External Date/Time Service and Scheduling

for the R-ION

External Date Time Service

The R-ION programmable touch screen controller doesn't have a real-time-clock.

It can still be configured to display actual time and execute time schedules that are user modifiable through the display; all thanks to the **ExternalDateTimeService**.

This service provides virtual real-time-clock functionality to support :

- Time & date display
- Scheduling
- Automatic summer/winter changeover
- Time synchronization from a master device
- Time synchronization from a compatible slave modbus device
  (e.g. Ontrol R/TIO or M/TIO input-output modules)

The ExternalDateTimeService doesn't have any native (hardware) dependencies.
It can, therefore, be used on any sedona device.

Four Essential Steps

1 Add ExternalDateTimeService to your app

2 Enable auto summer/winter daylight savings time

3 Configure synchronization

4 Add schedule components as necessary

STEP

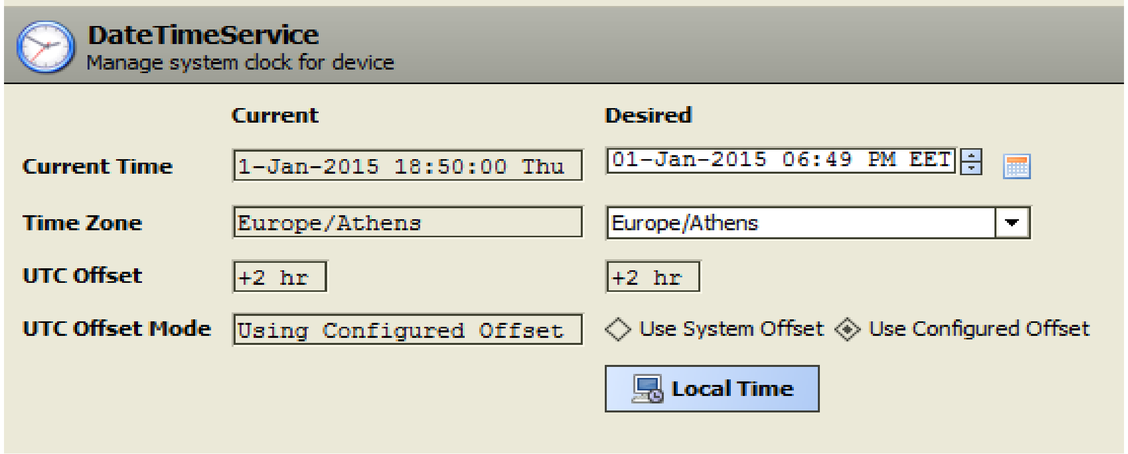
1

Add External DateTime Service to your app

Add ExternalDateTimeService to your app.

This service functions similarly to the standard Sedona DateTimeService.

Double-clicking it will show the standard DateTimeService:

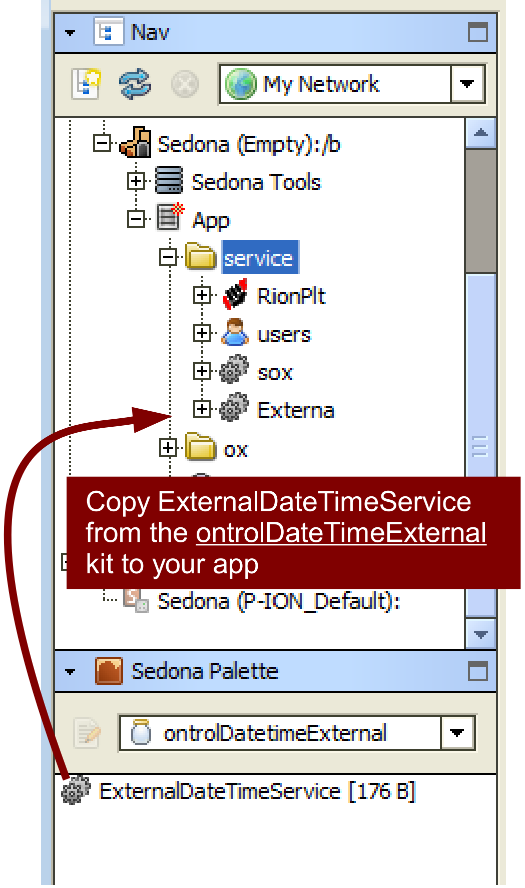


DateTimeService

Manage system clock for device

	Current	Desired
Current Time	1-Jan-2015 18:50:00 Thu	01-Jan-2015 06:49 PM EET
Time Zone	Europe/Athens	Europe/Athens
UTC Offset	+2 hr	+2 hr
UTC Offset Mode	Using Configured Offset	◇ Use System Offset ◈ Use Configured Offset

In this view, you can set the current time and date, as well as the time zone.



Copy ExternalDateTimeService from the ontrolDateTimeExternal kit to your app

STEP

2

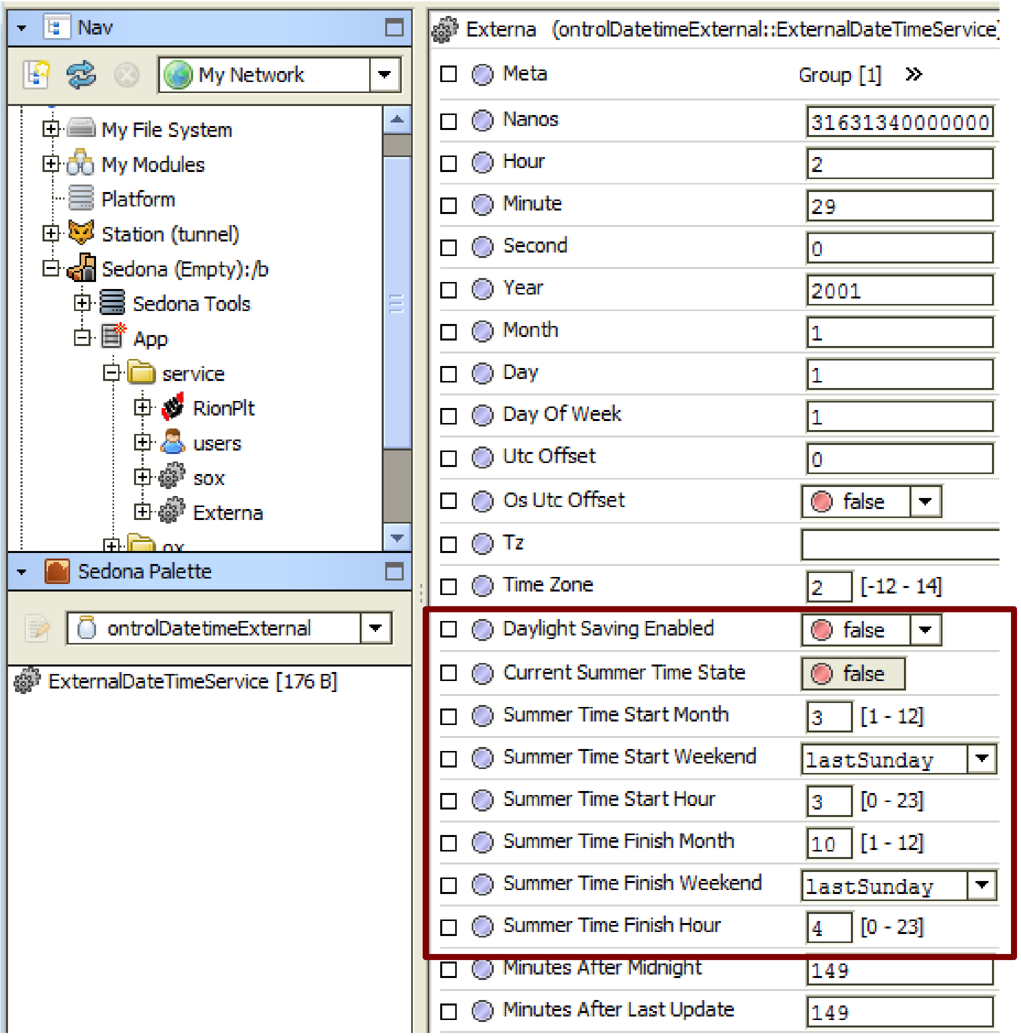
Enable auto daylight savings time

Enable auto daylight savings time

External DateTimeService provides an option to automatically change summer/winter daylight savings time.

This can be enabled on the property sheet of the service.

It can be fine tuned as well. The default settings are in agreement with regulations in most European countries as of 2015.



Externa (ontrolDatetimeExternal::ExternalDateTimeService)

□ Meta	Group [1] »
□ Nanos	31631340000000
□ Hour	2
□ Minute	29
□ Second	0
□ Year	2001
□ Month	1
□ Day	1
□ Day Of Week	1
□ Utc Offset	0
□ Os Utc Offset	false
□ Tz	
□ Time Zone	2 [-12 - 14]
□ Daylight Saving Enabled	false
□ Current Summer Time State	false
□ Summer Time Start Month	3 [1 - 12]
□ Summer Time Start Weekend	lastSunday
□ Summer Time Start Hour	3 [0 - 23]
□ Summer Time Finish Month	10 [1 - 12]
□ Summer Time Finish Weekend	lastSunday
□ Summer Time Finish Hour	4 [0 - 23]
□ Minutes After Midnight	149
□ Minutes After Last Update	149

STEP

3

Configure sync from an external source

Configure syncronization from an external source

ExternalDateTimeService relies on syncronization from an external device for accurate time-keeping.

A short-term loss of syncronization will not affect the time keeping functions. If the service is not receiving any syncronization updates, for example due to a communications fault, it will still maintain time using the internal crystal/oscilator of the device. But this is not precise, and would drift from the actual time over long periods.

There are several ways to keep the ExternalDateTime Service clock syncronized to actual time:

1. Using the real-time-clock on the R/TIO input/output module
2. Using the TimeSync feature of the sedona driver on a Niagara host (IP only)
3. Writing to registers using modbus or other protocol from a master device

...or a combination of the above.

Requirements and setup instructions for each are in the following pages.

STEP

3

Configure sync from an external source

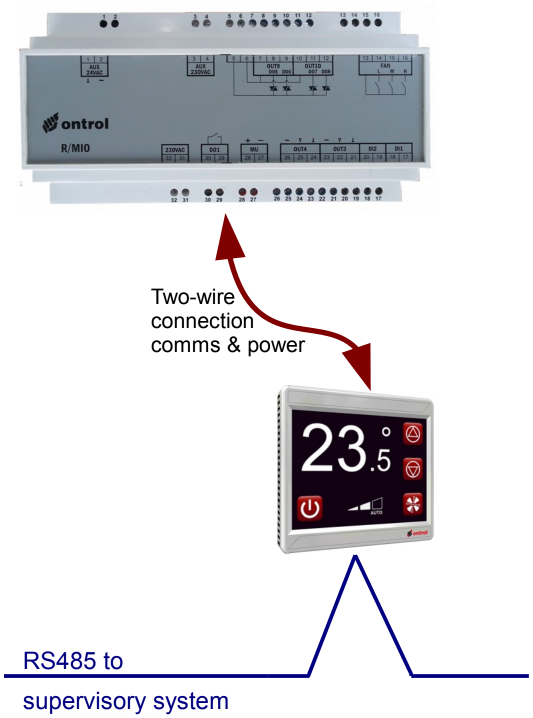
Option 1

RION with a R/TIO module

Option 1 : RION WITH A R/TIO MODULE

Requirements

R/TIO input output module[1]



How-to

Simply add a TimeDate component from the ontrolDeviceBus kit to your sedona app.

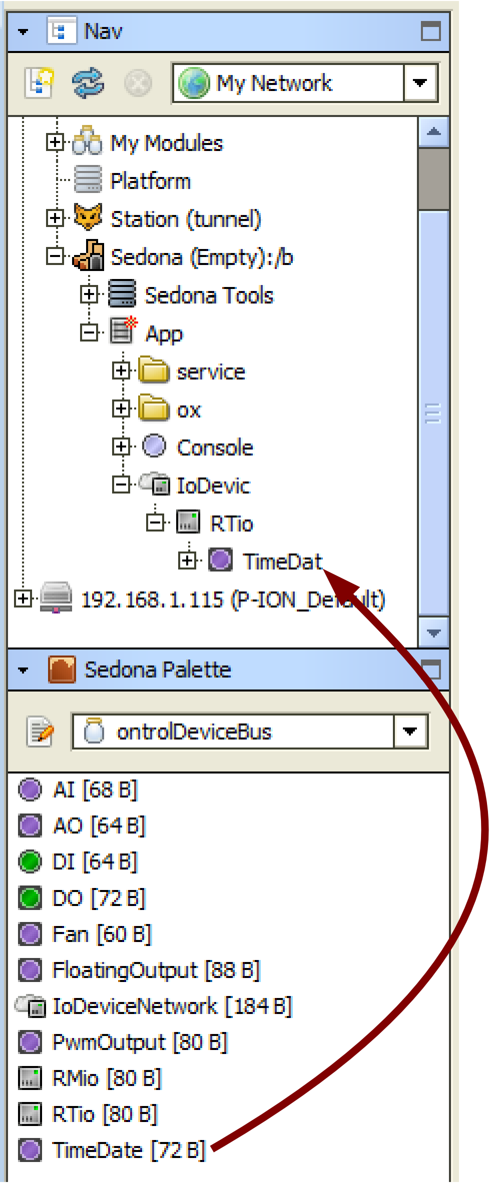
(See application note AN017 *Using dedicated IO modules with the R-ION* for details)

That is all!

IMPORTANT TIP:
Remember to also enable automatic summer/winter time change! See page 5

RECOMMENDED:
Whenever possible, configure additional synchronization from a master time keeping device. See following pages.

[1] R/TIO is a dedicated input/output module that works on a one-to-one connection with the R-ION.



Nav

My Network

- My Modules
- Platform
- Station (tunnel)
- Sedona (Empty):/b
  - Sedona Tools
  - App
    - service
    - ox
    - Console
    - IoDevic
      - RTio
        - TimeDat
- 192.168.1.115 (P-ION_Default)

Sedona Palette

ontrolDeviceBus

- AI [68 B]
- AO [64 B]
- DI [64 B]
- DO [72 B]
- Fan [60 B]
- FloatingOutput [88 B]
- IoDeviceNetwork [184 B]
- PwmOutput [80 B]
- RMio [80 B]
- RTio [80 B]
- TimeDate [72 B]

Time Synchronization : Option 2

STEP

3

Configure sync from an external source

Option 2

IP Based Sedona device connected to Niagara host

Option 2: IP BASED SEDONA DEVICE CONNECTED TO NIAGARA HOST

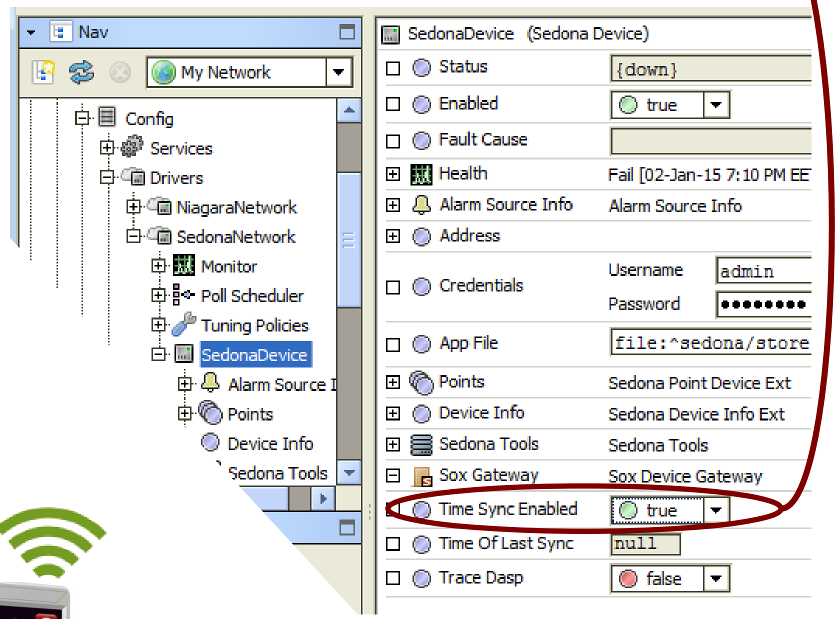
Requirements

- IP based Sedona devices (WIFI version of the R-ION)
- Niagara host (jace or supervisor) running station with Sedona driver

How-to

Simply set the TimeSyncEnabled property of the SedonaDevice in the Niagara host.

That is all!



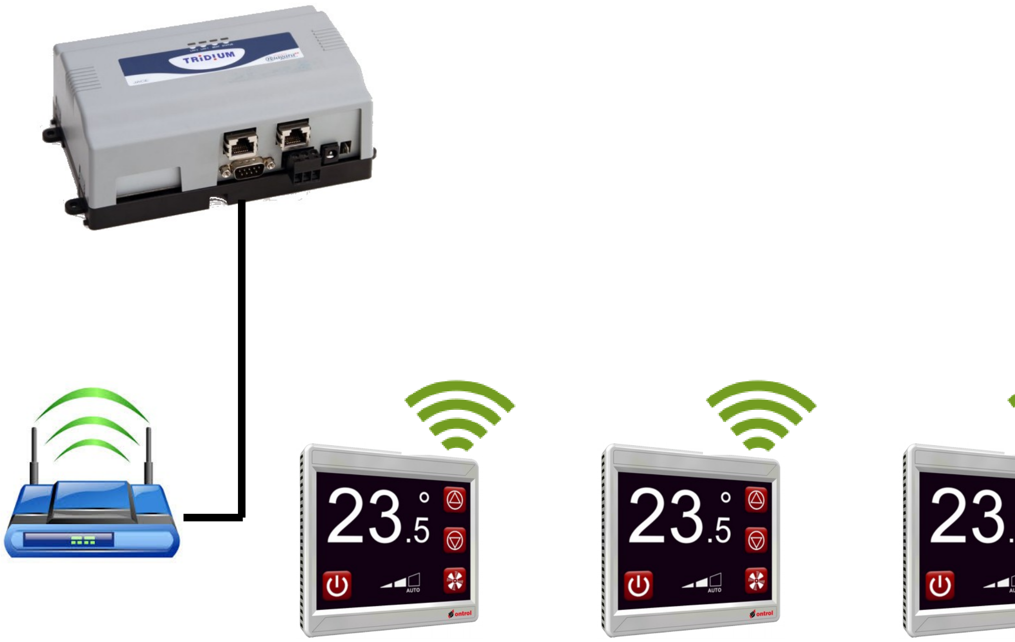
Nav

My Network

- Config
  - Services
  - Drivers
    - NiagaraNetwork
    - SedonaNetwork
      - Monitor
      - Poll Scheduler
      - Tuning Policies
      - SedonaDevice
        - Alarm Source I
        - Points
        - Device Info
        - Sedona Tools

SedonaDevice (Sedona Device)

□	Status	{down}
□	Enabled	true
□	Fault Cause	
⊞	Health	Fail [02-Jan-15 7:10 PM EE
⊞	Alarm Source Info	Alarm Source Info
⊞	Address	
□	Credentials	Username admin Password ●●●●●●●●
□	App File	file:^sedona/store
⊞	Points	Sedona Point Device Ext
⊞	Device Info	Sedona Device Info Ext
⊞	Sedona Tools	Sedona Tools
⊟	Sox Gateway	Sox Device Gateway
□	Time Sync Enabled	true
□	Time Of Last Sync	null
□	Trace Dasp	false



IMPORTANT TIP:
Remember to also enable automatic summer/winter time change! See page 5

STEP
3

Configure sync from an external source

Option 3

Sedona device as a modbus slave

Option 3 : SEDONA DEVICE AS A MODBUS SLAVE (Overview)

Requirements

- Sedona device configured as modbus slave using OntrolModbusSlaveSmart kit
- A modbus master device with a real-time-clock and programmable logic

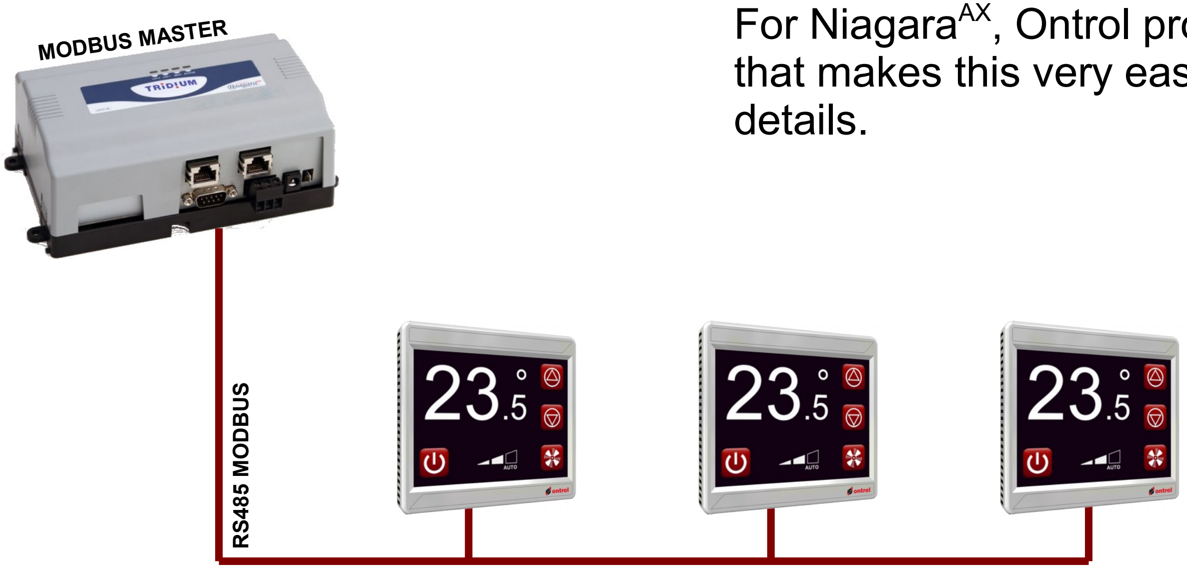
How-to

Configure your modbus master to write six integer values to properties of the ExternalDateTimeService component:

Hour – Minute – Second – Year – Month – Day

The modbus master must be configured to execute a “write multiple registers” command (16), so that all values are sent together and simultaneously.

For Niagara^{AX}, Ontrol provides a custom component that makes this very easy. See the next pages for details.



Time Synchronization : Option 3 (cont'd)

STEP

3

Configure sync from an external source

Option 3

Sedona device as a modbus slave

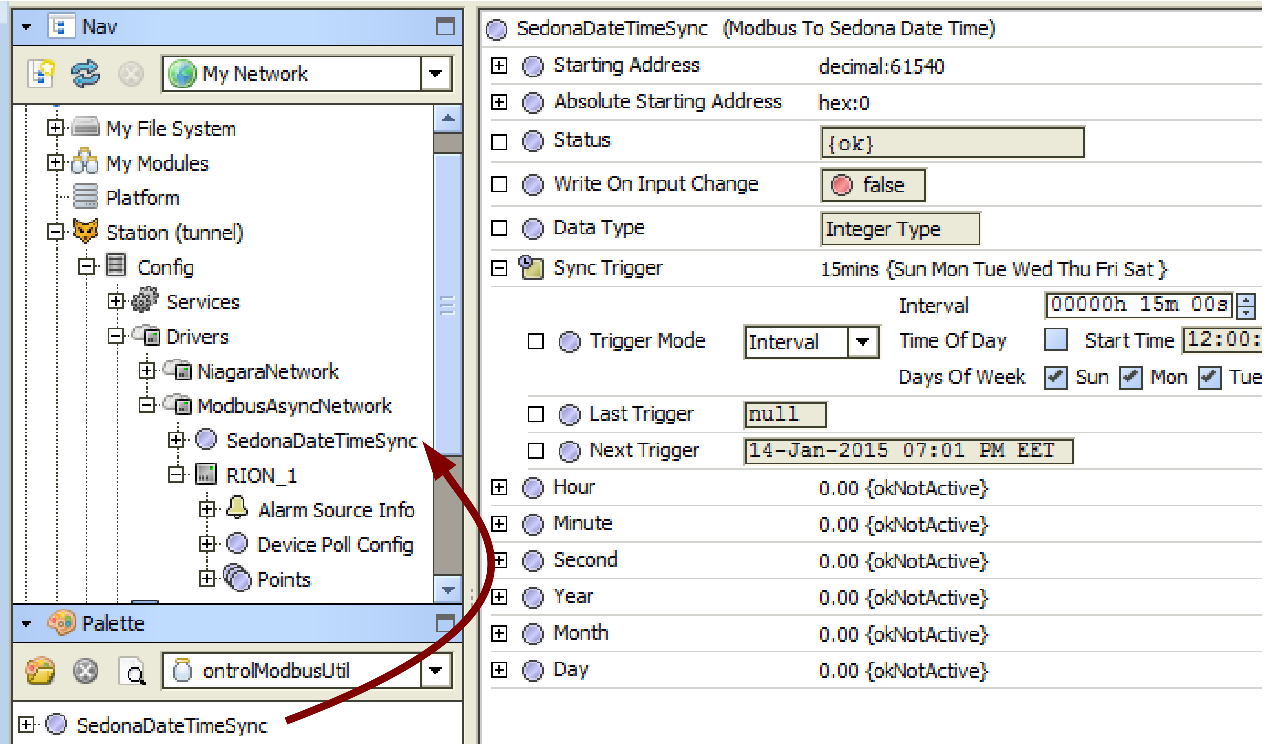
Option 3 : SEDONA DEVICE AS A MODBUS SLAVE
Niagara^{AX} side configuration

Simply add a SedonaDateTimeSync component from the ontrolModbusUtil module to your modbus network.

With no further settings, this component will ensure time syncronization to your Sedona device.

If this component is under the Points folder of a device, it will execute a time sync to that device only.

If it is under the ModbusNetwork directly, it will send a broadcast message to all devices on the network.



Nav

My Network

- My File System
- My Modules
- Platform
- Station (tunnel)
  - Config
    - Services
    - Drivers
      - NiagaraNetwork
      - ModbusAsyncNetwork
        - SedonaDateTimeSync
        - RION_1
          - Alarm Source Info
          - Device Poll Config
          - Points

Palette

ontrolModbusUtil

- SedonaDateTimeSync

SedonaDateTimeSync (Modbus To Sedona Date Time)

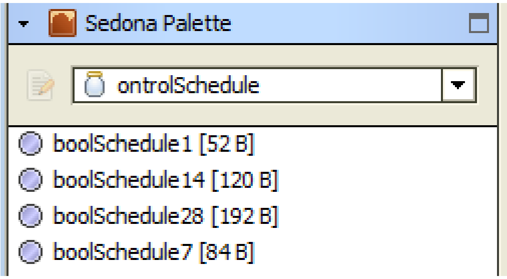
Starting Address	decimal:61540
Absolute Starting Address	hex:0
Status	{ok}
Write On Input Change	false
Data Type	Integer Type
Sync Trigger	15mins {Sun Mon Tue Wed Thu Fri Sat }
Trigger Mode	Interval
Interval	00000h 15m 00s
Time Of Day	□ Start Time 12:00:
Days Of Week	☑ Sun ☑ Mon ☑ Tue
Last Trigger	null
Next Trigger	14-Jan-2015 07:01 PM EET
Hour	0.00 {okNotActive}
Minute	0.00 {okNotActive}
Second	0.00 {okNotActive}
Year	0.00 {okNotActive}
Month	0.00 {okNotActive}
Day	0.00 {okNotActive}

Implementing schedules

STEP
4

Implement schedules

ontrolSchedule kit provides time schedule components with 1, 7, 14, or 28 periods per week:



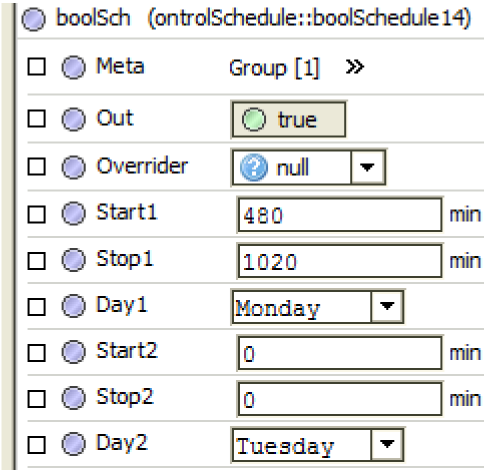
Sedona Palette

ontrolSchedule

- boolSchedule1 [52 B]
- boolSchedule14 [120 B]
- boolSchedule28 [192 B]
- boolSchedule7 [84 B]

These work on any sedona device with any kind of date/time service - including, of course, the **externaDateTimeService** described in this document.

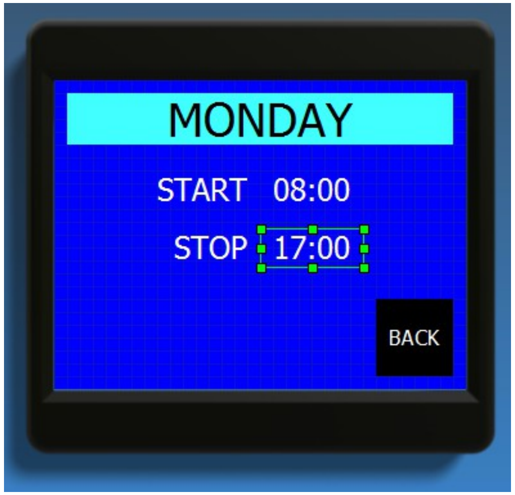
For each period, start and stop times are internally defined as minutes-after-midnight:



boolSch (ontrolSchedule::boolSchedule14)

□	Meta	Group [1] »	
□	Out	true	
□	Overrider	null	
□	Start1	480	min
□	Stop1	1020	min
□	Day1	Monday	
□	Start2	0	min
□	Stop2	0	min
□	Day2	Tuesday	

These slots can be associated with TimeLabelSet widgets to display *and* set them in a more familiar style :



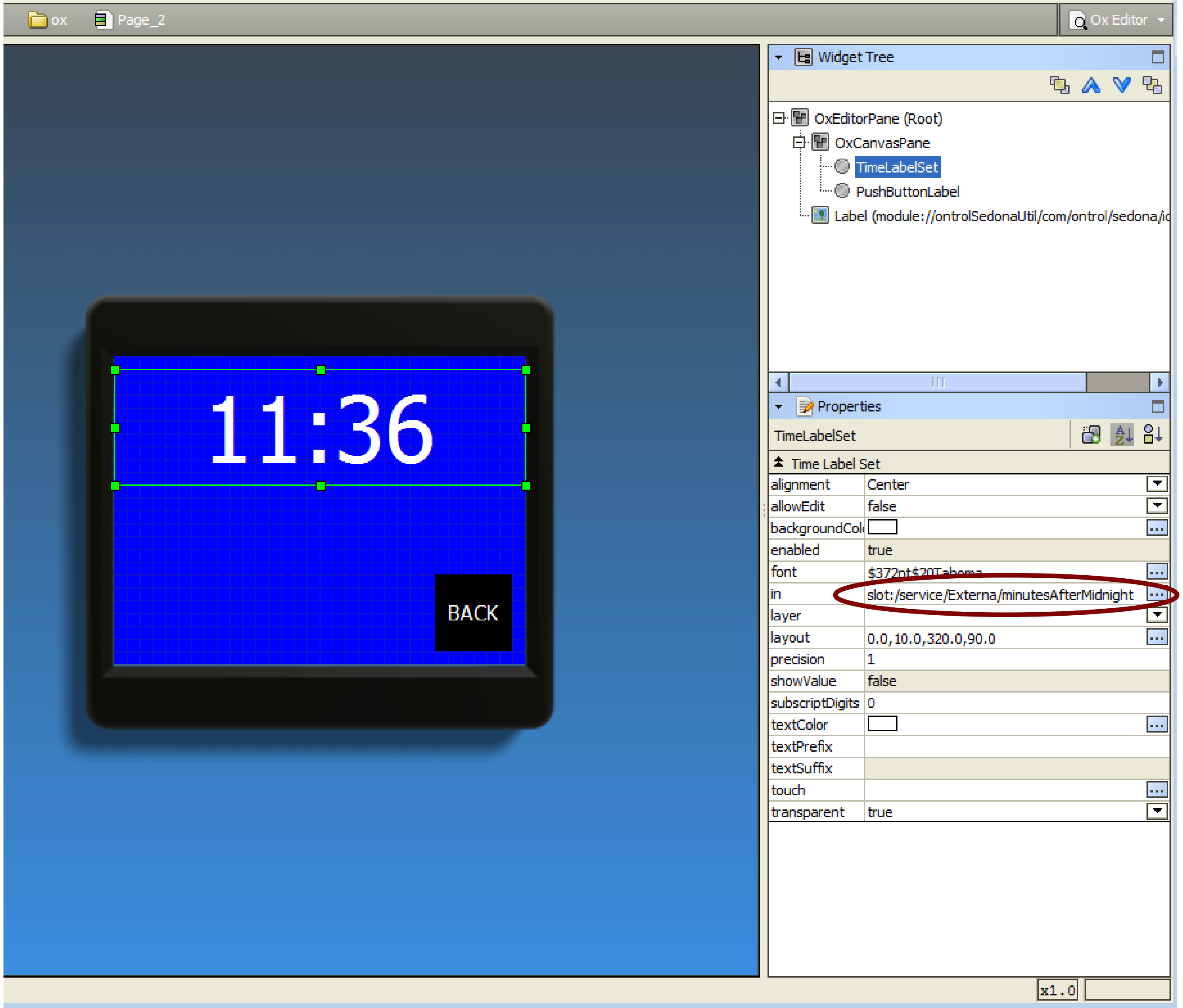
MONDAY

START 08:00

STOP 17:00

Each schedule component has a boolean 'out' slot that will be true when the actual time is within one of the set periods. This can be linked to logic to command equipment on/off.

Time display on the R-ION



TimeLabelSet

Time Label Set

alignment	Center
allowEdit	false
backgroundCol	□
enabled	true
font	$372pt$20Tahoma
in	slot:/service/Externa/minutesAfterMidnight
layer	
layout	0.0,10.0,320.0,90.0
precision	1
showValue	false
subscriptDigits	0
textColor	□
textPrefix	
textSuffix	
touch	
transparent	true

External DateTimeService has a property named 'MinutesAfterMidnight'

This can be associated with a TimeLabelSet widget on the R-ION to display actual time.