O3 Edge 2.1

EnOcean Application Guide

Edition 1.0

Delta CONTROLS

A Delta Group Company

Document edition: 1.0

Contents

Introduction

This guide describes how to connect an O3 Edge EnOcean® device to EnOcean, using BACnet for configuration.

The information in this document applies to O3 firmware version 2.1 and enteliWEB 4.19 for the following O3 Edge models: O3-EDGE-02-E8 and O3-EDGE-02-E9.

EnOcean Basics

EnOcean is a wireless protocol for ultra-low power energy harvesting sensors, switches, and controls. EnOcean devices offer batteryless, maintenance-free operation and come with standardized communication profiles to ensure interoperability across manufacturers.

EnOcean devices use short, infrequent radio signals and can transmit over distances of up to 30 m (100 ft) indoors. The signals are received and processed by line-powered devices such as relay switch actuators, repeaters, or gateways.

For more information about the EnOcean wireless standard, see https://www.enocean.com and https://www.enocean-alliance.org

About O3 Edge EnOcean

O3-EDGE-02-E8 and O3-EDGE-02-E9 devices are embedded with EnOcean gateway, connecting wireless EnOcean protocol to the O3 Edge, which can then be used to configure data through its connected interfaces (BACnet, MQTT, etc). The communication protocol is defined by EnOcean Serial Protocol 3.0.

The embedded gateway acts as both a receiver and a transmitter, reading from EnOcean input devices and writing to EnOcean output devices. Structured View (SV) objects represent the EnOcean devices on the network, with device data mapped to BACnet values according to a supplied XML mapping file. Normally these mappings are based on an EnOcean Equipment Profile (EEP), but custom mappings are also supported.

The gateway currently supports the following EEP types:

- **F6**: Repeated Switch Communication (RPS) Telegram, e.g., F6-02-01
- **D5**: 1 Byte Communication (1BS) Telegram, e.g., D5-00-01
- **A5**: 4 Byte Communication (4BS) Telegram, e.g., A5-20-01

For more information, see http://tools.enocean-alliance.org/EEPViewer.

Communication is established between the O3 Edge EnOcean device and an EnOcean device through a pairing process in which the device is put into learn mode and a "learn in" or "learn out" packet is sent from the gateway to the device.

O3 Edge EnOcean Specifications

The following table lists the specifications of the O3 Edge EnOcean-compatible device in O3 firmware version 2.1:

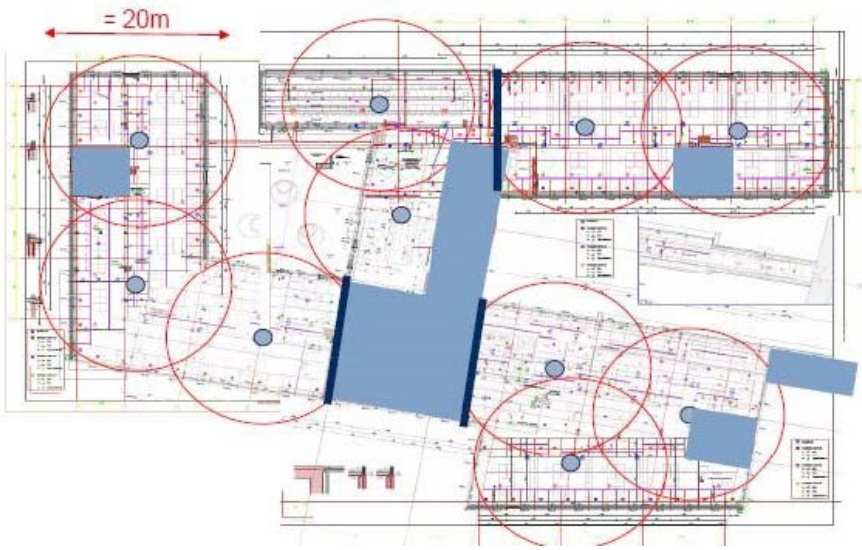
Gateway Capacity	• Up to 32 devices per O3 Edge with device numbers in the range 01-32. Device SV instances in the 33-99 range are not supported. • Recommend less than 13 points per EnOcean device.
EnOcean Data Units	Frame packet (physical layer), subtelegram (data link layer), telegram (network layer).
EnOcean Equipment Profiles	27 FIL object EEPs are included in the firmware.
BACnet Objects	Analog Value, Binary Value, Character String Value
Auto Mapping/Discovery	• Supports auto learn feature for inputs • Supports one click learn out, three click learn out, custom 4BS, and 4BS Variant 3 learn out for outputs • Allows manual addressing
Supported EnOcean Devices	• 868 MHz (Europe) or 902 MHz (North America) EnOcean input (read) devices • 868 MHz (Europe) or 902 MHz (North America) EnOcean output (write) devices
EnOcean Communication Protocol	Supports ESP3 (EnOcean Serial Protocol 3.0). Does not support older ESP versions (e.g., ESP2).

Getting Started

Before installing EnOcean devices, conduct a site survey to assess optimal placement of sensors and receivers.

To conduct a site survey:

1. Obtain a floor plan of the space.
2. Mark the locations of fire protection walls, staircases, elevator shafts, restrooms, server rooms, and any other areas of the building that may block radio signals.
3. Using a compass, draw a circle around each proposed receiver location (where you plan to install an O3 Edge. The circle represents the effective radio signal range. Each circle should cover approximately 20 m (66 ft) and overlap as little as possible with other circles.



EnOcean sensors should be located within a 10 m (33 ft) radius of the receiver. Ideally, the wireless signal should have an unobstructed line-of-sight path to the receiver. It should not cross any of the marked areas. For more details, see https://www.enocean.com/en/support.

As sensors are installed, conduct field tests with a handheld radio signal strength meter at each receiver location and verify that the signal is properly received. In some cases, you may need to move the sensors, change antenna positions, or add repeaters to achieve the desired result. As each receiver location is verified, record the signal strength measurements on the floor plan.

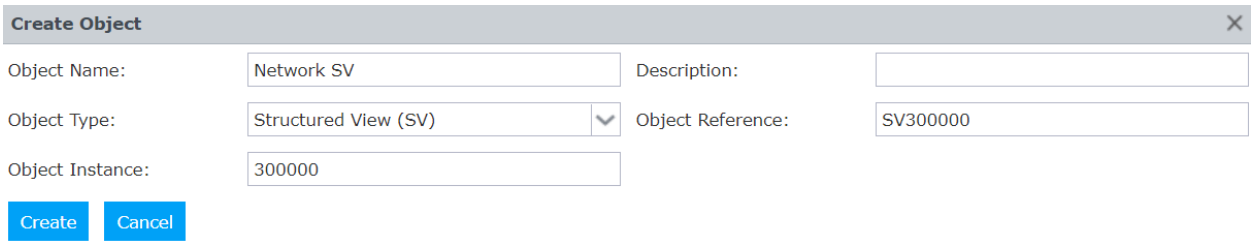
Configuring Input Devices

This section covers the following topics:

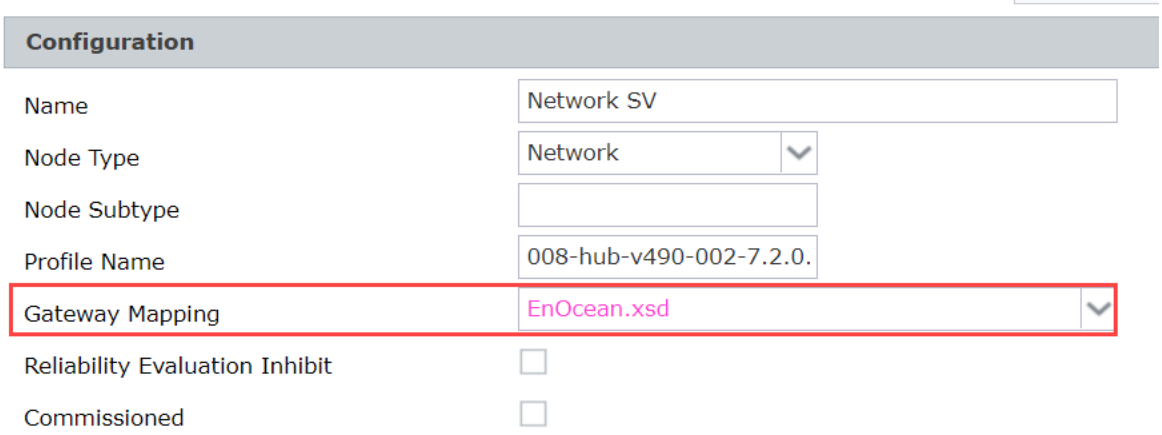
Configuring Input Devices Using an O3 Edge

To configure EnOcean input devices using an O3 Edge:

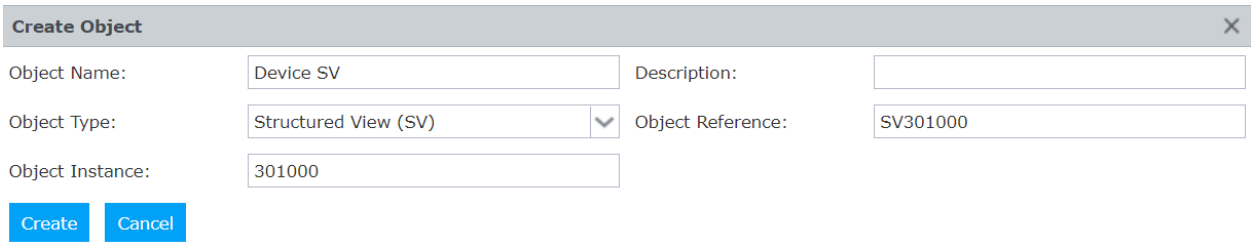
1. Create SV300000 in the device for the EnOcean network.



2. Once created, go to the device SV object and select **EnOcean** in Gateway Mapping, then click **Save**.

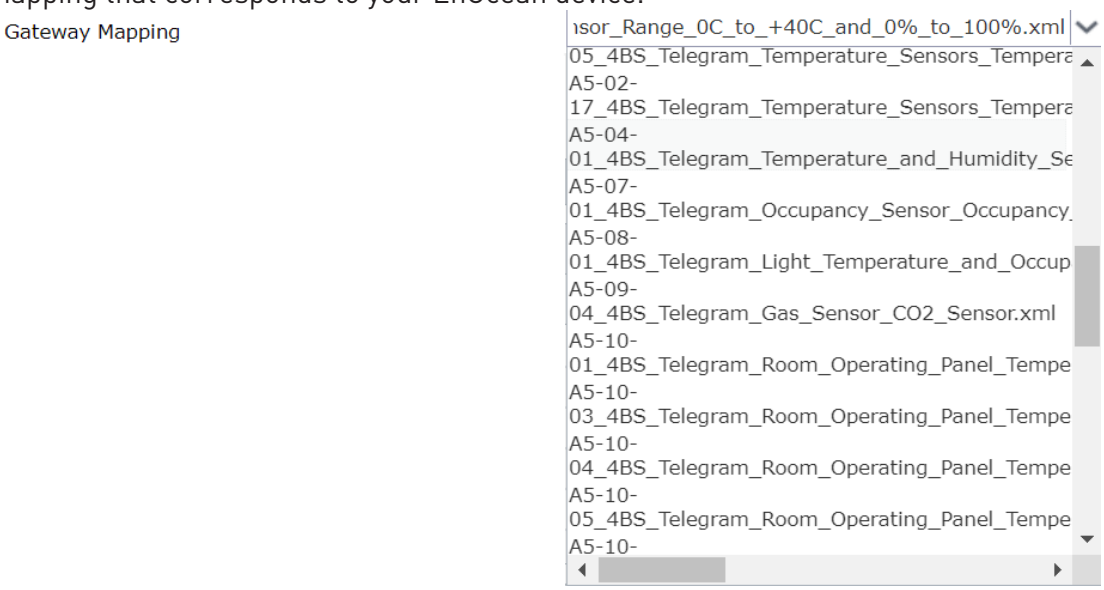


3. Create an SV object for each EnOcean input device (up to 32 per O3 Edge). The object instance follows the format 3DD000, where DD is the number of the EnOcean device in the range 01 to 32. For example, if the number you've assigned to the EnOcean device is 01, create SV301000; if you've assigned 02, create SV302000, and so on.

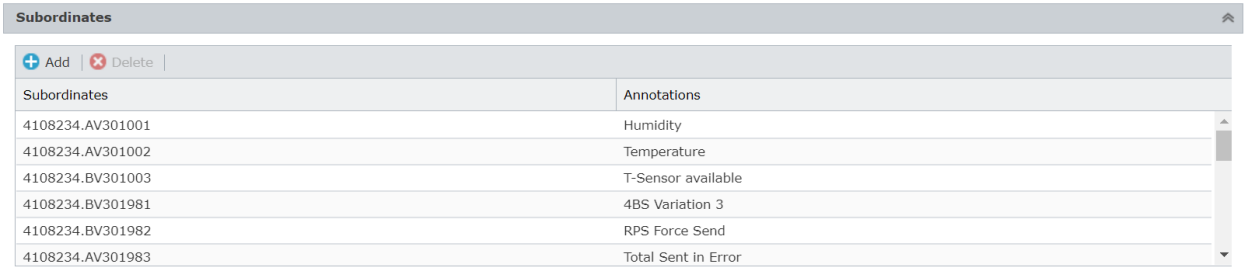


For O3 Edge device addresses, e.g., 301000, the first number ('3') is the fixed network address number. This number never changes. The following two numbers ('01') refer to the number, between the ranges 01-32, you arbitrarily assign to the EnOcean device to identify it. The last three digits ('000') refer to the parameters set in the EnOcean device.

4. Associate a mapping file with each device SV object.
   a. Open the device SV object.
   b. In Gateway Mapping, select the EnOcean Equipment Profile (EEP) or other XML mapping that corresponds to your EnOcean device.



   c. Click **Save**.
   The Subordinates section populates with a list of suggested BACnet object instances and names.



5. Create BACnet objects for each EnOcean device.
    a. Using the instances and names shown in the Subordinates section of the EnOcean device SV object, create the BACnet objects needed for your application. At a minimum, create the Learn Mode BV, Link Status CSV, and Device Address CSV objects.
    b. Verify that the created objects display as blue hyperlinks in the Subordinates section of the device SV object (clicking the link opens the object page).

The Learn Mode BV, Link Status CSV, and Device Address CSV objects follow the format 3DD9xx. For example, for the EEP profile F6-02-02, the Learn Mode object is 301998, the Link Status object is 301996, and the Device Address object is 301999.

Refer to the SV dialog to find the correct object instance to use for your profile.

6. Pair each EnOcean device to its device SV object.
    a. Command the Learn Mode BV object to **On**. The Link Status CSV object changes to "Pairing."
    b. If the EnOcean device has a Learn button, press the button to start pairing. When pairing is complete, the Device Address CSV object shows the device's ID and the Link Status CSV object changes to "Online." If the EnOcean device does not have a Learn button (for example, a rocker switch), activate the device (press the rocker switch) to allow the embedded gateway to learn the device ID. This should initiate pairing.
    c. Once the pairing is complete, command the Learn Mode BV object to **Off**.

If a FIL object associated with a Device SV object is changed or edited, a controller reboot is required to clear data from the previous profile.

Manually Pairing an EnOcean Device

If you already know the address of the EnOcean device you want to link to, you can simply open the Device Address CSV object and manually enter the hexadecimal address as the object value.

Resetting Pairing

To prevent re-pairing the same device, EnOcean learn packets received when the Device Address CSV has a non-zero value are ignored. If the O3 was previously linked to an EnOcean device in the Device Address CSV, then you must set the Device Address CSV object to 0, then put it back in Auto mode. The device is now ready to pair.

Configuring Output Devices

This section covers the following topics:

Main Steps for Configuring Output Devices

The initial steps for configuring an EnOcean output device are the same as for configuring an EnOcean input device. For detailed instructions regarding steps 1 to 4, refer to the previous section, Configuring Input Devices.

1. Create the network and device SV objects.
2. Associate a mapping file with each device SV object.
3. Create the BACnet objects needed for your application, either manually or using the Database Creation tool.
4. Pair each EnOcean device to its device SV object. The EnOcean output device is put into learn mode and a learn out packet is sent from the gateway to the device.

The remainder of this section will focus on steps 5 and 6.

Configuring Simple Relays

Simple relays provide basic on/off control and do not offer any feedback.

The mapping file should contain the following objects:

Name	Reference (O3 Edge)
Device Type	AV3DD993 (e.g., AV301993)
OFF Value	AV3DD990
ON Value	AV3DD991
Relay ONOFF	BV3DD001

DD = EnOcean device number (01–32)

The Device Type AV object is used for output control and defines two categories of devices: 0 = Input and output with feedback, and 1 = Output only. For a simple relay, the Device Type value is set to 1.

Depending on the vendor, required on and off levels may vary. The OFF Value and ON Value AV objects provide a flexible way to configure these values. These objects define the packet data value being sent out to command something on or off.

The following procedure assumes that you have already completed steps 1 to 4 in Main Steps for Configuring Output Devices.

To configure a simple relay:

1. Create the Device Type AV object and leave the value set to 0 to define an input or output with feedback, or set the value to 1 to define an output without feedback.
2. Create the OFF Value AV object and set the value. For example, 112(0x70).
3. Create the ON Value AV object and set the value. For example, 80(0x50H).
4. Create the Relay ONOFF BV object defined in the mapping file.
5. Create any other objects needed for your application.
6. Ensure that all Learn Out BV objects are **Off**.
7. Put the relay into Learn Mode. For specific instructions, consult the manufacturer's documentation.
8. Command the Learn Out BV object you are using (One Click Learn Out or Three Click Learn Out) to **On**.
9. After the relay has paired with the gateway, command the Learn Out BV object to **Off**.
10. Verify that the Relay ONOFF BV object controls the relay state.

RPS Force Send

Since simple relays do not include a feedback function, there is no indication of the relay's actual state. In some cases, you may want to send the same command again to ensure that the command was received and acted on correctly by the relay output device.

The RPS Force Send BV object (BV3DD982) provides a way to send the current relay output value even if that value has not changed.

RPS is an EnOcean term that refers to the repeated switch communication telegram type. EEPs that use RPS telegrams start with F6. For example, F6-02-01.

To use the RPS Force Send feature:

1. Verify that the OFF Value AV and ON Value AV objects are set to the required values.
2. Command the RPS Force Send BV object to On. This sends the current value of the Relay ONOFF Output BV object.

To use Force Send repeatedly, the Force Send BV object must be toggled ON and OFF at a rate slower than the program scan rate. Here are some sample programs that can help ensure that the ON/OFF signals are sent at the correct intervals.

The following pseudo code shows the idea of a state machine:

```
Variable Run As Integer
Variable SetpointOffset As Real
<other parts of the GCL, which sets "Run" and "SetpointOffset">

If (Run =0) Then

  If ( State > 1 ) Then
       BV = 0
       State = 0
  End If
Else
 DoEvery ( SetpointOffset )
  State = State + 1
  If (State = 1) Then
       BV = 1
  Else
       BV = 0
       State = 0
  End If
 End Do
End If

<the rest of the GCL+ program>
```

In the preceding code, "Run" is a GCL+ integer variable that indicates that RPS Force Send should be set periodically when it is non-zero. "SetpointOffset" is a GCL+ variable that indicates the value of the desired RPS Force Send interval. "State" represents the AV object that retains its value between scans. "BV" represents the RPS Force Send BV object.

You will need to create the following additional objects:

Name	Reference	Description
PG1	PG1	Master program to call PG2 and PG3
PG2	PG2	Program to send ON followed by Release
PG3	PG3	Program to send OFF followed by Release
Rocker ON and release switch	BV1	For PG2
Rocker OFF and release switch	BV2	For PG3
Rocker ON state variable	AV1	For PG2
Rocker OFF state variable	AV2	For PG3

PG1 Code: Calls the Two Programs

`Call PG2`

`Call PG3`

PG2 Example Code: Sends the ON Followed by a Release Packet

```
Variable ON_VALUE As Integer
Variable RELEASE_VALUE As Integer

ON_VALUE = 80
RELEASE_VALUE = 0


If ('Rocker ON and release switch' = 0) Then
       If ('Rocker ON state variable' > 7) Then
               'RPS Force Send' = Off
               'Rocker ON state variable' = 0
       End If
Else
       DoEvery 0.1S
               'Rocker ON state variable' = 'Rocker ON state variable' + 1
       If ('Rocker ON state variable' = 1) Then
               'ON Value' = ON_VALUE
```

```
                'RPS Force Send' = Off
        ElseIf ('Rocker ON state variable' = 2) Then
// send the ON packet here
        If ('RPS Send .Manual_Override' = Off) Then
                'RPS Send .Manual_Override' = On
        Else
                'RPS Force Send' = On
        End If
        ElseIf  ('Rocker ON state variable' = 3) Then
        'RPS Force Send' = Off
        ElseIf  ('Rocker ON state variable' = 4) Then
        'ON Value' = RELEASE_VALUE
        ElseIf  ('Rocker ON state variable' = 5) Then
// send the release packet here
        'RPS Force Send' = On
        ElseIf  ('Rocker ON state variable' = 6) Then
                'RPS Force Send' = Off
        Else
                'ON Value' = ON_VALUE
                'Rocker ON state variable' = 0
                'Rocker ON and release switch.Manual_Override' = 0
        End If
        End Do
End If
```

PG3 Example Code: Sends the OFF Followed by a Release Packet

```
Variable OFF_VALUE As Integer
Variable RELEASE_VALUE As Integer

OFF_VALUE = 112
RELEASE_VALUE = 0

If ('Rocker OFF and release switch' = 0) Then
        If ('Rocker OFF state variable' > 7) Then
                'RPS Force Send' = Off
```

```
                'Rocker OFF state variable' = 0
        End If
Else
        DoEvery 0.1S
                'Rocker OFF state variable' = 'Rocker OFF state variable' + 1
        If ('Rocker OFF state variable' = 1) Then
                'OFF Value' = OFF_VALUE
                'RPS Force Send' = Off
        ElseIf  ('Rocker OFF state variable' = 2) Then
// send the ON packet here
        If ('RPS Send .Manual_Override' = On) Then
                'RPS Send .Manual_Override' = Off
        Else
                'RPS Force Send' = On
        End If
ElseIf  ('Rocker OFF state variable' = 3) Then
                'RPS Force Send' = Off
ElseIf  ('Rocker OFF state variable' = 4) Then
                'OFF Value' = RELEASE_VALUE
ElseIf  ('Rocker OFF state variable' = 5) Then
// send the release packet here
        'RPS Force Send' = On
ElseIf  ('Rocker OFF state variable' = 6) Then
        'RPS Force Send' = Off
Else
        'OFF Value' = OFF_VALUE
        'Rocker OFF state variable' = 0
        'Rocker OFF and release switch.Manual_Override' = 0
                End If
        End Do
End If
```

Configuring Load Controllers with Central Command Support

Load controllers are typically used in lighting and HVAC circuit applications and usually include a feedback signal.

Many EnOcean output devices offer central command support (A5-38-08). This feature is typically used in combination with another profile. For example, the gateway can use the A5-38-

08 lighting dimming profile and the A5-11-01 lighting controller status feedback profile together to control the output device.

Download A5-38-08+A5-11-01 from https://support.deltacontrols.com/Products/EnOceanEEPs

Currently, the O3 Edge only supports the switching (0x01) and dimming (0x02) command bytes for A5-38-08.

Be extra careful when setting up the packet data. The gateway has the following behaviors for out-of-range or unexpected data values:

- The command byte can be set to an out-of-range value.
- Other values can be set out of range. For example, the Dimming Value AV object for Command 0x02 has a range of 0 to 255 but can be set to a value greater than 255. If you set to a value greater than 255, the gateway will overflow. Since the device has no way to know that it has overflowed, it will use the overflow value as its data.
- Negative values that are out of range are set to 0 by the gateway. A zero in the packet data indicates an error.

The combined A5-38-08 + A5-11-01 mapping file should contain following objects:

Name	Reference (O3 Edge)
Command	AV3DD001 (e.g. AV301001)
Dimming Value	AV3DD002
Ramping Time	AV3DD003
Dimming Range Relative Value	BV3DD004
Store Final Value Yes	BV3DD005
Switching Command ON	BV3DD006
Illumination	AV3DD007
Illumination Set Point	AV3DD008
Dimming Output Level	AV3DD009
Repeater Enabled	BV3DD010
Power Relay Timer Enabled	BV3DD011
Daylight Harvesting Enabled	BV3DD012
Dimming Load	BV3DD013
Magnet Contact Closed	BV3DD014

Name	Reference (O3 Edge)
Occupancy Occupied	BV3DD015
Power Relay ON	BV3DD016
ONOFF	BV3DD017
Device Type	AV3DD993
Link Status	CSV3DD996
Reliability Timer	AV3DD997
Learn Mode	BV3DD998
4BS Send	BV3DD985
4BS Learn Out	BV3DD986
Learn Out Func	CSV3DD987
Learn Out Type	CSV3DD988
Learn Mode	BV3DD998
Device Address	CSV3DD999

DD = EnOcean device number (01–32)

The following procedure assumes that you have already completed steps 1 to 4 in Main Steps for Configuring Output Devices and that you've uploaded an XML mapping file that contains the combined profile A5-38-08+A5-11-01.

To configure a load controller with central command support:

1. Create the mapping and configuration objects needed for your application.
2. Set the Device Type AV object value to 2 to define an output with feedback.
3. Ensure that all Learn Out BV objects are **Off**.
4. Set the value of the Learn Out Func CSV object to 38.
5. Set the value of the Learn Out Type CSV object to 08.
6. Put the load controller into learn mode. For specific instructions, consult the manufacturer's documentation.
7. Command the 4BS Learn Out BV object to **On**.
8. After the load controller has paired with the gateway, command the 4BS Learn Out BV object to **Off**.
9. Set the Command AV object to a value allowed by the profile. For example, to control dimming, set the value to 2.

10. Command the 4BS Send BV object to **On** to send a control packet. Confirm that the load controller responds correctly when you send a command.

Setting Up Feedback

If the output device supports feedback, you can set up feedback to show the response to an output command.

To set up feedback:

1. Command the Learn Mode BV object to **On**. The Link Status CSV object changes to "Pairing."
2. On the EnOcean output device, generate a learn packet from the device using the method specified by the manufacturer. When pairing is complete, the Device Address CSV object shows the device's ID and the Link Status CSV object changes to "Online."
3. Once pairing is complete, command the Learn Mode BV object to **Off**.
4. Confirm that the feedback value matches the actual output value or state.

Configuring Radiator Valve Controllers

A radiator valve controller is an actuator for a heating radiator application that uses the A5 20 04 EEP profile (FIL1227).

The mapping file should contain the following objects:

Name	Reference (O3 Edge)
Temperature	AV3DD001 (e.g., AV301001)
4BS Variation 3	BV3DD981
RPS Force Send	BV3DD982
Total Send Error	AV3DD983
Total Send	AV3DD984
4BS Send	BV3DD985
4BS Learn Out	BV3DD986
Learn Out Func	CSV3DD987
Learn Out Type	CSV3DD988
Three Click Learn Out	BV3DD989

Name	Reference (O3 Edge)
OFF Value	AV3DD990
ON Value	AV3DD991
One Click Learn Out	BV3DD992
Device Type	AV3DD993
Total Receive Error	AV3DD994
Total Receive	AV3DD995
Link Status	CSV3DD996
Reliability Timer	AV3DD997
Learn Mode	BV3DD998
Device Address	CSV3DD999

DD = EnOcean device number (01–32)

The following procedure assumes that you have already completed steps 1 to 4 in Main Steps for Configuring Output Devices and that you've associated the A5-20-04 profile with the device SV object.

To configure a radiator valve controller:

1. Create the mapping and configuration objects needed for your application.
2. Command the 4BS Variation 3 BV object to **On**.
3. Ensure that all Learn Out BV objects are **Off**.
4. Set the value of the Learn Out Func CSV object to 20.
5. Set the value of the Learn Out Type CSV object to 04.
6. Command the Learn Mode BV object to **On**.
7. Put the radiator valve controller into learn mode. For specific instructions, consult the manufacturer's documentation.
8. Command the 4BS Learn Out BV object to **On**. The Device Address CSV object should show the device's ID.
9. Command the 4BS Learn Out BV object and the Learn Mode BV object to Off.
10. Confirm that the radiator valve controller responds correctly when you send a command.

To control a radiator valve controller:

1. Ensure that the Learn Mode BV object is **Off**.
2. Set the control parameters based on the A5-20-04 profile.
3. Set a value for an object such as the valve position.

The wake-up cycle is typically at least 10 minutes for a radiator valve application, so you can expect a significant delay before a new valve position is received and acted on.

Monitoring Network Communication

This section covers the following topics:

Understanding the Link Status

Link Status has four status values that indicate the current stages of the EnOcean device in the linking process:

- **Unpaired**: The status of Unpaired indicates that the device has not set up a link with an EnOcean device. For manual pairing, the status changes from Unpaired to Online when the first packet is received.
- **Pairing**: Indicates that learning is enabled. When learn mode is enabled, the link process gets the new device address, and sets up the link.
- **Online**: Indicates a successful learn operation with a valid address.
- **Offline**: The address CSV holds a valid address, but the reliability timer is expired. An Offline status indicates that a device was learned but no new packet came in during the defined reliability timer period.

The Link Status requires up to one minute to reflect a new status.

When the Learn Mode BV object is commanded to On, the Link Status object changes to Pairing. When the first packet is received, the Link Status changes to Online. When the pairing process is complete, the Device Address CSV shows a hexadecimal value.

Understanding the Reliability Timer

The Reliability Timer confirms that the device is active and communicating.

Name	Reference (O3 Edge)
Reliability Timer	AV3DD997

DD = EnOcean device number (01–32)

The Reliability Timer object displays a value in seconds (although the unit displays as %). When a new packet is received, the timer resets and begins another count down cycle. If a new packet is not received from the paired EnOcean device before the timer expires, the Link Status CSV

object changes to Offline. To disable the Reliability Timer for a device, manually set the Reliability Timer object value to 0.

Monitoring Communication Statistics

Incoming and outgoing communications statistics provide a way to verify that EnOcean data is being transferred without an excessive number of errors.

Name	Reference (O3 Edge)
Total Send Error	AV3DD983
Total Send	AV3DD984
Total Receive Error	AV3DD994
Total Receive	AV3DD995

DD = EnOcean device number (01–32)

The ratio of error packets to total packets received should be less than 5 percent.

The Total Receive object holds the count for total packets received with no errors. The Total Receive Error object holds the count for total packets received with errors.

The Total Send object holds the count for total packets sent with no errors. This number is reported by the EnOcean transceiver and does not necessarily mean that the packets were received at their destination. The Total Send Error object holds the count for total packets sent with errors. The Total Send count does not include packets that cause the Total Send Error count to be incremented.

You can create an AV object (AV3DD059) to monitor the total number of packets received by a particular EnOcean-capable O3 Edge. However, this will also count packets received with errors, so it is not a good indicator of the total number of successful packets.

Working With Range and Scale Values

This section covers the following topics:

- Understanding Range and Scale Values
- Using a BV Object to Show Off/On States

Understanding Range and Scale Values

EnOcean device values are scaled according to the minRange, maxRange, minScale, and maxScale configurations specified in the XML mapping file.

For example, here is the XML for FIL1204:

```
<?xml version="1.0" encoding="utf-8"?>
<si:mapping


xmlns:si="http://schema.deltacontrols.com/schemas/entelibus/SpecialIn
terface1.0"



xmlns:eno="http://schema.deltacontrols.com/schemas/entelibus/enOceanM
apping"
type="enocean">
       <si:variable>
               <si:name>Supply voltage (OPTIONAL)</si:name>
               <si:type>AV</si:type>
               <si:advalue>
                       <eno:analogValue>
                              <eno:ioNumber>1</eno:ioNumber>
                              <eno:offset>0</eno:offset>
                              <eno:size>8</eno:size>
                              <eno:minRange>0</eno:minRange>
                              <eno:maxRange>250</eno:maxRange>
                              <eno:minScale>0</eno:minScale>
                              <eno:maxScale>5.0</eno:maxScale>
                              <eno:value>0</eno:value>
```

```
                    </eno:analogValue>
                </si:advalue>
            </si:variable>
```

In the above code, the raw value range of 0–250 corresponds to engineering values of 0–5 VDC.

The minRange may sometimes be greater than the maxRange, as in the following example:

```
<eno:minRange>255</eno:minRange>
<eno:maxRange>0</eno:maxRange>
<eno:minScale>-40</eno:minScale>
<eno:maxScale>0</eno:maxScale>
```

In the above code, the raw value 255 corresponds to -40°C (-40°F) and the raw value 0 corresponds to 0°C (32°F).

You can edit the minRange, maxRange, minScale, and maxScale values in the Translation table in the FIL object. See Working with Mapping Files.

The following example shows two mapping rows: the first shows a 0°C to 40°C (32°F to 104°F) temperature range and the second shows a -40°C to 0°C (-40°F to 32°C) temperature range.

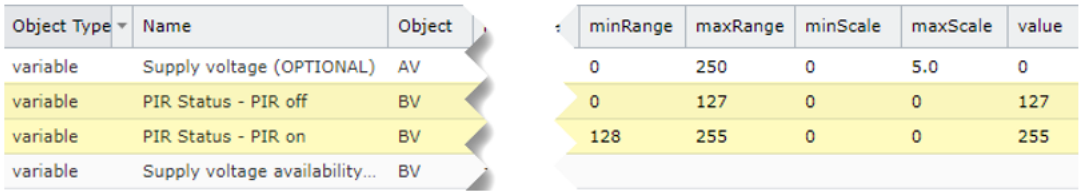
ioNumber	offset	size	minRange	maxRange	minScale	maxScale
1	16	8	255	0	0	+40
2	16	8	255	0	-40	0

For an AV mapping, the gateway does not support minScale and maxScale both having a range value of 0. If you do not want scales for an AV, match the scale values to the range values.

Using a BV Object to Show Off/On States

You can use a BV object to show the simple Off and On states of a device.

The EEP profile uses a scale and range match to determine the current state of the device. For example, in the A5-07-01 (FIL1204) profile for motion/occupancy sensors, a raw PIR value in the range 0 to 127 corresponds to OFF (unoccupied), while a raw PIR value in the range 128 to 255 corresponds to ON (occupied).



Object Type	Name	Object	minRange	maxRange	minScale	maxScale	value
variable	Supply voltage (OPTIONAL)	AV	0	250	0	5.0	0
variable	PIR Status - PIR off	BV	0	127	0	0	127
variable	PIR Status - PIR on	BV	128	255	0	0	255
variable	Supply voltage availability...	BV					

You could create the following objects to represent the device's PIR value:

Name	Reference (O3 Edge)
PIR Status - Off	BV3DD002 (e.g., BV301002)
PIR Status - On	BV3DD003 (e.g., BV301003)

A value greater than 128 would indicate that the room is occupied, while a value less than or equal to 128 would indicate that the room is unoccupied.

Working With Mapping Files

This section covers the following topics:

- Modifying a Default Mapping File
- Creating a Custom XML Mapping File
- Saving Mapping Data

The EEP profiles included in the default database (FIL1201–1227) should be used whenever possible. Additional tested profiles may be available on support.deltacontrols.com. You can also contact Delta's Professional Development Services group to have a custom mapping file made.

Modifying a Default Mapping File

Occasionally, you may want to modify one of the default EEP mapping files. For example, you may want to adjust the range and scale values.

To modify a default mapping file:

1. Open the FIL object you want to modify (e.g., FIL1215) and click **Download File**. The XML file (A5-10-10_4BS_Telegram_Room_Operating_Panel_Temperature_and_Humidity _ Sensor_Set_Point_and_Occupancy_Control.xml) is saved to the Downloads folder.
2. Create a new FIL object (e.g., FIL115).
3. Open the new FIL object (e.g., FIL115) and click **Upload File**. Upload the XML file you downloaded in step 1.
4. Modify the mapping values in the Translation table as needed.

- To edit a value, double-click the value you want to edit, enter the new value, then click **Save**.
- To add a new mapping row, click **Add**, double-click each cell and enter the desired value, then click **Save**.
- To copy a mapping row, select the row, then click **Copy**. The row is added to the end of table.
- To delete a mapping row, select the row, then click **Delete**.

If no values appear in the Translation table, the uploaded XML file may contain invalid syntax or unsupported options.

If a FIL object associated with a Device SV object is changed or edited, a controller reboot is required to clear data from the previous profile.

Translation Table Details

The following table describes each column in the Translation table and notes which values are not supported.

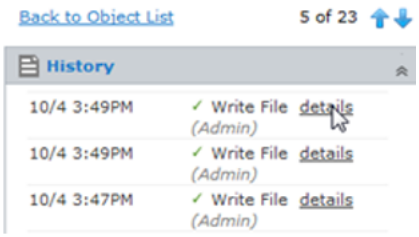
Column	Values
Object Type	variable input, output (not supported)
Name	Name of the BACnet object. Can be up to 47 characters including spaces.
Object	AB, BV AI, BI, MI, MV, AO, BO, MO, LO, CSV (not supported)
Mapping	advalue complex, hoa-switch, feedback (not supported)
Mapping Type	analogValue, binaryValue
Direction	get = read (for EnOcean inputs) set = write (for EnOcean outputs) [*blank*] = read/write
ioNumber	Instance number of the BACnet object.
offset	The bit offset in the data packet, where an offset of '0' is the most significant bit of the first byte in the data packet. For example, offset 0 and size 3 in a byte that has a value of 0x55 (0x01010101) means a value of 0000010. The three most significant bits are selected (010).
size	Number of bits that make up the object value. The size starts at the most significant bit of the first byte. For AV objects, the size can be up to 32 bits. For BV objects, the size is always 1.
minRange	Lowest possible raw value.

Column	Values
maxRange	Highest possible raw value.
minScale	Engineering value that corresponds to the minRange value.
maxScale	Engineering value that corresponds to the maxRange value.
value	The minRange, maxRange, minScale, and maxScale elements provide a way to scale this raw value into engineering units. This field is mandatory. Put in "0" here unless instructed to do otherwise.

Differences Between Current and Previous FIL Object Mappings

The History pane in enteliWEB provides a convenient way to see changes made to a FIL object.

To see the differences between the current and previous version of a FIL object, in the History pane at the right of the enteliWEB page, click the details link.



A window opens showing the differences between the current and previous version of the file. Scroll down to see recent changes.

Creating a Custom XML Mapping File

While it is recommended to use the predefined EEP mapping files included in the default database (FIL1201–1227), you can also create your own XML mapping files that follow the syntax rules defined in the Special Interface and EnOcean XML schema files (FIL1001 and FIL1012). You will need an XML editor for this, such as Microsoft Visual Studio.

To create a mapping file using an XML editor:

1. Download SpecialInterface.xsd (FIL1001) and EnOcean.xsd (FIL1012) to your PC.
2. Open your XML editor and create a new XML document.
3. Open the XML document and add the following lines to set up the XML document file to refer to the two schema files:

```
<?xml version="1.0" encoding="utf-8"?>
<si:mapping


xmlns:si="http://schema.deltacontrols.com/schemas/entelibus/SpecialInterface1.0"

xmlns:eno="http://schema.deltacontrols.com/schemas/entelibus/enOceanMapping"
type="enocean">

</si:mapping>
```

4. If necessary, assign the schema files to the XML document using the XML editor. For example, Visual Studio allows you to assign schema files that help it validate the XML document.
5. Add mappings to the XML document (e.g., ioNumber, offset, size, minRange, etc.). The value number and ioNumber (object instance) are mandatory.
6. Save the XML document to the same folder as the schema files.

Instances (ioNumber) must follow sequentially for the mapping translation to work. If non-sequential instance numbers are used, the FIL interface will accept them but the mappings will not work correctly.

Saving Mapping Data

If you do an enteliWEB database save, the .zdd file does not include SV objects or XML mapping content. To save mapping data, you must use the **Download File** button on the FIL page.

Appendix A: Troubleshooting

This section covers the following topics:

Troubleshooting an O3 Edge

The O3 Edge is not communicating

- Check that the O3 Edge (the model name must end in "E8" or "E9") is communicating over the correct frequency (868 MHz for O3-EDGE-02-E8 models, 902 MHz for O3-EDGE-02-E9 models).

Communications are unreliable

- Check the overall send and receive packet counts. If necessary, create the Total Receive, Total Receive Error, Total Send, and Total Send Error objects.
- Use a packet sniffer such as DolphinView to detect devices broadcasting excess packets.
- Check for changes in the physical environment. For example, a new concrete or metal wall can affect communications adversely. Be aware of potential noise sources such as lighting ballasts or electrical equipment that can interfere with EnOcean signal strength.
- Modify the environment to improve communications. If this is not possible, consider moving the O3 Edge or installing a repeater.

Troubleshooting EnOcean Objects

EnOcean object values are not updating

- Check that the network SV and device SV objects have been created with the correct instance numbers.
- Check that the device SV object is using the correct XML mapping file.
- Check that the EnOcean objects (AVs, BVs, CSVs) have been created with the correct instances. An active hyperlink appears in the device SV Subordinates column when the object is created correctly.

The Device Address CSV object is not learning a new address

- Manually set the Device Address CSV object value to 0 to clear any previous addresses.

EnOcean object values are lost when a controller is shut down

- When an O3 Edge is shut down for a short time, events from COV-only input sensors (like window or door contacts) that were triggered during the shutdown period may be lost. These device types are unable to resynchronize after power up. The only way to resynchronize is to trigger a packet from the input sensor.

Appendix B: FIL EEP Profiles

Firmware version 2.1 includes 27 predefined FIL object EEP profiles.

Many EnOcean devices use EEP profiles that are not included in the predefined FIL objects. These untested devices may or may not work with the gateway. Additional tested profiles may be available on support.deltacontrols.com/Products/EnOceanEEPs.

The following table contains a list of the FIL EEP profiles that are included with EnOcean compatible devices.

EEP	FIL	Application	Device Example
A5-02-05	1201	Temperature sensor	Echoflex MT-17 Echoflex RTS-1T-UW sensors Thermoken SR65 Thermoken SRC-D0
A5-02-17	1202	Outdoor temp sensor	Thermokon SR65
A5-04-01	1203	Temp and humidity sensors	Echoflex RTS-1H-UW sensors Thermokon SR04-RH
A5-07-01	1204	Occupancy sensor	ILLUMRA E3x-x12GP Echoflex ERM-DL Thermokon SRC-D0 (Heating and/or cooling application)
A5-08-01	1205	Light, temp, humidity and occupancy sensor	Thermokon SRC-D0 (Heating and/or cooling, humistat, and lighting control)
A5-09-04	1206	CO2 sensor	Thermokon SR04 C02
A5-10-01	1207	Temp, SP, FS, OCC	Thermokon SR04PST
A5-10-03	1208	Temp and SP sensors	Echoflex RTS-1TS-UW
A5-10-04	1209	Temp, setpoint, fan speed	Thermokon SR04PS

EEP	FIL	Application	Device Example
A5-10-05	1210	Temp, setpoint, occupancy	Thermokon SR04PT
A5-10-06	1211	Temp, setpoint & override button	Thermokon SR04P-MS
A5-10-0A	1212	Deprecated	
A5-10-0B	1213	Deprecated	
A5-10-0C	1214	Deprecated	
A5-10-10	1215	Temp, humidity, setpoint, occupancy	Thermokon SR04PT-RH
A5-10-11	1216	Temp, humidity, override button	Thermokon SR04P-MS-rH
A5-10-12	1217	Temp, humidity, setpoint	Thermokon SR04P-RH
A5-10-19	1218	Temp, setpoint, humidity, fan speed, occupancy	Echoflex RTS-2HS-UW sensors
A5-30-02	1219	Window contact switch	Digital input single input contact Echoflex MC-21
D5-00-01	1220	Window contact switch (single input contact)	Thermokon SRW01 Thermokon SRC-D0 Echoflex ERM series ILLUMRA E3x-x12GP
F6-02-01	1221	Light and blind control (EU application style)	Thermokon 4 channel switches Echoflex ETRH2(*)-W ILLUMRA E3x-x12GP

EEP	FIL	Application	Device Example
F6-02-02	1222	Light and blind control (NA application style)	Echoflex PTM265D(*) Echoflex ERM series Echoflex ETRH2(*)-W ILLUMRA E3x-x12GP
F6-04-01	1223	Key card activated switch	Echoflex PTM265KCAU Echoflex ERM series ILLUMRA E3x-x12GP
A5-20-01	1224	Battery-powered actuator (BI-DIR)	Micropelt MVA002/003/004 Thermokon SAB+
A5-20-02	1225	Basic actuator (BI-DIR)	
A5-20-03	1226	Line-powered actuator (BI-DIR)	Spartan ME83xx
A5-20-04	1227	Heating radiator valve actuator with feed temp, room temp, position, setpoint button, display (BI-DIR)	HORA SmartDrive

Revision History

Edition	Date	Description
1.0	October 2021	New document.