Standard Motor Interface Driver for Tridium Niagara

User Guide

baudrate.io

niagara4

February 22, 2026

Contents

1 Introduction **2**

2 Niagara and SMI **2**

3 Requirements **4**

4 Quick Start **4**

5 Smi Network **4**

6 Smi Interface **5**

7 Smi Points **5**

8 Smi Command **6**

9 FAQ **8**

9.1 I cannot discover any serial interfaces 8

9.2 Motors are not responding to commands 8

9.3 How do I resolve duplicate motor addresses? 8

1 Introduction

SMI[1] or Standard Motor Interface is a protocol for electric drives used for sun protection devices: shutters, roller blinds, awnings, exterior and interior venetian blinds, pleated blinds, roman shades. For blinds and shutters, SMI is the same as DALI is for lights. Like DALI, SMI allows combining products of various manufacturers and using open standard technology.

Figure 1: Types of window coverings

Key features:

- Up to 16 drives can be wired and controlled in parallel
- A single five-wire line is used for power and communication
- High-precision intermediate positioning
- Query current motor positions
- Motor feedback with diagnostic information

2 Niagara and SMI

SMI driver for Niagara can be deployed in any Niagara-compatible embedded device like Tridium JACE, Honeywell WEB8000, Distech Controls EC-BOS-8, Johnson Controls FX80 etc. Niagara device connects via RS-485 port to one to sixteen Vestamatic IF SMI RS-485 interfaces (further referred to as

[1] All trademarks or registered trademarks are the property of their respective owners

"serial interface"), each of which communicates with up to 16 SMI motors. This allows controlling and monitoring up to 256 motors per RS-485 port.

In the next driver version an IP gateway will be supported. Then Niagara device will be able to communicate with many hundreds of motors via IP backbone.

SMI driver for Niagara discovers all connected serial interfaces and SMI motors, periodically reads motor points, and sends commands to an individual motor or multiple motors at once. In addition, SMI driver can initiate motor auto-addressing to resolve address conflicts and manually change motor SMI addresses.

After all SMI points are discovered and added to Niagara station, they can be used as any other driver point: * to build mobile-friendly user interfaces, * to translate them into other communication protocols (BACnet, LonWorks, Modbus, MQTT, etc.), * to integrate into complex control strategies using position of the sun, weather services, indoor and outdoor climate, lighting.

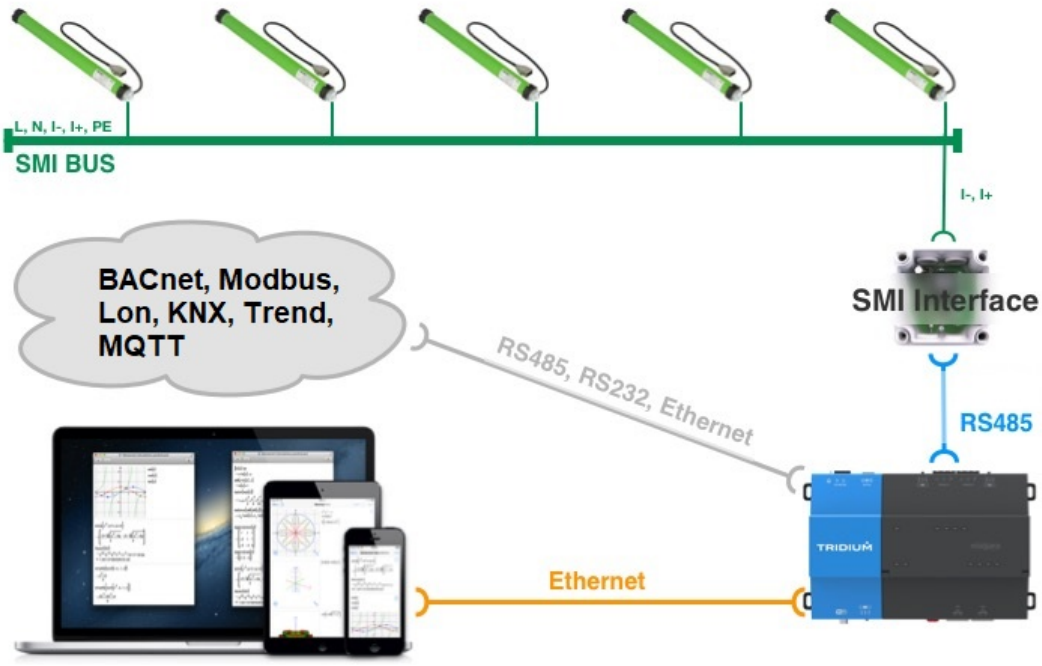

Figure 2: SMI - Niagara physical architecture

3 Requirements

- Niagara-powered device with software v4.8 or later, including JACE and all its OEM versions
- smi-rt.jar driver module and license
- one or more SMI / RS-485 serial interfaces
- SMI motors, commissioned and wired to serial interfaces according to SMI guidelines

Niagara JACE is connected to daisy-chained serial interfaces (up to 16) via its RS-485 port.

4 Quick Start

1. Copy smi-rt.jar to the Workbench *modules* folder
2. Restart Workbench software
3. Deploy smi-rt.jar to JACE from *Software Manager*
4. Restart JACE station
5. Add **Smi Network** to JACE station
6. Enter license code in **License** property under **Smi Network**
7. Select COM port **Serial Config / Port Name** property
8. Open **Smi Network** and **Discover** devices. The driver will show all serial interfaces connected to the COM port with their unique addresses and hardware versions. Add discovered interfaces to the station. The added interface will instantly display how many motors it is connected to as **Motors Present**.
9. Open device points extension and **Discover** motors. The driver will display connected motors, each with a logical address (0-15) and manufacturer hexadecimal serial number. The motor is represented as a container with points: motor position, tilt, and status.
10. Add points to the station database and Niagara starts reading their values. Writable points (position, tilt) can be modified, which initiates commands to the motors. This allows controlling blinds individually. For group control, which is the recommended way, use **SmiCommand** component from the palette.

5 Smi Network

Smi Network contains many standard Niagara properties, including:

- **License** – the code that allows the driver to run on your host.
- **Serial Config / Port Name** – JACE serial port name to which motors are connected.
- **Serial Config** – serial port settings. Defaults are 19200 / 8 / 1 / None.

- **Monitor / Ping Frequency** – how often to check communication with each serial interface.
- **Tuning Policies / Default Policy / Write On** – whether the commands should be re-sent on station restart and interface enabling.
- **Poll Scheduler** – specifies polling rates for periodic data reading:
  - Fast Rate: 5 seconds
  - Normal Rate: 10 seconds
  - Slow Rate: 30 seconds

In most cases only **License** and **Serial Config / Port Name** values need to be changed.

6 Smi Interface

Each Smi Interface (Smi Device) contains the following properties:

- **Address** – unique device address (0-15) set with DIP switches in serial interfaces.
- **Motors Present** – how many motors are currently communicating with the interface.
- **Poll Frequency** – how often all interface points are polled, refer to **Smi Network / Poll Scheduler**.

Every SMI motor contains information about its vendor (Manufacturer ID), its unique serial number and logical address (also called SMI ID). The logical address must be from 0 to 15, and there must be no identical addresses connected to one serial interface. It is expected that all SMI motors are fully commissioned and properly addressed using vendor tools before being integrated into Niagara. Still, it is possible to change motor addresses from the driver.

There are two special actions available in each Smi Device:

- **Auto Addressing** – initiate SMI auto-addressing procedure, useful for resolving duplicate addresses. May take up to 60 seconds to finish.
- **Set Motor Address** – assign motor logical address given its vendor id and hexadecimal serial number. Both vendor id and serial number can be found on motor label. Serial number is also shown during point discovery. This function is useful when a broken motor is replaced.

7 Smi Points

Every SMI motor contains three points: position (0% to 100%), tilt (-128 to +127) and status.

Position 0% – top, 50% – middle, 100% – bottom.

Tilt -128 – slats completely closed in up direction, 0 – slats completely open (horizontal), +127 – slats completely closed in down direction.

Status – enumeration value showing whether the motor is moving, stopped, or in fault. It can be used to determine the loss of communication with the motor – just add **EnumChangeOfStateAlarmExt** to generate alarm when status value is **Invalid Response**.

The points are periodically polled when they are assigned a *subscribed* status by Niagara framework. That happens in one of the following cases:

- point has alarm, history, or any other extension, except for proxy extension
- point is linked to another Niagara component
- point is used on Niagara graphics page *and* this page is open now

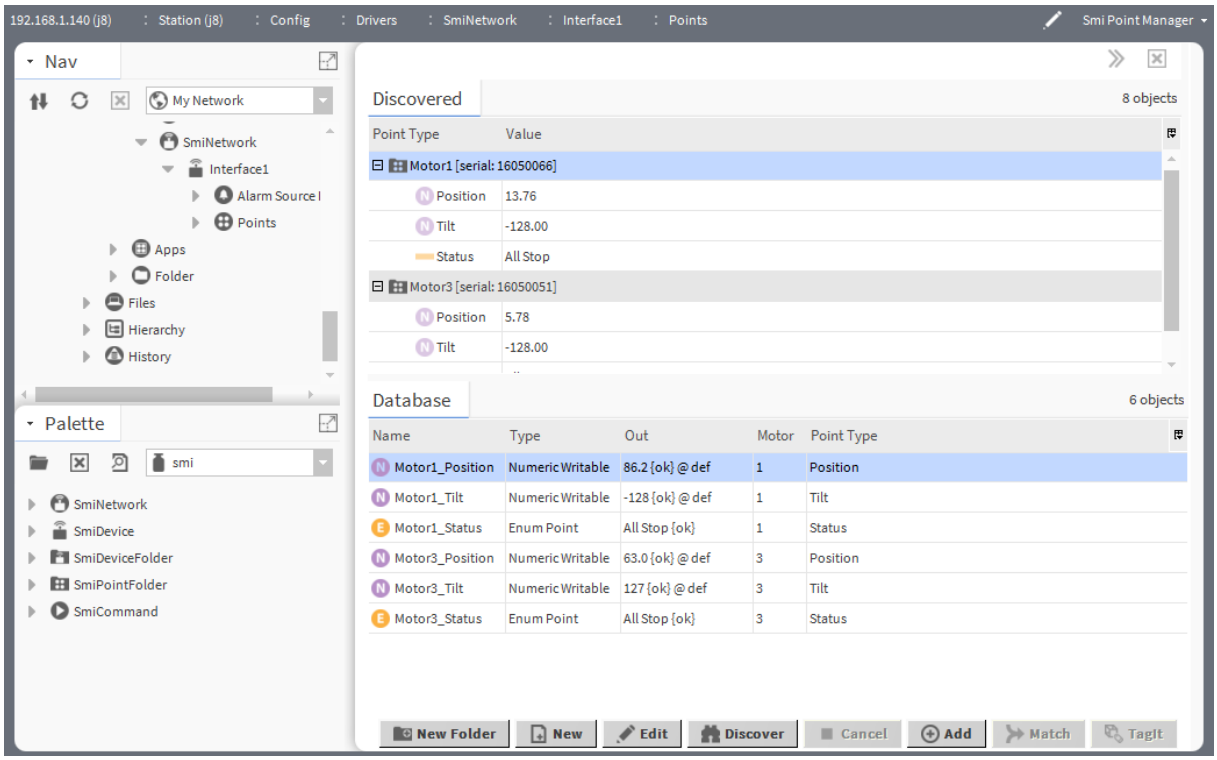

Figure 3: Smi devices and points

8 Smi Command

A very powerful SMI protocol feature is group commands. It enables multicasting – sending one command to any combination of SMI motors from one to 16. The group commands are supported by the SMI driver.

In order to send group commands, drag and drop one or a few **SmiCommand** components from the driver palette into **points** extension under serial interface. **SmiCommand** has two properties:

- **Motor Mask** – 16-digit binary mask, numbering from right to left. Examples: **1111111111111111** – all motors connected to the interface, **10011** – motors number 4, 1, and 0.
- **Status** – whether the last command was sent successfully or failed.

Property **Motor Mask** is to configure recipients of the command. Actual commands are sent when one of component actions is invoked:

- **Up** – move up until it reaches the top position or any other command is received.
- **Up X Steps** – move up for a number of steps (1-255). One step is defined as 2° rotation of the output shaft of the SMI motor (same for **Down X Steps**).
- **Stop** – stop all movements.
- **Down** – move down until it reaches the bottom position or any other command is received.
- **Down X Steps** – move down for a number of steps (1-255).
- **Position** – move to an absolute position (0-100%, where 0% is top and 100% is bottom).
- **Tilt** – move venetian blinds to an absolute tilt orientation (-128 to 127). This command uses a manufacturer-specific SMI command that is currently only supported by Vestamatic SMI motors.

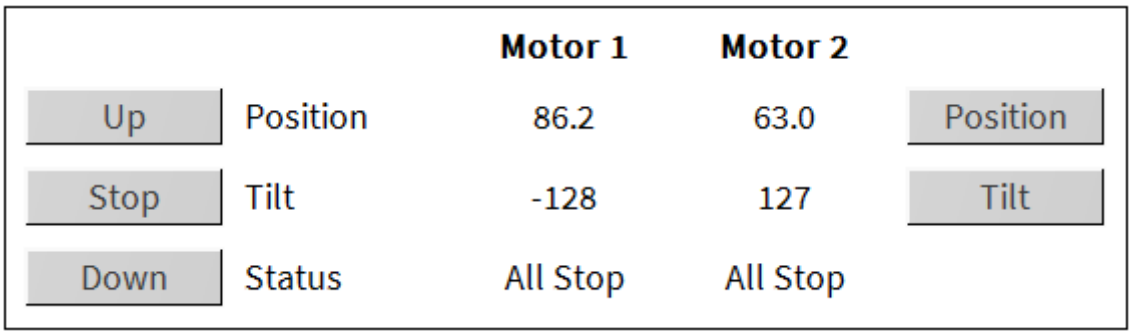

Figure 4: Example of PX graphics

There are many ways these actions can be invoked in Niagara. For example:

- Initiate actions to control all or some blinds during commissioning
- Add a few **Action Button** widgets (from kitPx palette) to PX graphics and assign their *ord* properties to **up**, **down** and **stop** actions. Users will be able to manually open, close, and stop blinds from their computers or mobile phones.
- Do the same for **position** or **tilt** and set *actionArg* property to preset value or activate *Prompt user*. Users will be able to move / tilt blinds to an absolute position.
- Make a link from control logic components to **SmiCommand**. Group commands will be sent when the link source is changed and the source value will be used as an argument (e.g., requested position).

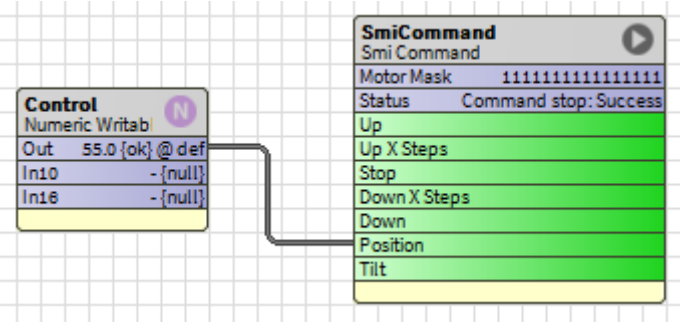

Figure 5: Smi Command on wiresheet

9 FAQ

9.1 I cannot discover any serial interfaces

- Verify the COM port name is correct and the port is not in use by another application.
- Check RS-485 wiring between the JACE and the serial interfaces.
- Verify that the serial port settings match the default (19200 / 8 / 1 / None).
- Ensure the serial interfaces are powered and the DIP switch addresses (0–15) are unique.
- Check that the license is valid.

9.2 Motors are not responding to commands

- Verify the motors are commissioned and properly addressed using vendor tools before integration.
- Check that the motor status point does not show **Invalid Response**, which indicates loss of communication.
- Ensure the motor logical addresses (0–15) are unique within each serial interface.
- If a motor was recently replaced, use the **Set Motor Address** action to assign the correct address.

9.3 How do I resolve duplicate motor addresses?

Use the **Auto Addressing** action on the serial interface device. This initiates the SMI auto-addressing procedure, which may take up to 60 seconds to complete. All motors on the interface will be assigned new unique addresses.