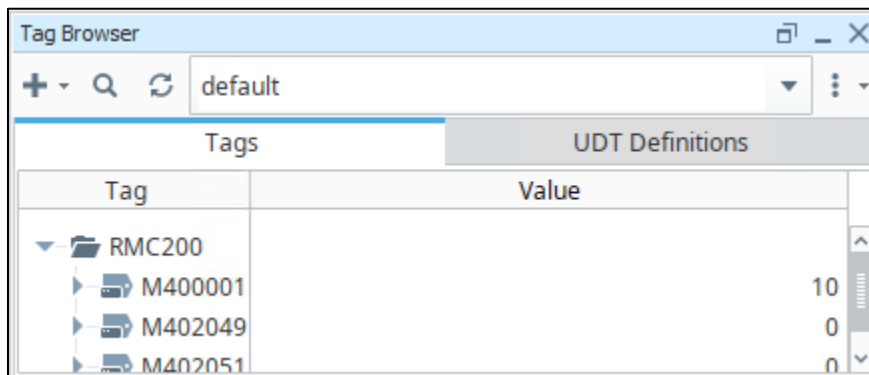
The folder will now contain all tags of interest.

Click the **OK** button to save the tag assignments.

The tags are now imported into the Project and will appear in the Tag Browser. The current values of tags from the connected device are immediately visible in the Tag Browser.

This process may be repeated if more Modbus register addresses are added to the device later.

The imported Tags are now available to be used within HMI screens of the project.

Designing a basic Vision HMI

Ignition offers two tools for design human-machine interface (HMI) applications: **Vision** and **Perspective**. These are two different SKUs offering different designer and user experiences. Perspective is the newest design tool for Ignition and was developed for mobile applications to be used on phones and tablet devices, though apps built with this tool may also be used on the desktop. Vision is the original design tool of Ignition and is designed for desktop or industrial PC applications.

For the purposes of this tutorial, we will use the Vision designer.

Let's imagine a single screen where we wish to control the position of a hydraulic cylinder via some buttons. We may wish to command the cylinder to discrete positions, such as 5 or 10 position units. We may also wish to 'jog' the cylinder by an amount specified by the user (extend or retract). We would like to view the position of the cylinder graphically and observe its motion. Additionally, we would like to record into a database and graph the motion resulting from our commands. Many more topics can be addressed with Ignition, such as generation of alarms when parameters exceed limits we set, notifications about such conditions, and reports, but these are out of scope for this tutorial.

We will simulate a hydraulic cylinder graphically using a simple built-in control of Ignition, the progress bar. As mentioned, buttons will serve to issue commands to the RMC controller.

Creating the first window

In the center of the Ignition Designer is a set of choices for creating a window. You may enter the name for the new window and select the type to create.

Enter **RMC200** as the window name, select **Main Window** and click **Create**.

The Vision designer will now appear with a blank canvas onto which controls may be established.

Laying out the HMI

The available controls are listed in the Component Palette at the right edge of the Designer window. They are categorized by type. These may be dragged from the palette onto the canvas.

Start by dragging to the canvas a **Progress Bar** and a **Button** and a **Numeric Text Field**. Position them so that the text field is next to the progress bar.

Position units referenced here correspond to our test environment. For your environment, in RMCTools, you may find the limits of travel using the Axis Tools listing, and from there you may also use the Simulator Wizard to configure the limits.

On the canvas, click the button and set its text using the Text property in the **Vision Property Editor** found at the lower left of the window. Let's title it 'Move to 5'. The layout should look something like this:
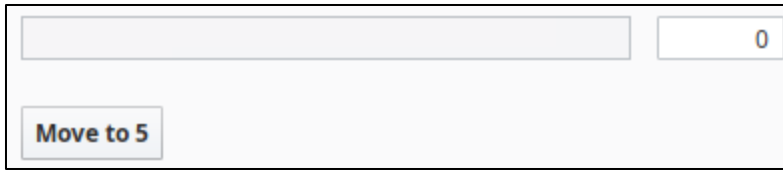
To indicate the current position in the progress bar and in the text field, we can reference Tag M400001.

From the Tag Browser on the left, click and drag M400001 to the progress bar and release the mouse button. Repeat the drag and drop to the text field using the same Tag. The values for these two controls will update immediately.

To set the limits for the progress bar, in RMCTools the upper limit is 30 position units (pu). Click the progress bar and set the Maximum in the Vision Property Editor to 30. The display now looks something like this:
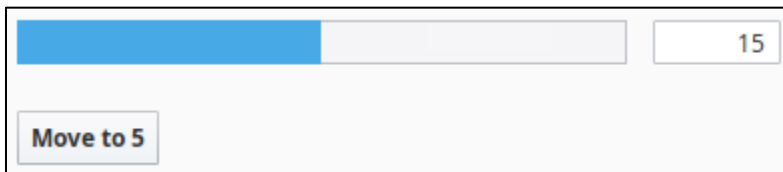
Let's program the button to move to a position of 5 pu.

Double-click the button and in the Component Scripting dialog box that appears, observe that the context is to specify an action in the actionPerformed event. We will use a script. Select the **Script Editor** tab at the upper right.
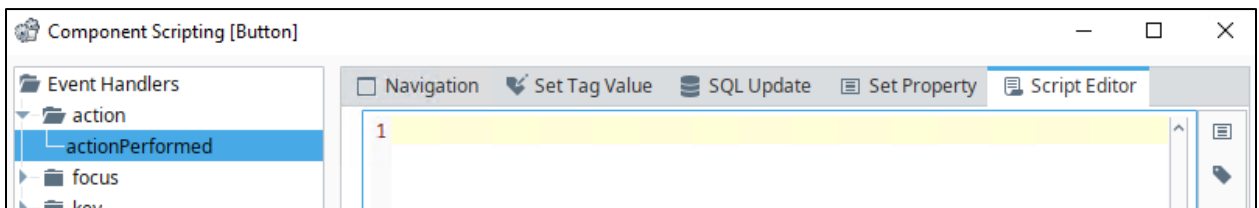
Since to send a Move Absolute (20) command to the RMC, we need to configure a command with four parameters, we will set up two arrays, one for the Tags needed for the command, and one for the Tag values. Then with a single statement (system.tag.writeBlocking) we can send the values to the RMC via the Tags. As with the Quick Client, we will position the command value last to ensure all parameters are written first. The Tags of interest, as before, are M402049 through M402055.

In script, Tags are identified using their Tag paths. To easily obtain a Tag path, right click a Tag in the Tag Browser and select **Copy Path**. This will place the text of the path on the clipboard.

The array of Tag paths will contain each in quotes separated by commas. We will accelerate and decelerate by 10, move at a speed of 7 to a position of 5 using command 20.

The code for the script is the following:

```
tags = ["[default]RMC200/M402057", "[default]RMC200/M402055",
"[default]RMC200/M402053", "[default]RMC200/M402051",
"[default]RMC200/M402049"]
```

```
values = [10.0, 10.0, 7.0, 5.0, 20]
system.tag.writeBlocking(tags, values)
```

Click **OK** to close the Component Scripting dialog box containing the above script.

Click the button once and use copy & paste to establish a second copy of the button.

Double-click the second button and change the value 5.0 in the **values** array to a higher value like 22.

Use the Vision Property Editor to update the text of the button accordingly. Most likely the button will now show "Move to …" since the text cannot fit. To make the button wider, click the button once and then use the arrows at the right edge, dragging to make it wider.
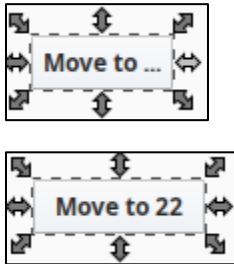
Now the project is almost ready to send commands and observe the results. There is one additional setting that must be configured prior to testing, however: click the **Project** menu in the menu bar and select **Comm Read/Write**. You'll notice it is initially set to **Comm Read-Only**, which will prevent our buttons from writing to the Tags.

To test, click **Project -> Preview Mode** (or use the F5 key). Click the buttons and observe that the progress bar and numeric text field reflect the motion.
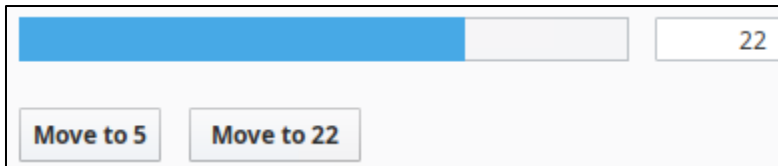
If you have RMCTools also connected to the same controller in Online mode, you will observe the same motion in the Actuator View; also, an RMCTools plot may be activated to capture the same.

In Ignition Designer, return to design mode using the same menu or the F5 key.

Instead of buttons, the commands can also be sent by direct user input. Let's try the **Slider** control. Drag a Slider onto the canvas from the palette. Using the Vision Property Editor, change the Maximum value to the upper limit of your axis, which in our case is 30.

To send a command when the slider changes, we will apply a script similar to our button script.

To assign a script, right-click the slider and select **Scripting…** on the menu. In the **Event Handlers** list on the left, click to expand the propertyChange folder, then click **propertyChange**. Click **Script Editor** among the tabs along the top.
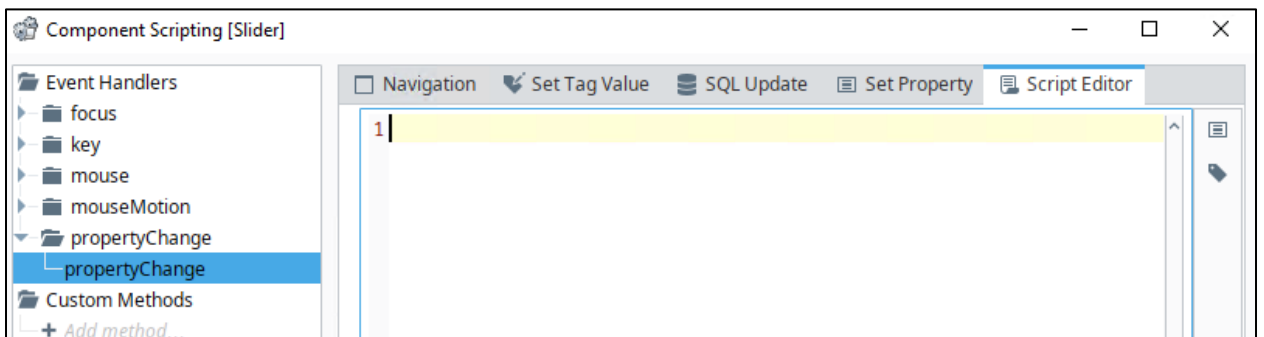

The script will obtain the target position value from the slider, otherwise it is the same as used for the buttons. The slider value can be obtained by referencing **event.source.value**.

To make the axis more responsive for the slider, let's set the velocity to 30 pu/s.

```
tags = ["[default]RMC200/M402057", "[default]RMC200/M402055",
"[default]RMC200/M402053", "[default]RMC200/M402051",
"[default]RMC200/M402049"]

values = [10.0, 10.0, 30.0, event.source.value, 20]

system.tag.writeBlocking(tags, values)
```

Use the F5 key or click **Project -> Preview Mode** to test the slider. The progress bar and text field should quickly reflect the commanded position indicated by the slider.
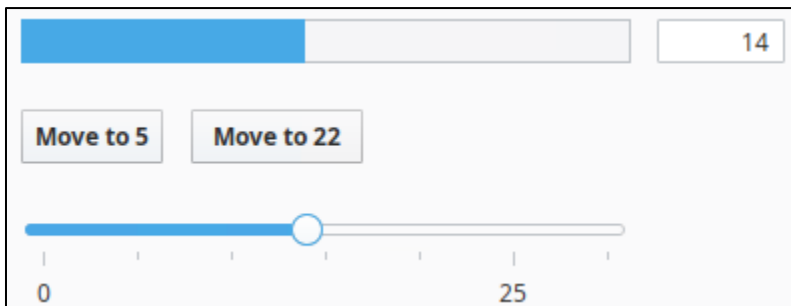
You may notice the updates seem to occur about once per second. This is true initially, and the sampling rate may be changed.

The sampling rate is configured for the tag, as opposed to the controls. Each tag is assigned to a Tag Group, and the Tag Group has a sampling rate. To change the rate for the current position tag, we first need to establish a new Tag Group, then assign it to the tag.

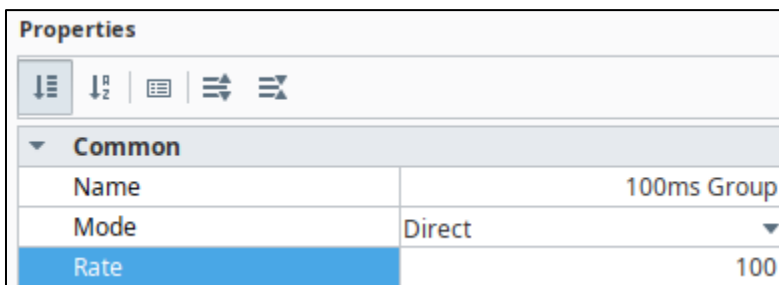In the **Tag Browser**, click the vertical ellipsis icon and select **Edit Tag Groups**.
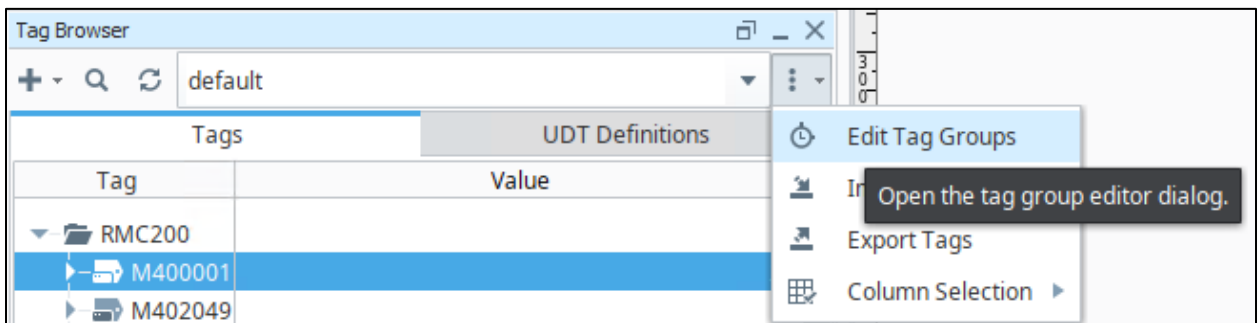
In the **Tag Group Editor** dialog, click the '+' icon next to the **Tag Groups** list.
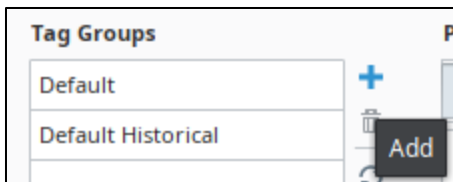
Enter a name, replacing the text 'New Tag Group' in the **Name** field. Let's use the name '100ms Group'.

Change the **Rate** from 1,000 to 100 (ms).

Click **OK** to save the changes.

To change the sampling rate for Tag M400001, double-click **M400001** in the Tag Browser. Change the Tag Group property from **Default** to **100ms Group**.
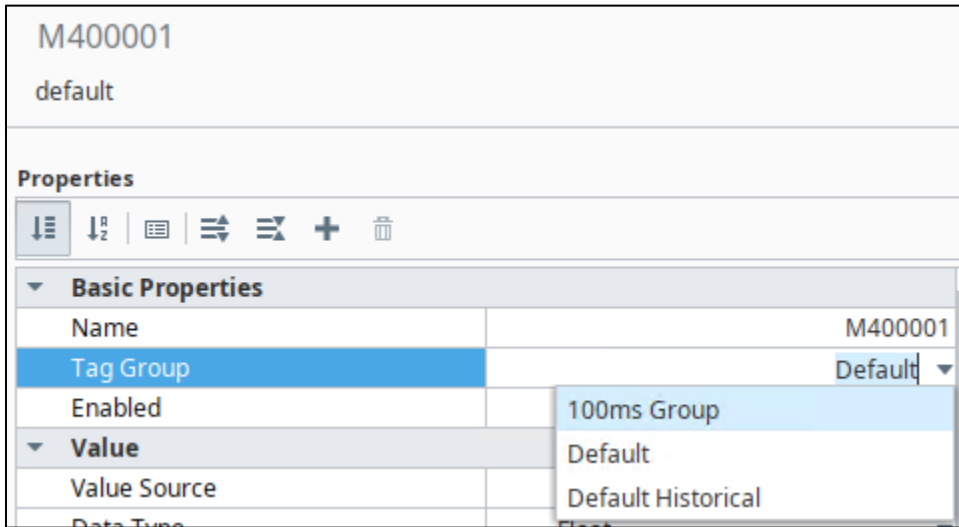

Now preview the application again and observe that the refresh occurs 10 times per second.

Capturing history

We have established real-time control and feedback of an RMC from Ignition. It is also possible to record the history of tag values in a database and subsequently depict the results in charts. Let's configure the current position tag to record to the demo database available due to installing the Quick Start option.

To enable history for a tag, double-click the tag in the Tag Browser. Scroll to the bottom of the properties list and change the value for **History Enabled** to **true**.
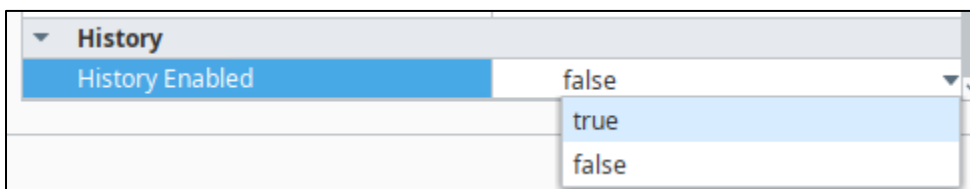
 Scroll down and change the value for **Storage Provider** to **Sample_SQLLite_Database**.
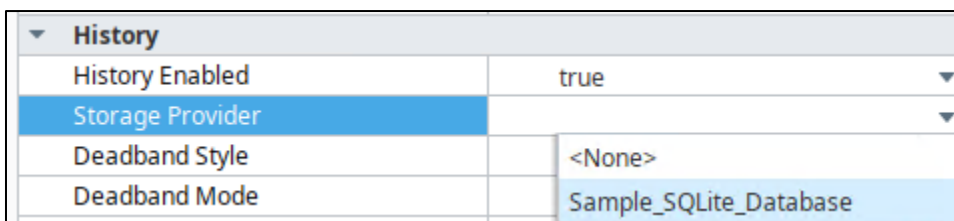
Click **OK** to close the dialog box.

Other database connections can be configured in the Gateway administration and used here.

Let's capture some history and then add a chart to show the results.

Enter preview mode (F5) and use the slider to perform some motion on the axis. Exit preview mode.

From the Component Palette, under Charts, drag an **Easy Chart** control to the canvas and size it to your liking.

Double-click the Easy Chart to view the **Easy Chart Customizer** dialog.

Click the '+' icon to the right of the Tag History Pens list. Give the pen a name.
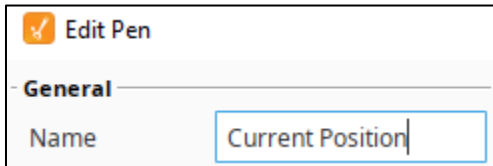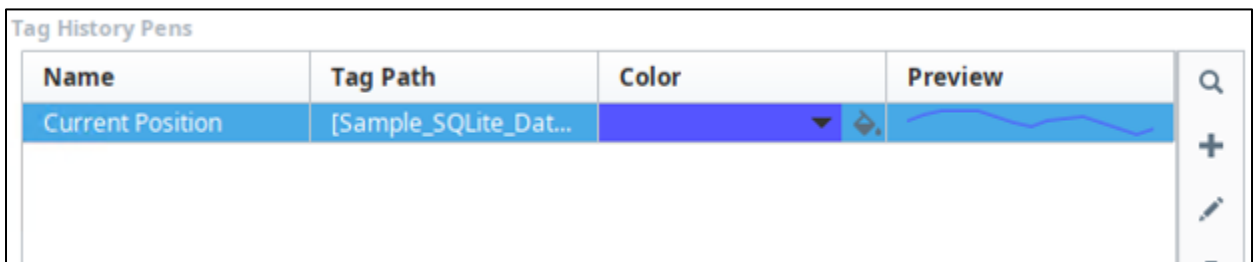

To add the tag to the chart, click the icon to the right of the Tag Path field, then expand the tree for 'default' down to the tag name and click the tag name.
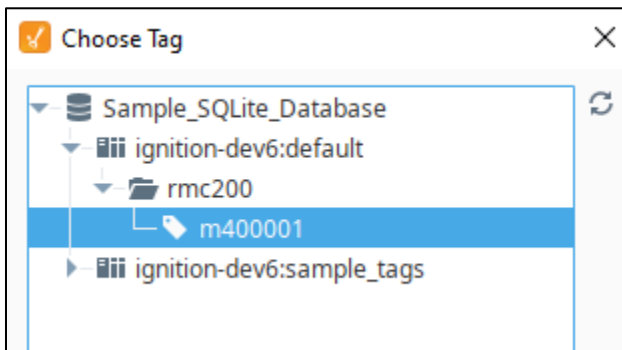
Click **OK** to confirm the choice. Click **OK** again to close the Edit Pen dialog.

Observe the tag in the **Tag History Pens** list. Click **OK** to accept this change.

With the Easy Chart selected on the canvas, in the Vision Property Editor panel (lower left), change **Chart Mode** from **Historical** to **Realtime**. Since we changed the tag sample rate to 100 ms by assigning a new Tag Group, change the Poll Rate for the Easy Chart to 100.

Enter preview mode (F5) and use the slider to generate motion on the axis.

Observe the results in the Easy Chart.
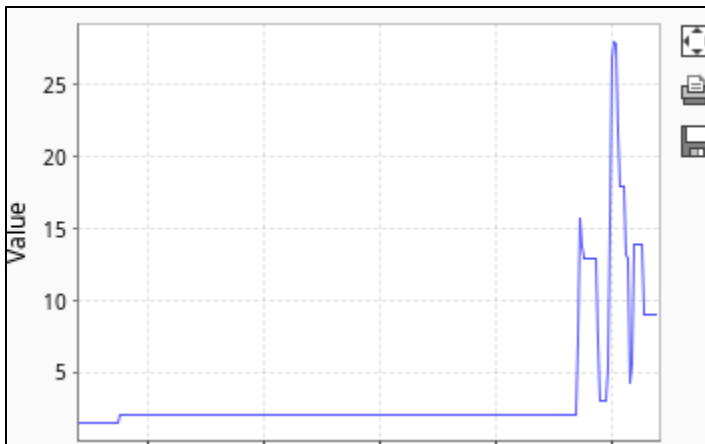
You may change the time range plus start and stop the Easy Chart display using the controls below the chart.

Ignition will continue recording history even when motion is not occurring, depending on a setting known as "deadband". It may not be desirable to store much non-movement data in the database. To better control this, you may change the **Historical Deadband** property for the tag. To do this, double-click the tag in the Tag Browser panel, scroll down to the History section, and enter a larger value in the Historical Deadband field. For example, enter 0.1 to replace 0.01.

▼　History	
History Enabled	true　▼
Storage Provider	Sample_SQLite_Database　▼
Deadband Style	Auto　▼
Deadband Mode	Absolute　▼
Historical Deadband	0.1

Other tag parameters also affect the recording into history. Namely, the **Min Time** settings control how often Ignition will record a value independent of the deadband. Increase to reduce the amount of history recorded for the tag.

| Sample Mode	On Change　▼
Min Time Between Samples	1
Min Time Units	Seconds　▼
Max Time Between Samples	Milliseconds
Max Time Units	Seconds
	Minutes
	Hours

Click **OK** to save the change(s).

You may read more about how the historian works here in the Ignition documentation online: https://docs.inductiveautomation.com/display/DOC81/How+the+Tag+Historian+System+Works

When changes to a tag occur from behavior outside of an Ignition HMI, Ignition will continue to record the history into the database.

Connecting multiple RMC controllers

As simply as we have configured control, monitoring and history for a single controller, the above process can be used to establish connectivity to multiple RMC controllers, even of varying types. A single control such as a button or a slider can send commands simultaneously to combinations of controllers. To do this, in the Gateway administration, add devices and configure addresses; then in the Designer, import the tags and repeat the above steps to establish controls on the HMI.

It is easy to view tag data from multiple controllers on a single chart by configuring history for desired tags, then adding corresponding pens to a chart.
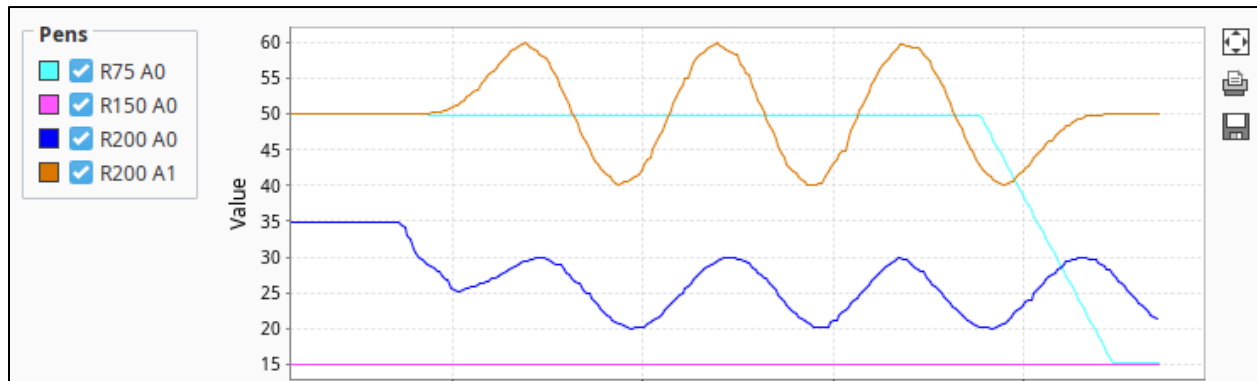
To send commands simultaneously to multiple controllers, add Python code for additional tag paths and values to a control such as a button, as we've seen. Below are the results from an example commanding multiple controllers at once.
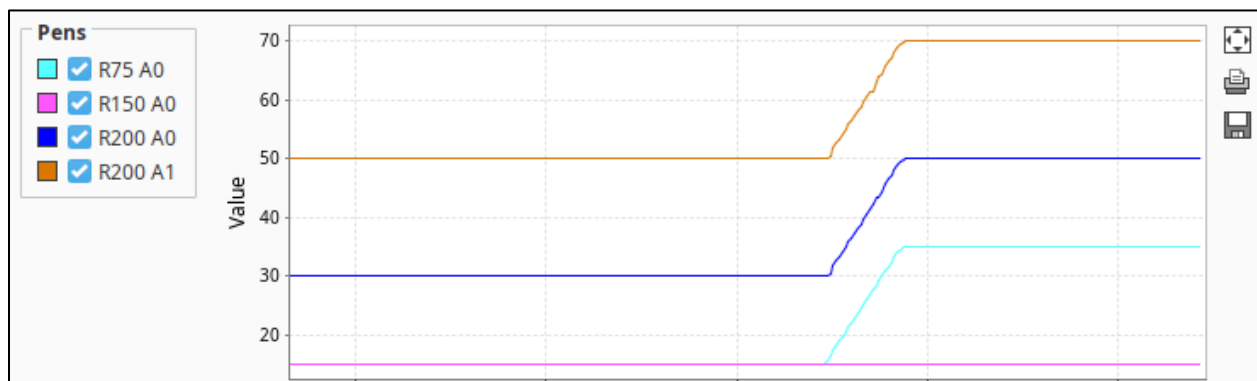
Summary

This tutorial has guided you through the process of preparing your RMC for Modbus connectivity, establishing Modbus addresses in Ignition, and designing a basic HMI to read and write to the RMC, as well as recording and displaying history for a tag.

Inductive Automation offers a free online University for learning about using and designing with Ignition. Visit https://inductiveuniversity.com/ for more information. General information about Ignition may be found at https://inductiveautomation.com.