

EnOcean Link with own profiles extensions ADD OWN PROFILE DEFINITIONS TO ENOCEAN LINK

1. INTRODUCTION

In EnOcean Link [1] are all implementations of all actual EEP [2] profiles included. This Application Note is aimed for use cases, where special profiles need to be added to EnOcean Link:

- Manufacturer Specific Profiles – MSC Telegrams [2]
- Profiles in development, to be submitted to EnOcean Alliance
- Other cases of extensions

After a profile is officially approved by the EnOcean Alliance then it gets included into EnOcean Link by next EnOcean Link release. This application note enables to test and benchmark the profile with EnOcean Link before this process.

In general EnOcean Link can be adjusted by the developer in any way, because the source code is part of the delivery. This application notes focuses on the details how to add / change profile implementations in EnOcean Link.

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
DoS	Denial of service
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

EnOcean Link with own profiles extensions
ADD OWN PROFILE DEFINITIONS TO ENOCEAN LINK

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] EnOcean Wireless Standard http://www.enocean-alliance.org/en/enocean_standard/
- [9] Security of EnOcean Radio Networks, <http://www.enocean.com/en/security-specification/>
- [10] EnOcean Application Notes, <http://www.enocean.com/en/application-notes/>

1.3. Revision History

No	Major Changes
1.0.	First version

EnOcean Link with own profiles extensions ADD OWN PROFILE DEFINITIONS TO ENOCEAN LINK

2. PROFILE DEFINITION IN ENOCEAN LINK

Under profile definition we understand:

- EnOcean Equipment profiles [2]
- Generic Profiles [3]

Generic profiles sensors use already a self-descriptive language, so every sensor is specific, but abstract enough to be understood by EnOcean Link without additional changes. Changes in Generic Profiles language itself is not trivial and therefore is not part of this application note.

2.1. EnOcean Equipment Profiles

In this Application Note we focus on extensions to EnOcean Equipment Profiles - EEP. EEP is represented by tree numbers:

- RORG
- FUNC
- TYPE

An EEP expresses how an incoming data telegrams should be parsed to get real application values. Bidirectional profiles define the outgoing structures as well. In EnOcean link are profiles represented by classes. The main class for EEP is `eoEEPProfile`

(http://www.enocean.com/fileadmin/redaktion/support/enocean-link/classeo_e_e_profile.html)

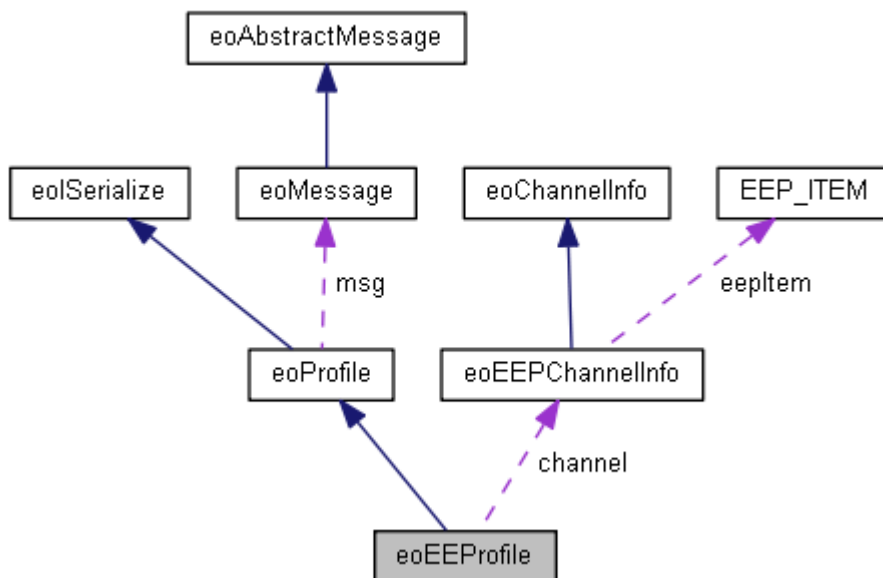


Figure 1 `eoEEPProfile` collaboration diagram

Profiles classes are then separated by their RORG. E.g.

- RORG = RPS - 0xF6 - `eoF6EEPProfile`

http://www.enocean.com/fileadmin/redaktion/support/enocean-link/classeo_f6_e_e_profile.html

EnOcean Link with own profiles extensions
 ADD OWN PROFILE DEFINITIONS TO ENOCEAN LINK

- RORG = 4BS - 0xA5 - eoA5EEPProfile
http://www.enocean.com/fileadmin/redaktion/support/enocean-TheLink/classeo_a5_e_e_profile.html
- RORG = VLD - 0xD2 - eoD2Profile
http://www.enocean.com/fileadmin/redaktion/support/enocean-link/classeo_d2_e_e_profile.html

Further are the profiles separated by their FUNC. FUNC E.g.

- RORG = 0xA5, FUNC = 0x05 - eoEEP_A505xx
http://www.enocean.com/fileadmin/redaktion/support/enocean-link/classeo_e_e_p_a505xx.html
- RORG = 0xA5, FUNC = 0x02 - eoEEP_A502xx
http://www.enocean.com/fileadmin/redaktion/support/enocean-link/classeo_e_e_p_a502xx.html

These classes are inherited from the above "RORG classes". The specific profile implementation is then realized in these classes. Please see hierarchy diagram for RORG = 0xD2.

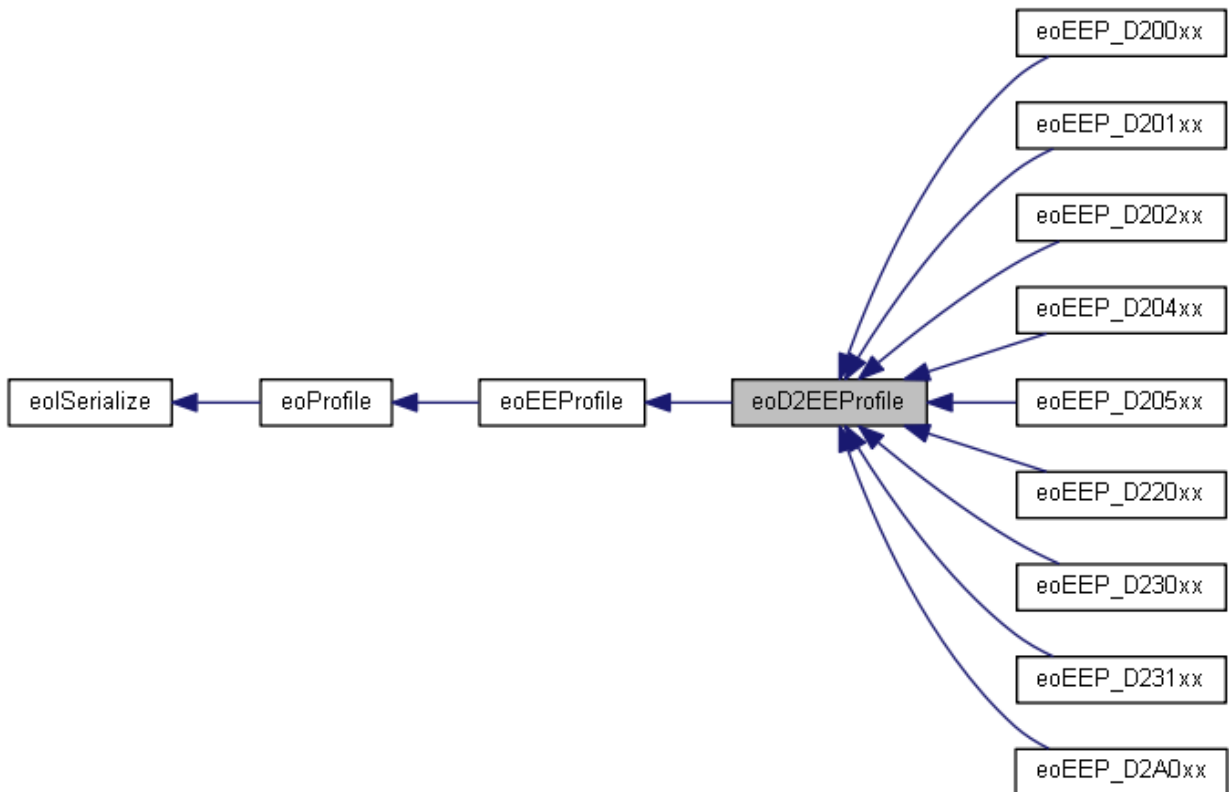


Figure 2 0xD2 Inheritance diagram

The `GetChannel()` function provides the real values from an received telegram. The channels are there parsed by a predefined matrix structure of parameters – e.g. bit offset, resolution, scaling, physical unit etc. One matrix entry is defined as:

EnOcean Link with own profiles extensions ADD OWN PROFILE DEFINITIONS TO ENOCEAN LINK

- `EEP_ITEM`
http://www.enocean.com/fileadmin/redaktion/support/enocean-link/structure_eep_item.html

With the Matrix of `EEP_ITEMS` ~70% of all EEPs can be represented and parsed. The Matrix strongly resembles the EEP XML file which is provided by the EnOcean Alliance [8].

Experience in profile parsing showed us that the remaining ~30% profiles have special profile and parsing logic which does not fit into the abstract Matrix structure. Therefore we enabled specific parsing routines for determined profiles to conform the definitions made by EnOcean Alliance members. The specific routines are represented by profile specific source code, which is included in the profile class.

Please consider that parsing can be based on:

- `EEP_ITEM` – common parsing routine, defined in the `eoEEPProfile` class
- Profile specific parsing source code – embedded in the respected class of the profile

2.2. Profile Factory

A specific profile instance is assigned to a device instance via pointer. The profile is created if an teach-in is received or by manual calling. Please see the manual for details:

- http://www.enocean.com/fileadmin/redaktion/support/enocean-link/eo_teach_in_module_8h.html
- http://www.enocean.com/fileadmin/redaktion/support/enocean-link/classeo_profile_factory.html

The Class `eoProfileFactory` is designed according to the factory design pattern¹. The Factory `create()` method is then called at teach-in time. A concrete Profile is then created and provided as return value. To add new profiles and use them during teach-in the profiles need to added to the Factory class.

¹ <http://www.oodesign.com/factory-pattern.html>

EnOcean Link with own profiles extensions ADD OWN PROFILE DEFINITIONS TO ENOCEAN LINK

3. ADDING OR CHANGING PROFILES IN ENOCEAN LINK

To add a profile there a following options:

- Add MSC profiles – pure manufacturer specific profiles
- Add Profiles with new FUNC, new TYPE and with existing RORG
- Add Profiles with new TYPE, existing FUNC and RORG
- Alter/ extend existing profiles with Manufacturer specific identification – MAN_ID

To add MSC profiles we recommend not to extend the existing classes, but to define own profiles, as MSC is pure manufacturer specific.

All other uses cases can be easily realized by extending the existing classes or adding new classes.

Assuming the new added profiles supports one of the defined teach-in processes [2], then it is only required to add the profile to the eoProfileFactory.

First add header of new file / class – in case it is not included in existing file:

```
...
#include "eoEEP_D205xx.h"
#include "eoEEP_D220xx.h"
#include "eoEEP_D230xx.h"
#include "eoEEP_D231xx.h"
```

...

```
#include "eoEEP_NEW_PROFILE.h"
```

Then add the profile to the respective RORG switch/case:

- If MSC – then add to the `CreateProfile(const eoMessage &msg)` function.

```
eoProfile* eoProfileFactory::CreateProfile(const eoMessage &msg)
{
    switch (msg.RORG)
    {
        case RORG_4BS:
            if ((msg.data[3] & 0x88) == 0x80) // Is Learn telegram?
            {
                uint8_t rorg = RORG_4BS;
                uint8_t func = msg.data[0] >> 2;
                uint8_t type = ((msg.data[0] & 3) << 5) | (msg.data[1] >> 3);

                eoProfile *profile = eoProfileFactory::CreateProfile(rorg, func, type);
                if (profile != NULL)
                    profile->manufacturer = (msg.data[1] & 7) << 8 | msg.data[2];

                return profile;
            }

            break;

        ...
        case RORG_MSC:
            //OWN MSC HANDLING
```

EnOcean Link with own profiles extensions ADD OWN PROFILE DEFINITIONS TO ENOCEAN LINK

...

In case you add an new FUNC add the required create method to `CreateProfile(const uint8_t rorg, const uint8_t func, const uint8_t type)` function.

```
eoProfile* eoProfileFactory::CreateProfile(const uint8_t rorg, const uint8_t func,
const uint8_t type)
{
    eoProfile *eep = NULL;
    switch (rorg)
    {
    ...
        case RORG_4BS:
            switch (func)
            {
                case 0x02:
                    eep = new eoEEP_A502xx();
                    break;
                case 0x04:
                    eep = new eoEEP_A504xx();
                    break;
                case 0x05:
                    eep = new eoEEP_A505xx();
                    break;
                case 0x06:
                    eep = new eoEEP_A506xx();
                    break;
                case 0xXX: // NEW FUNC
                    eep = new eoEEP_A5XXxx(); // new FUNC CREATE
            }
    }
}
```

The type is selected later on by the respected `SetType()` function of the specific `eoEEPProfile` class inheritance - http://www.enocean.com/fileadmin/redaktion/support/enocean-link/classeo_profile.html.

If you add a new TYPE to existing FUNC, be sure to add the Matrix of EEP_ITEM to the class implementation (cpp file). For RORG = 0xA5, FUNC = 0x06 the EEP_MATRIX is

```
const EEP_ITEM listA506xx[numOfProfiles][numOfChan] =
{
// exist,bitoffs,bitssize,rangeMin,rangeMax,scaleMin, scaleMax,type;
//TYPE:00 - DOES NOT EXIST ONLZ PLACE HOLDER
{
//Temperature
{ false, 16, 8, 255, 0, -40.0, 00.0, S_RES, 0 },
{ false, 16, 8, 255, 0, -40.0, 00.0, S_RES, 0 },
{ false, 16, 8, 255, 0, -40.0, 00.0, S_RES, 0 }, },
//TYPE:01
{
{ true, 0, 8, 0, 255, 0, 5.1F, S_VOLTAGE, 0 }, //Voltage
{ true, 8, 8, 0, 255, 300, 30000, S_LUMINANCE, 0 }, //Light
{ true, 16, 8, 0, 255, 600, 60000, S_LUMINANCE, 0 }, //Light
},
//TYPE:02
```

EnOcean Link with own profiles extensions ADD OWN PROFILE DEFINITIONS TO ENOCEAN LINK

```
{
{ true, 0, 8, 0, 255, 0, 5.1F, S_VOLTAGE, 0 }, //Voltage
{ true, 8, 8, 0, 255, 0, 510, S_LUMINANCE, 0 }, //Light
{ true, 16, 8, 0, 255, 0, 1020, S_LUMINANCE, 0 } //Light
},
//TYPE:03
{
{ true, 0, 8, 0, 255, 0, 5.1F, S_VOLTAGE, 0 }, //Voltage
{ true, 8, 10, 0, 1000, 0, 1000, S_LUMINANCE, 0 }, //Light
{ false, 16, 8, 255, 0, -40.0, 00.0, S_RES, 0 }, }, };
```

http://www.enocean.com/fileadmin/redaktion/support/enocean-link/classeo_e_e_p_a506xx.html

If you require to implement special parsing logic source code please add it to the `SetValue()` function. Otherwise the inherited `SetValue()` function is called and parsing according to EEP_ITEM Matrix definition will be executed only.

In the `SetValue` function you can execute also manufacturer specific alterations of existing profiles. Be sure to check for the `manufacturer` attribute of the `eoProfile` Class. http://www.enocean.com/fileadmin/redaktion/support/enocean-link/classeo_profile.html

3.1. EXAMPLE

Please see attachment of this Application note to find reference implementation of Manufacturer specific alteration of existing Profiles by Eltako.

Occupancy sensor: FBH55, FBH63

EEP: A5-08-01

Teach In telegram includes Manufacturer ID - 0x00D.

These alterations were executed:

- Illumination Scaling extended from 0..510 lux to 0..2048 lux
- No Temperatur in DB1
- No Occupancy Button im DB0_Bit0

Illumination sensor: FAH60

EEP: A5-06-01

Teach In telegram includes Manufacturer ID - 0x00D.

These alterations were executed:

- Data_byte3 = 0 – 100 lux, linear n = 0x00 – 0x64 (valid when DB2 = 0x00)
- Data_byte2 = Illuminance 300 – 30.000 lux, linear n = 0x00 – 0xFF
- Data_byte1 = not used

Disclaimer

The information provided in this document describes typical features of the EnOcean radio system and should not be misunderstood as specified operating characteristics. No liability is assumed for errors and / or omissions. We reserve the right to make changes without prior notice. For the latest documentation visit the EnOcean website at www.enocean.com

EnOcean Link with own profiles extensions
ADD OWN PROFILE DEFINITIONS TO ENOCEAN LINK
