Green.Smart.Wireless.
enocean®

EnOcean Link in the Cloud

COMBINING ENOCEAN LINK WITH A WEBSERVICE

Table of contents

1. INTRODUCTION

This document describes the usage of EnOcean Link in a web service application. EnOcean Link is a middleware library and interface to the EnOcean network. Any customer Gateway or Software which deals with EnOcean based products can embed the library and benefit from EnOcean Link services. EnOcean Link converts the bits and bytes of an EnOcean telegram directly into data values. In doing so, the middleware automatically takes into account all specifications of the EnOcean protocol stack and the EnOcean Equipment Profiles (EEPs) [2] and Generic Profiles (GP) [3] of the EnOcean Alliance as well as encryption mechanisms. This means EnOcean Link provides all wireless applications with a ready-made key to the energy harvesting wireless world. Since the software interprets all data, it also ensures the interoperability of equipment from different manufacturers. Please refer to reference [1] for more details on EnOcean Link.

EnOcean Link is a software project. Target hardware are external micro controllers – not provided by EnOcean. It is intended for embedded devices that are processing EnOcean telegrams provided by an ESP 3 (EnOcean Serial Protocol 3) [5] EnOcean Gateway (i.e. USB 300 or TCM 310) [6], [7]. Basic scenario is in the figure below.
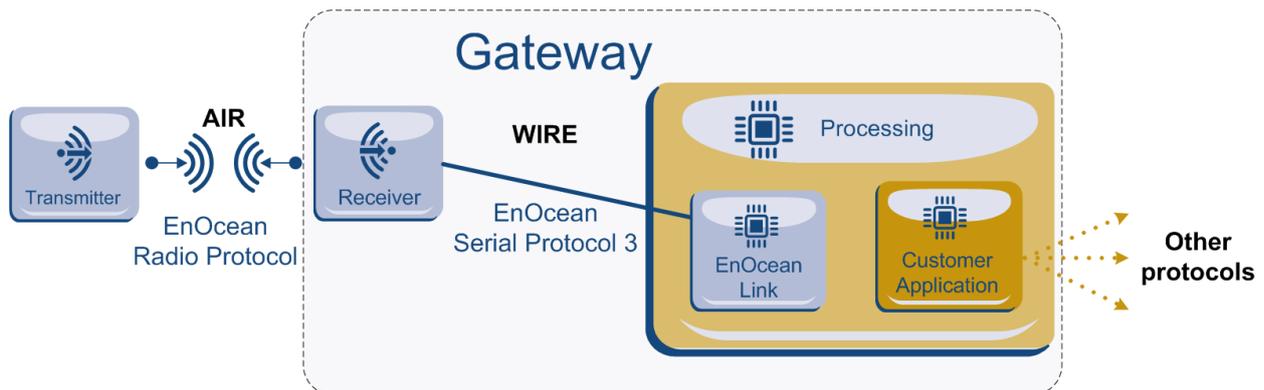

Figure 1 EnOcean Link in a Gateway

In this document we focused on a different use case, in which EnOcean Link runs separate in the cloud as web service and the processing unit / home gateway access EnOcean Links services through the Web. Please refer to the figure below:
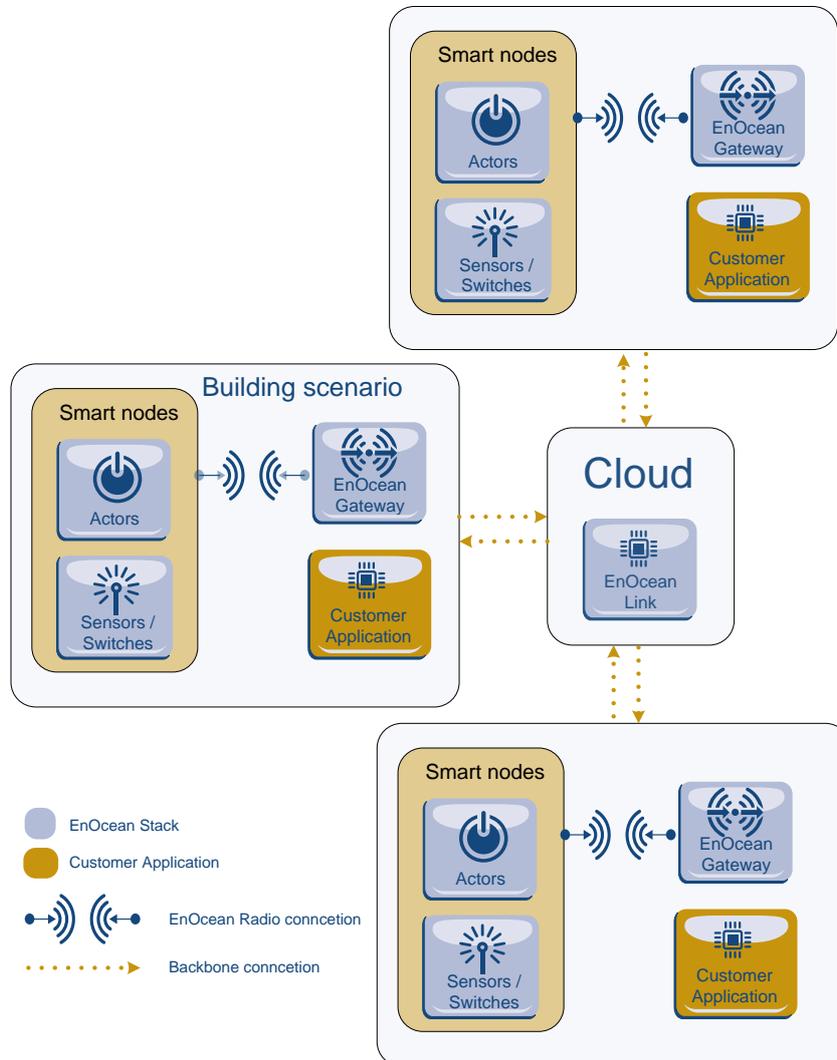
Figure 2 EnOcean Link in the Cloud

There can be a huge quantity of Building scenarios / separate installations. All of them have an EnOcean Gateway and an Application running. The EnOcean Gateway and the application can be placed together onto a separate hardware. They are using the Internet communication to put request to EnOcean Link in the cloud to interpret their received telegrams from Smart Nodes. As return they get from EnOcean Link the actual data and can continue with their application. This way the Home Gateway stays as simple and cost effective as possible, because the main EnOcean specific processing part is executed by EnOcean Link in the cloud.

The actual web application is up and running and can be used for laboratory use on "as-is" basis. It is useful for smaller demonstrators or as a starting point for your own web service development. Its main purpose is to demonstrate the huge potential of EnOcean Link and the use of it in the cloud. Please be inspired by it.

The EnOcean Link Webservice is running on a detached server outside EnOcean. On the same server a web interface (internet site) for the Web service is running. This web inter-

face is useful if you need to check your data, see a demonstration of EnOcean Link func-tionality or just want to have an impression of what EnOcean Link is.

The EnOcean Link Webservice is also available via a Web page. You can link to this service also from your pages. Just mention it is run by EnOcean Link. As reference implementation you can even use the included HTML files and scripts.

Please visit this link to access the web interface:

http://85.214.110.59/

This document describes:

■ How to use the EnOcean Link Webservice trough any Website or from your applica-tion

■ How to build EnOcean Link Webservice on your own

1.1. Definitions

Term / Abbr.	Description
µC	Microcontroller (external)
AES	Advanced Encryption Standard
API	Application Programming Interface
APP	Application
ASK	Amplitude Shift Keying
CBC	Cipher Block Chaining
CMAC	Cipher Based Message Authentication Code
CRC	Cyclic Redundancy Codes
DATA	Payload of a radio telegram
Device	Customer end-device with an integrated EnOcean radio module
EEP	EnOcean Equipment Profile
EHW	Energy Harvested Wireless protocol
ERP	EnOcean Radio Protocol (ERP1 = Version 1, ERP2 = Version 2)
ESP3	EnOcean Serial Protocol V3
FSK	Frequency Shift Keying
Gateway	Module with a bidirectional serial communication connected to a HOST
GP	Generic Profiles
ID	Unique module identification number
KEY	Specific parameter used to encrypt / decrypt / transform DATA
MAC	Message Authentication Code
MSB	Most Significant Byte
PSK	Pre-shared Key
PTM	Pushbutton Transmitter Module
RLC	Rolling Code
R-ORG	Message parameter identifying the message type
SLF	Security Level Format specifying which security parameters are used
TXID	ID of a transmitter
VAES	Variable AES

1.2. References

[1] EnOcean Link - http://www.enocean.com/en/enocean-software/enocean-link/

[2] EEP Specification - http://www.enocean-alliance.org/eep/

[3] GP Specification - http://www.enocean-alliance.org/de/enocean_standard/

[4] EnOcean Radio Protocol 1 - http://www.enocean.com/fileadmin/redaktion/pdf/tec_docs/EnOceanRadioProtocol.pdf

[5] ESP 3 - http://www.enocean.com/esp

[6] Gateway Controller - http://www.enocean.com/en/enocean-software/gateway-controller/

[7] Dolphin V4 Gateway Controller http://www.enocean.com/en/enocean-software/

[8] HTerm - http://www.der-hammer.info/terminal/

1.3. Revision History

No	Major Changes
1.0.	First version

2. ENOCEAN LINK WEBSERVICE

The EnOcean Link Webservice is a service which acts like EnOcean Link via HTTP-Requests. The received telegram from the EnOcean Gateway are forwarded to the EnOcean Link service in the cloud for further analysis. The format of the received data can be exact the same as the EnOcean gateway provided it (ESP3), but for demonstration purposes we also included other formats.

Currently, it supports:

- ESP3 [5]-
- ERP1-data [4]
- manually entered data.

For ESP3- or ERP1-requests, it can analyze the telegrams which are supplied and return the complete parsed telegram structure (e.g. RORG, DATA, Source-ID etc).

If the forwarded telegram is a Teach-In request then EnOcean Link Webservice stores the ID and EEP for further development. After a limited time period the Database is erased.

If a Telegram is being forwarded and the ID and EEP are known, then EnOcean Link Webservice interprets the given data according to the chosen profile. The interpreted data are then returned.

An example is given in the following text. For an example of the Web interface please refer to chapter 3.

2.1. Specification

The EnOcean Link Webservice can be accessed at http://85.214.110.59 under port 8080. Therefore, the base URL for every request is http://85.214.110.59:8080. The EnOcean Link Webservice supports:

- HTTP-GET
- HTTP-POST

For both requests, the parameters are given in the well-known format for key-value-pairs: key0=value0&key1=value1 etc.

For a GET request, the parameters are appended after a '?' at the URL, for a POST request, the parameters are given in the HTTP-BODY (please refer to the Tutorial of usinG EnOcean Link Webservice section where you can see an example of how to use jQuery to create requests with JavaScript).

There are several kinds of requests, each requires different parameters and supports different return values. These requests are:

- **Packet-Request:** For ESP3- and ERP1-data format. It is called Packet-Request because all provided data is given in a single packet of hex-data. The provided telegram can also be a teach-in telegram. The ESP3- and ERP1-data format is defined in the EnOcean standard.

- **Manual-Teach-In-Request**: Manually define a device in the database. You provide a Device-id as well as the EEP [2] (RORG, FUNC, TYPE) and this entry is then stored. This learns a device into the system.

- **Data-Request**: You provide a Device-id as well as payload hex-data. The data will be interpreted according to the profile specified in the teach-in procedure.

Please note that for data interpretation with Packet Request or Data Request the Device-ID and the profile (EEP) must be known by EnOcean Link Webservice. This can be done by manual Teach-in or by providing a Teach-In telegram via Packet Request.

Currently supports EnOcean Link Webservice RPS, 1BS and 4BS-telegrams which it was tested with. Following Profiles are supported:

- F6-02-XX, F6-03-XX, F6-04-XX, F6-10-XX
- A5-02-XX, A5-04-XX, A5-06-XX, A5-07-XX, A5-10-XX, A5-12-XX
- D5-00-01

2.2. Parameters of Requests

The necessary parameters for GET and POST request are as follows:

2.2.1. Packet-Request

key	value
datatype	"esp3" or "erp1"
data	your esp3-packet resp. your erp1-telegram as a string of hex-characters without whitespaces, the length should be a multiple of 2 because every 2 characters represent 1 byte

2.2.2. Manual-Teach-In-Request

key	value
deviceId	the device-id of the device which will be teached in: 8 hex-characters (equals 4 bytes which is the standard length of a device id)
rorg	the rorg of your desired profile, 1 or 2 hex-characters
func	the func of your desired profile, 1 or 2 hex-characters
type	the type of your desired profile, 1 or 2 hex-characters

2.2.3. Data-Request

key	value
datatype	"data"
deviceId	the device-id of the device that was previously teached in: 8 hex-characters (equals 4 bytes which is the standard length of a device id)
data	the data to interpret: 2 hex-characters for RPS- or 1BS-profiles, 8 hex-characters for 4BS-profiles

Remember that every parameter given in an HTTP-Request will be automatically treated as string. Thus, the strings do not need to be quoted. For example, a correct query-string would look like this:

datatype=data&deviceId=12345678&data=0A.

2.3. Return-Values

The EnOcean Link Webservice uses JSON to return its results. The return value is always - in case of success as well as in case of an error - a single JSON-object. Depending on the type and the success of the request the JSON-object contains different fields.

2.3.1. Error case

In case of an error the JSON-object contains only one single field "error", the value of which is a string containing an error message.

2.3.2. Success case

In case of a successful interpretation the return field is coded in JSON Format. There are two basic structures to be considered:

- Telegram

  This structure is returned in case of Packet-Request.

- Profile

  This structure is returned in case of Manual-Teach-In-Requests, Data-Request (when the device is known) and Packet-Request (when the device is known or the provided telegram is an actual Teach-In telegram)

Telegram

The "telegram"-field contains a JSON-object with the following fields:

"rorg"	Number
"data"	String
"srcId"	Number
"status"	Number
"destinationId"	Number
"dBm"	Number
"subtelCount"	Number
"rorg"	Number

Profile

The "profile"-field contains an array of JSON-objects. Each of them represents a channel and contains the following fields:

"name"	String
"min"	Number
"max"	Number
"value"	Number
"unit"	String

Note that according to JSON the Number values can be integer or float. Be careful when printing these values because normally they will be printed as decimal values but sometimes you will rather print them as hex values, e.g. the RORG-field or Device-Ids.

3. TUTORIAL OF USING ENOCEAN LINK WEBSERVICE

A Web-Frontend for using the EnOcean Link Webservice can be found at http://85.214.110.59. The usage should be quite straightforward: On the left side you can choose which kind of request you want to send to the EnOcean Link Webservice, please refer to Figure 3. Then, the appropriate input-fields are shown and you can configure your request and send it to the EnOcean Link Webservice. The results will be shown below.

Figure 3 EnOcean Link Webservice Web Frontend

The front end pages can be downloaded also with the scripts, this way you can build it also into your own webpage.

3.1. HTTP-GET

Using the EnOcean Link Webservice is very simple. The easiest method to use it is by sending Http-GET requests via your browser which just means calling a specific URL. Please visit:

http://85.214.110.59/tutorial/

to see a URL Tutorial. There are several clickable URLs for you below. As a result, you can see the returned JSON. Please see Figure 4.

EnOcean Link Webservice - Tutorial

JavaScript Source Code

Send Request

Manual-Teach-In-Request	deviceId=FEFFFEBC, rorg=F6, func=02, type=01
Data-Request	datatype=data, deviceId=FEFFFEBC, data=30
ERP1-Request	datatype=erp1, data=F6 30 FE FF FE BC 30 00
ESP3-Request	datatype=esp3, data=55 00 07 07 01 7A F6 30 FE FF FE BC 30 01 FF FF FF FF 2D 00 C6

Result

no request sent

Figure 4 EnOcean Link Webservice Tutorial

The Content-Type of the returned JSON is "application/json". Therefore, some browsers do not show the results but offer you to save a file.

3.1.1. Manual-Teach-In-Request

We teach in a device with an arbitrary device-id, we use the following profile: `RORG=A5, FUNC=08, TYPE=03`. That is

- a 4BS-profile;
- Light, Temperature and Occupancy Sensor;
- Range 0lx to 1530lx, -30°C to +50°C and Occupancy Button.

http://85.214.110.59:8080?deviceId=12345678&rorg=A5&func=08&type=03

3.1.2. Data-Request

We send data which should be interpreted with our previously teached-in profile. The data should encode the following values:

- supply voltage 5.1V,
- illumination 1530lx,
- temperature 50°C,
- button: pressed.

http://85.214.110.59:8080?deviceId=12345678&data=FFFFFF0A

3.1.3. Packet-Request with ERP1

Now we send the same data as above, but packed into an ERP1-telegram.

http://85.214.110.59:8080?datatype=erp1&data=A5FFFFFF0A123456780000

3.2. HTTP-POST

Creating a HTTP-POST request is not as easy as typing an URL into the browser. An approach to create and send HTTP-POST requests is by using JavaScript with the jQuery-library for Ajax-Requests. The jQuery-library natively supports Cross-Origin-Resource-Sharing (CORS) which is necessary to send Ajax-Requests to a different server than the one which supplied the website. A very simple website is located at

http://85.214.110.59/tutorial.

The source code of the script can be found at

http://85.214.110.59/tutorial/request.js.

Or click on the direct link. It contains comments which explain the very basics used in this example.

3.3. Using HTerm

If you have a USB stick with EnOcean Gateway, you can use it to receive telegrams which will then be sent as ESP3-packets to your machine. Maybe you can use the tool HTerm to see these packets and use them for the EnOcean Link Webservice. Please see reference [8] for details and user conditions.

To connect HTerm to your USB EnOcean Gateway, select it under port and configure the Baud-rate to 57600, then click Connect. It is a good idea to change the output to Hex-format and configure the "Newline after … ms receive pause" to 2, because then the packets will be separated properly.

You can select a packet in HTerm and copy it as Hex and then paste it to the Webpage to send it to the EnOcean Link Webservice. Please refer to the following Pictures.
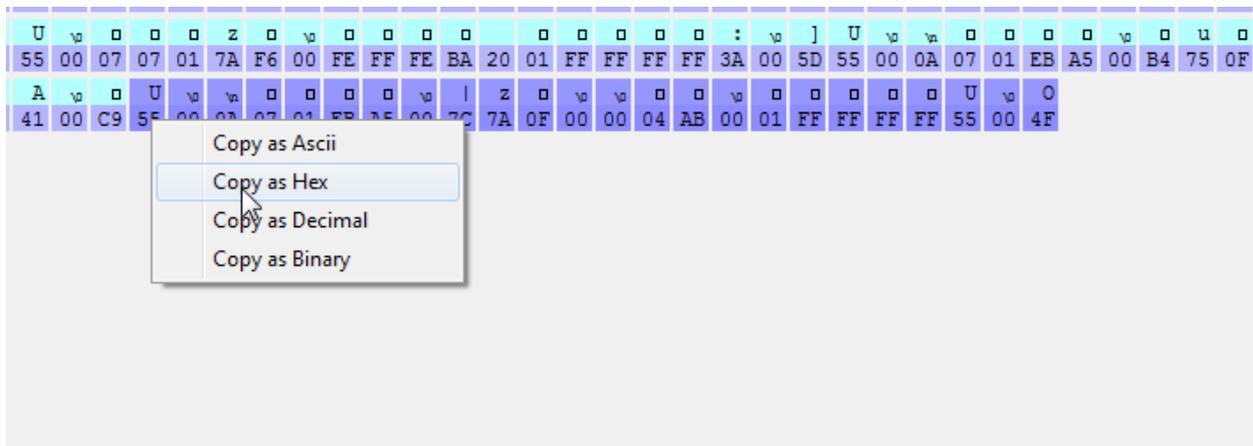

Figure 5 Copy the Telegram Input as Hex

Figure 6 Insert it and submit as ESP 3 Packet

4. HOW TO BUILD ENOCEAN LINK WEBSERVICE

Attached to this Application Note you will find the source code of the actual EnOcean Link Webservice which is running on detached server. There is further development needed to make it a serial product, be inspired by this tutorial to continue your development.

4.1. Directory structure

The folder www/ contains the whole website, including the tutorial.

EoWebservice/ contains the eclipse project, which can be copied as it is into the eclipse workspace and then be added via File -> Import -> Existing Projects into Workspace -> Select root directory (Browse). Then select the copied directory in the workspace.

EoWebserviceTest/ contains the eclipse project of EoWebserviceTest, which tests the functionality of EnOcean Link Webservice and in case of an error sends an email. Check the URL as well as the email-address in the source-code!

ubuntuRoot/ contains directories/files copied from the test-/development-machine. Maybe they can shorten the installation and configuration process. Assuming that ubuntuRoot corresponds to / , then the directory structure indicates where the files need to be located. See information below!

4.2. Tutorial

In the read me TXT file further description is located.

Disclaimer